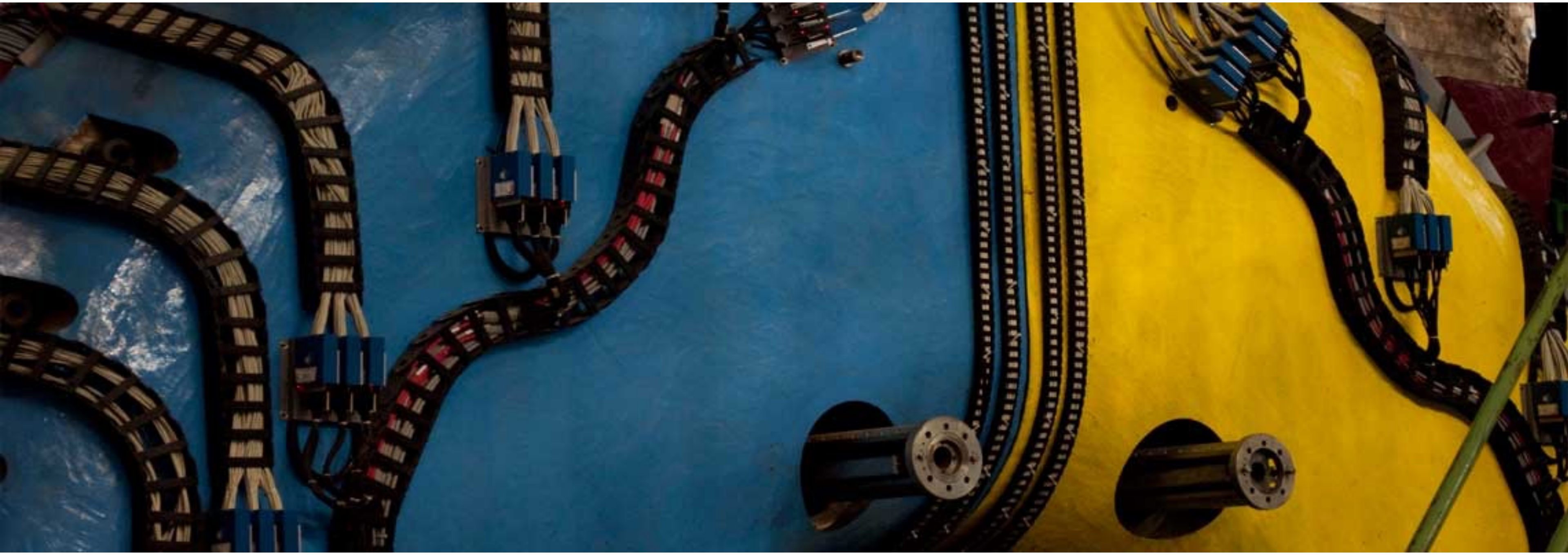
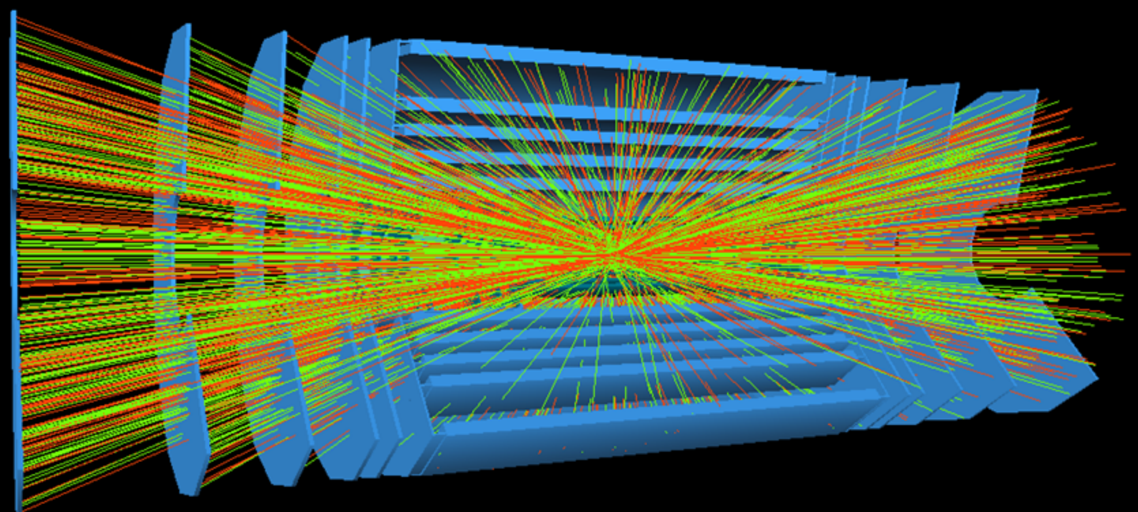


Introduction to trigger concepts

F.Pastore (Royal Holloway Univ. of London)

francesca.pastore@cern.ch

The data deluge

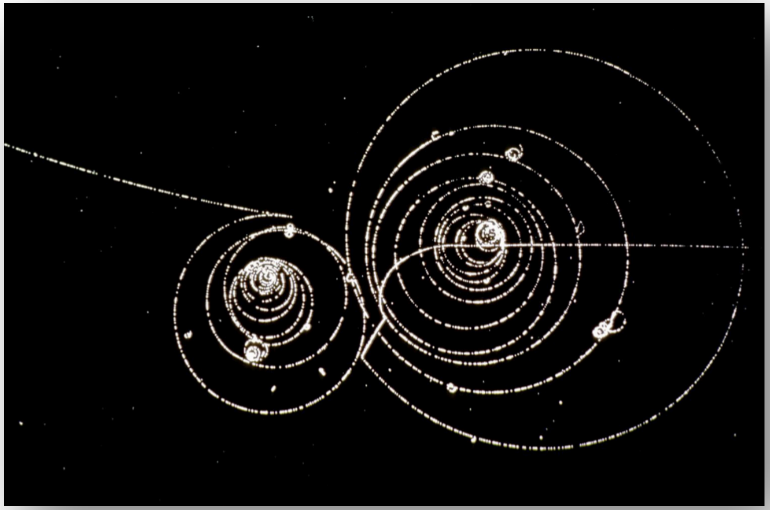


- In many systems, like particle physics or astronomy experiments, to store all the possibly relevant data provided by the sensors is UNREALISTIC and often becomes also UNDESIRABLE
- Three approaches are possible:
  - Reduced amount of data (packing and/or filtering) → ***Trigger!***
  - Faster data transmission and processing
  - Both!

The trigger concept

A very good photo during your holidays

Cloud-chamber images recorded on film



- **click the button:** open the bolt and let the sensors operate
  - take the photo when the **subjects** are ready
  - **focus** the image
  - only if there is enough **light** for your lenses (or add a flash light)
  - only if your hand is not **shaking**

→ **The trigger starts the photo process**

→ First identify the interesting event

→ Ensure the sensitivity to parameters

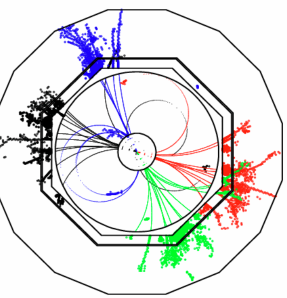
→ Ensure a good synchronisation

How to decide when taking the photo?

Trigger concept in HEP

limited by time constraints

limited by computing resources



Trigger

Identify the interesting process

DAQ

Start data acquisition

Computing

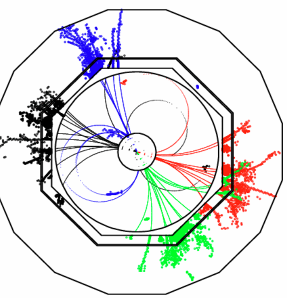
Storage

Record & Process data

Trigger concept in HEP

limited by time constraints

limited by computing resources



Trigger

Identify the interesting process

DAQ

Start data acquisition

Computing

Storage

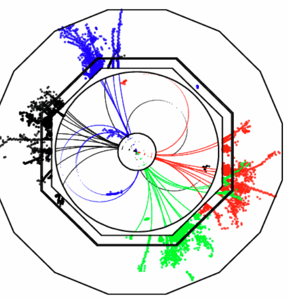
Record & Process data

- **The** **constrain** **between trigger and DAQ rate is the storage and the offline computing capabilities**
  - Define the maximum allowed rate

Trigger concept in HEP

limited by time constraints

limited by computing resources



Trigger

Identify the interesting process

DAQ

Start data acquisition

Computing

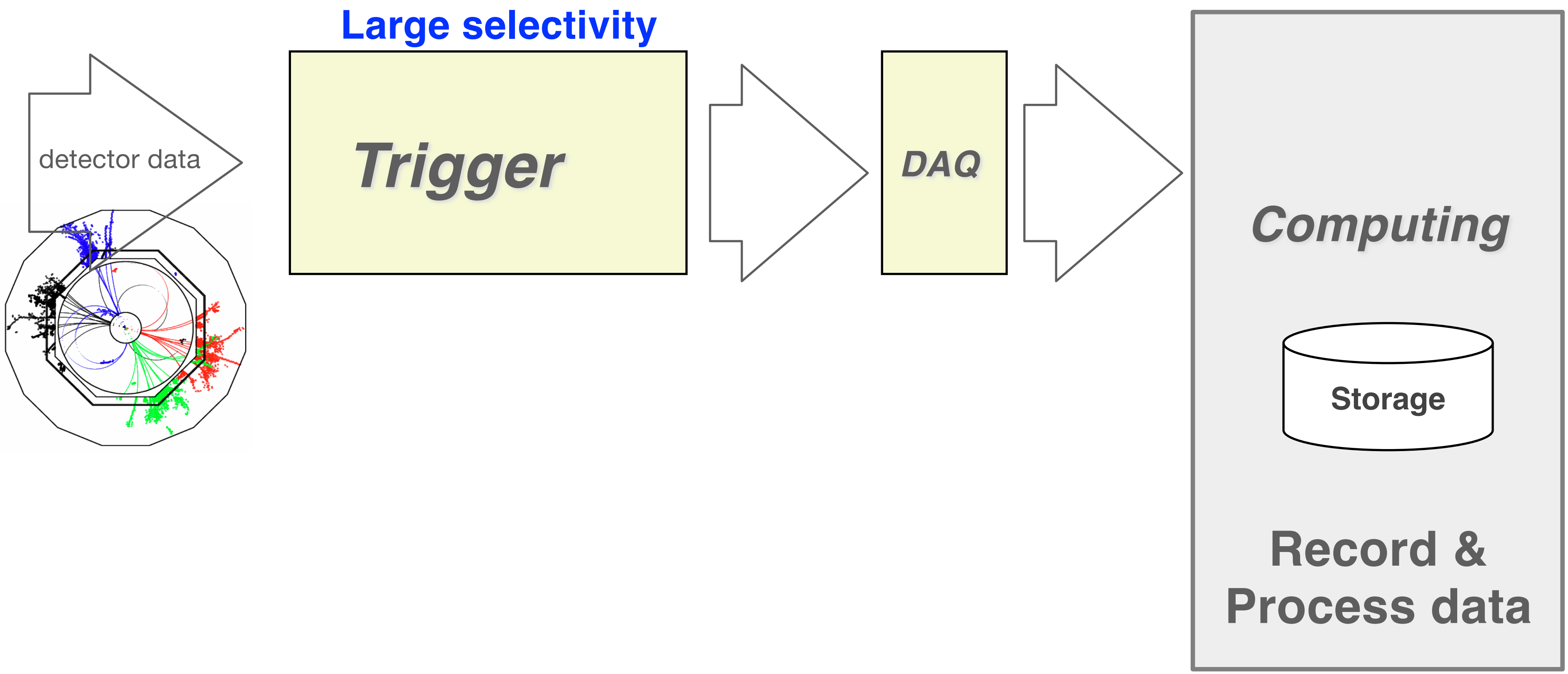
Storage

Record & Process data

- **The** **constrain** **between trigger and DAQ rate is the storage and the offline computing capabilities**
  - Define the maximum allowed rate
- **Which is the** **balance** **between Trigger and DAQ resources?**

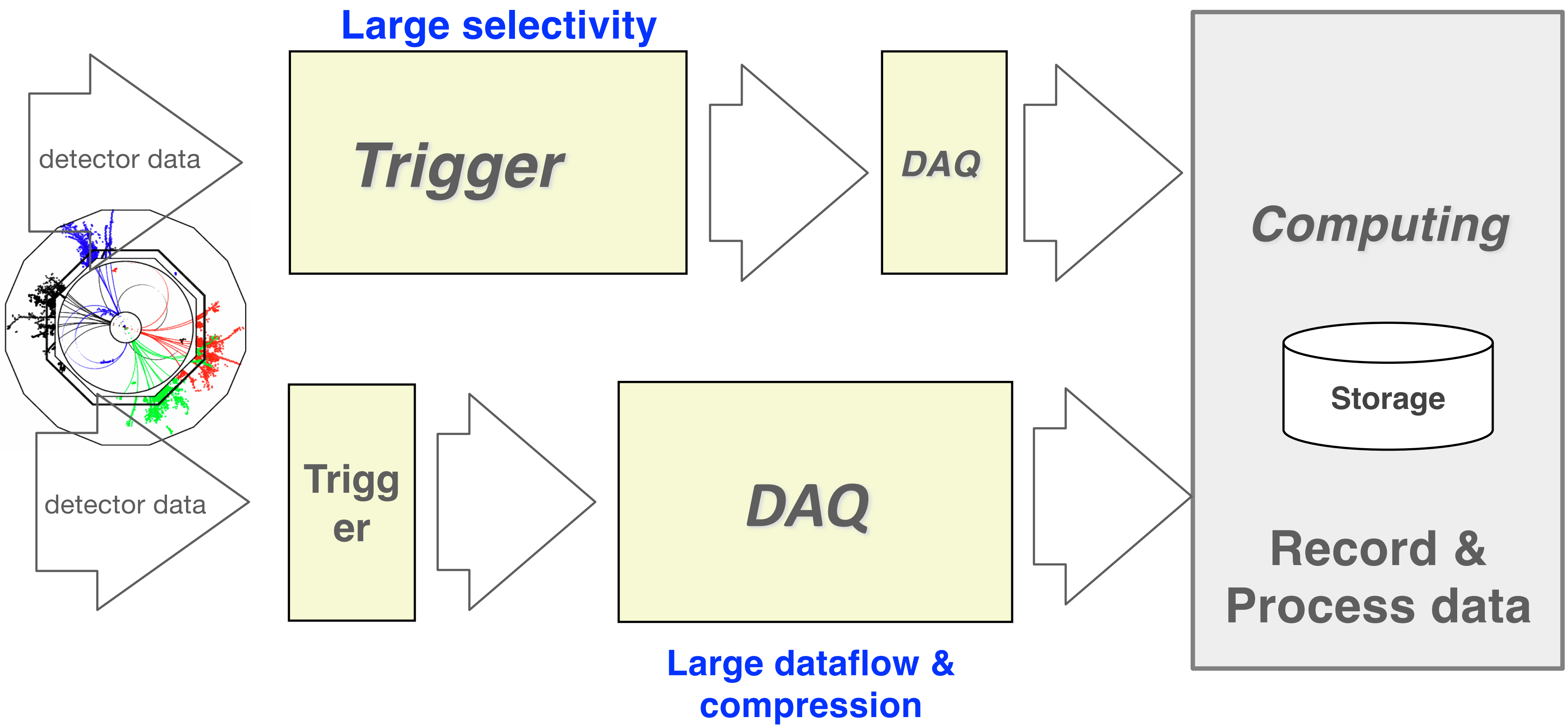
Balance between trigger and DAQ

- **If the trigger is highly selective, one can reduce the size of the dataflow**



Balance between trigger and DAQ

- If the trigger is highly selective, one can reduce the size of the dataflow



- If the selectivity of the trigger is not enough, due to large irreducible background, a large data flow (and data compression) is needed

1: Which is the expected input rate?

The expected rate is derived from the physics process (X-section) and the detector sensitivity

For a collider experiment: x Luminosity

$$R = \sigma_{in} \times L$$

LHC: the trigger challenge!

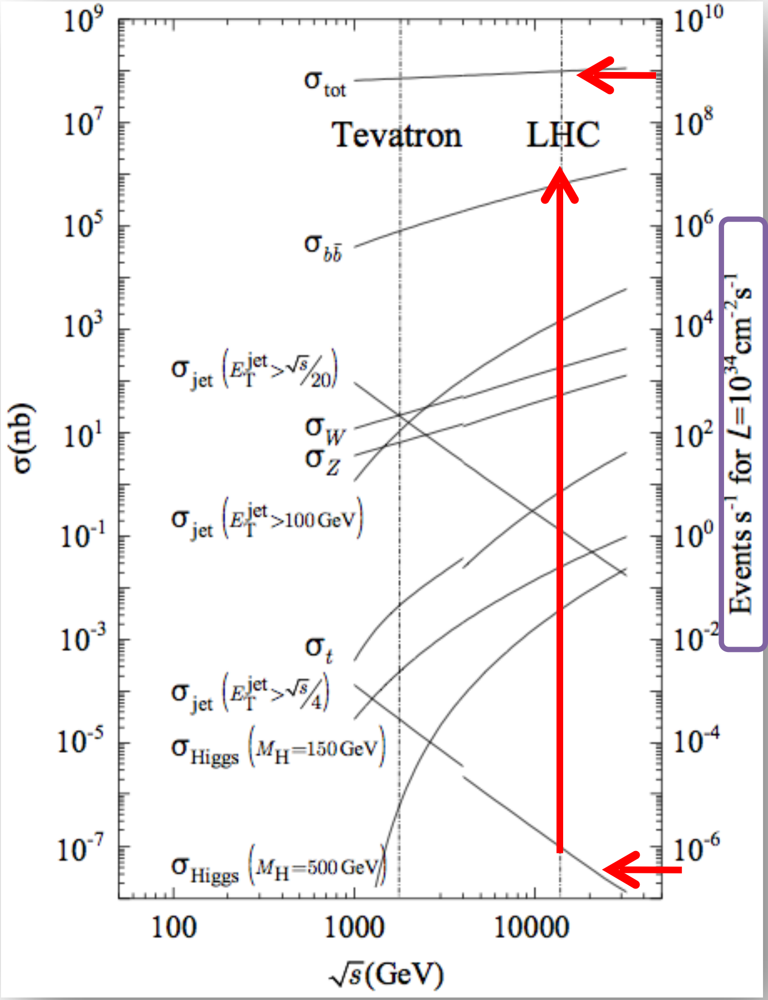
Total non-diffractive p-p cross section is **70 mb**

Total expected rate is ~ GHz!!!

Huge range of cross-sections and production rates at design:

Beauty (0.7 mb)	– 1000	Hz
W/Z (200/60 nb)	– 100	Hz
Top (0.8 nb)	– 10	Hz
Higgs - 125 GeV (30 pb)	– 0.1	Hz

$$\frac{\sigma_{tot}}{\sigma_{H(500\,\mathrm{GeV})}} \approx \frac{100\,mb}{1\,pb} \approx 10^{11}$$



In hadronic colliders, trigger selection is crucial

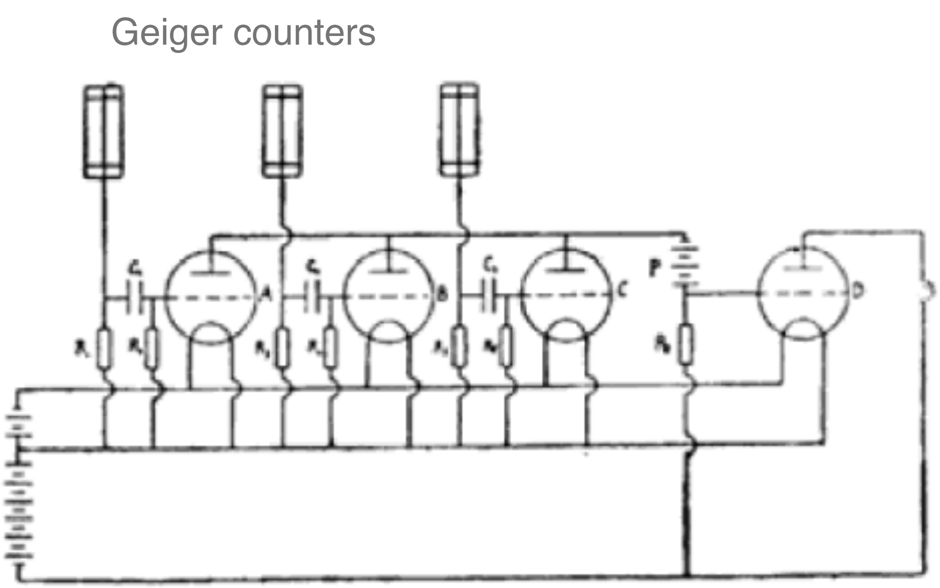
2: which is an easy selection?

- Use discriminating features within widely extended systems
- Reality is:
  - The trigger accepts events with features **similar** to the signal
  - The final rate is often dominated by not interesting physics

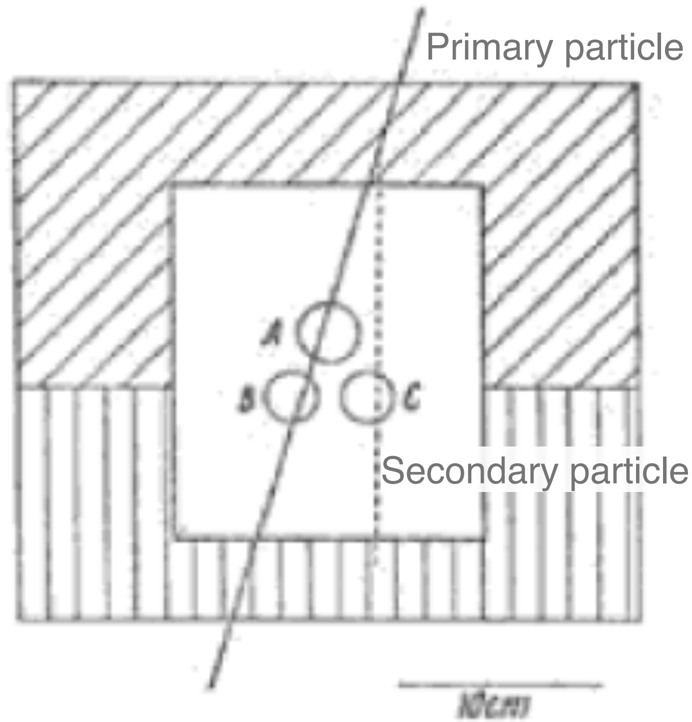
"Method of Registering Multiple Simultaneous Impulses of Several Geiger Counters"

Bruno Rossi, Nature 1930

– Online coincidence of three signals

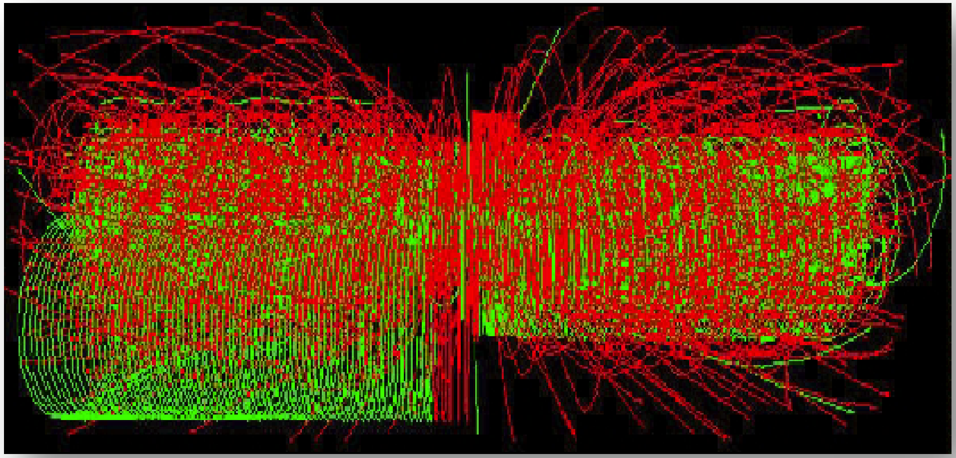


Astronomia e Fisica a Firenze: dalla Specola ad Arcetri, Firenze Universiry Press, 2017



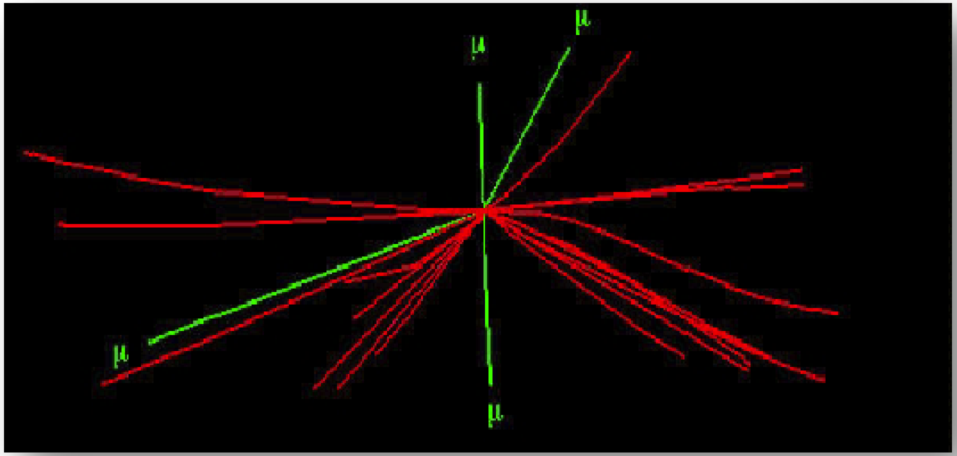
Which is a good trigger for the Higgs Boson @LHC?

All tracks



Only high-pT tracks

+30 MinBias



Higgs -> 4μ

@LHC: proton-proton collider

signal events **Higgs → 4μ** as it appears at the LHC (concurrently with soft collisions from other p-p interactions)

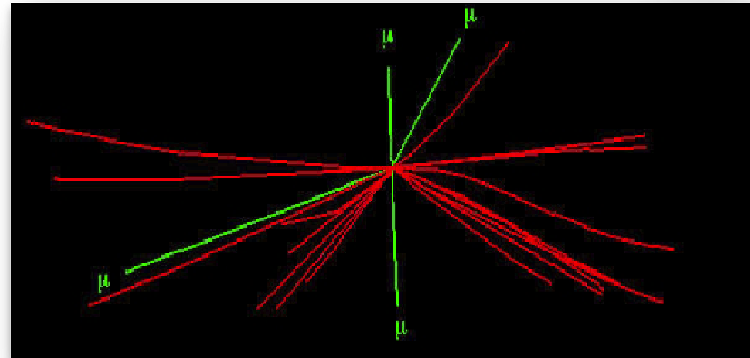
The trigger selection based on **high transverse momentum muons** (at least one)

Which is the best filter?

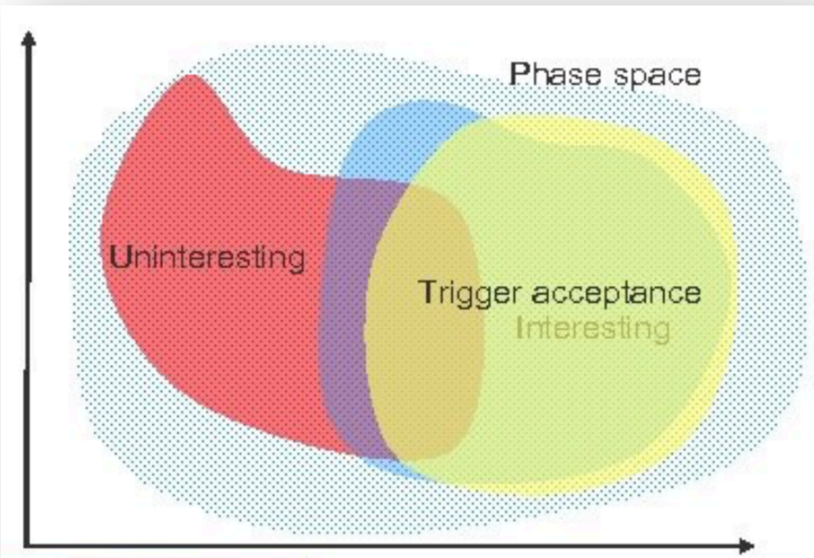
Trigger requirements

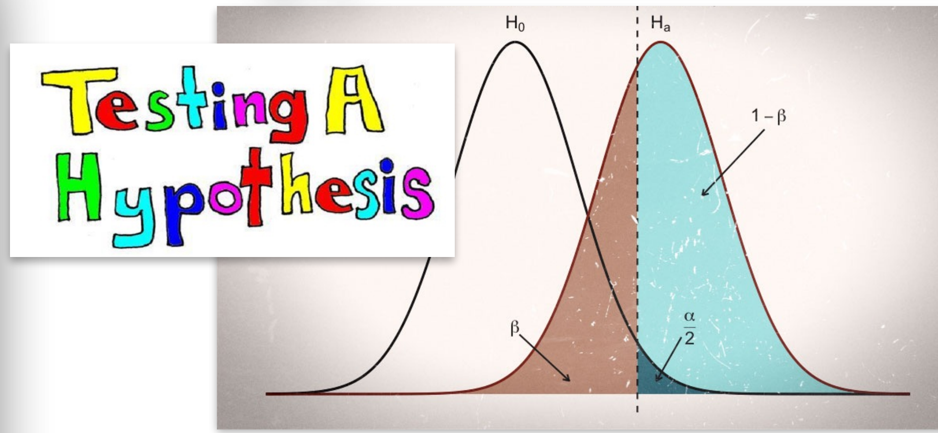
Trigger parameters to easily distinguish

- Remember the Higgs discovery @LHC:
  - high p_T muons are interesting (**signal**)
  - low p_T muons are not interesting (**background**)
- Which p_T threshold then?
  - background is unavoidable!



Higgs -> 4μ





compromise strongly depends on the source of physics events: cosmic rays? lepton colliders? hadron colliders?

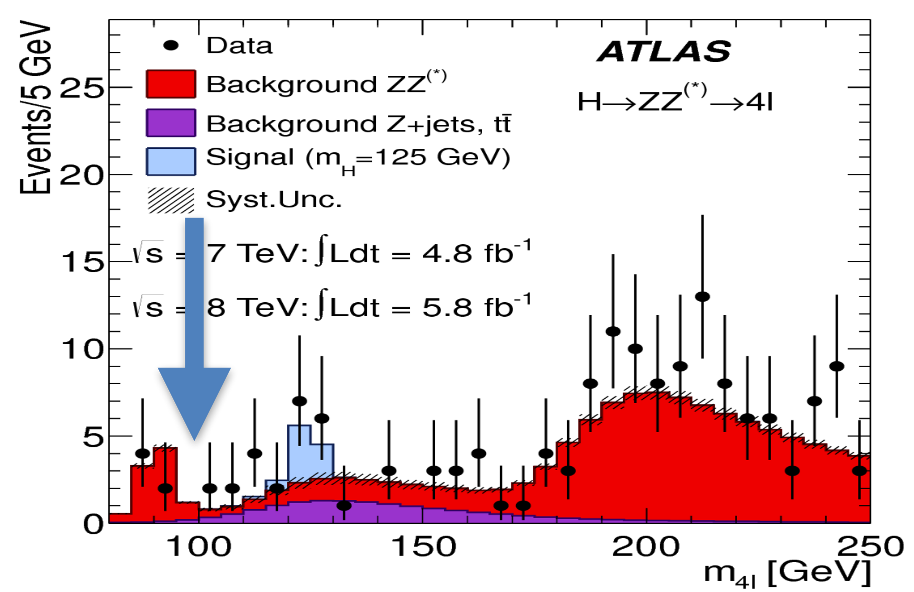
Requirement n.1: *high signal efficiency*

$$\epsilon_{trigger} = \frac{N_{good}^{accepted}}{N_{good}^{expected}}$$

drives the design of the experiment and of the Trigger and DAQ architecture

- **Depends on the acceptance**
  - of detectors, DAQ,.....
- **Depends on the selection**
  - tuned on Monte Carlo simulations

4-leptons invariant mass, selected for H→ZZ→4l



Which is an acceptable efficiency?

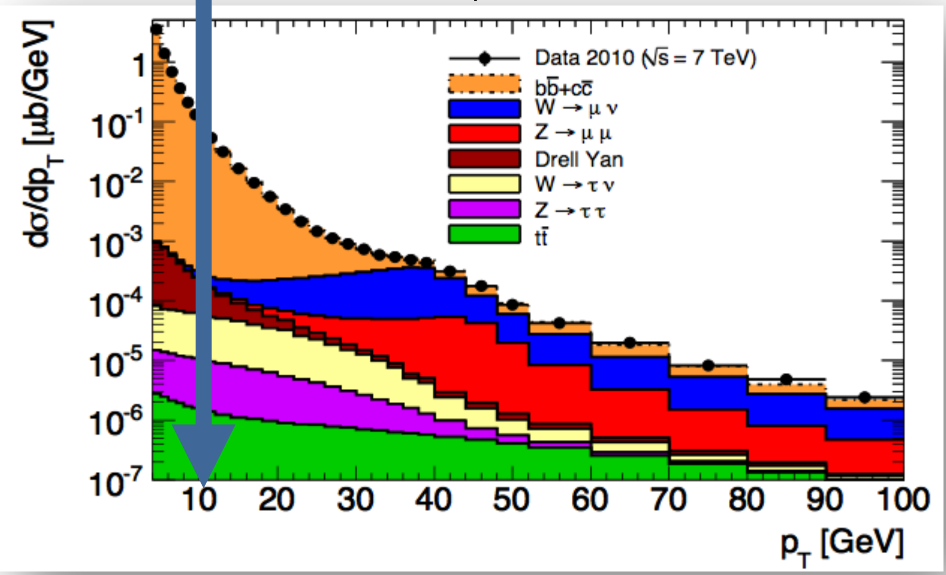
- **Lepton colliders (precision machines)**
  - **all is interesting** (no physics background)
  - efficiency close to 100%
  - 99.9% accepted events, 0.1% rejected
- **Hadron colliders (discovery machines)**
  - **large physics background**
  - good efficiency > 50%
  - 99.9% rejected events, 0,1% accepted

Requirement 2: *good background rejection*

$$R_{trigger} = \frac{N_{bad}^{expected}}{N_{bad}^{accepted}}$$

- **Rate control is critical**
  - in particular on hadron colliders
- **Need solid understanding of background shapes**
  - Monte Carlo simulation of **all physics**, detector noise, machine backgrounds
- **Not easy!**
- **Backgrounds are often known with great uncertainties**
  - trigger must be **flexible and robust**

Inclusive single muon p_T spectrum @LHC



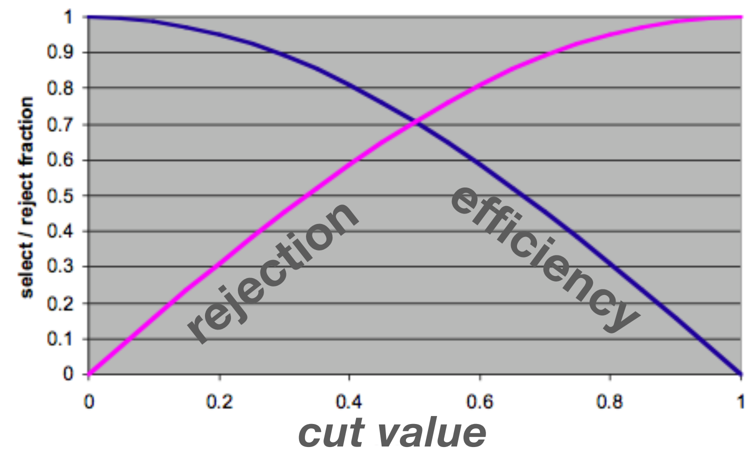
Which is an acceptable rejection?

- **Lepton colliders (precision machines)**
  - good background rejections ~10
- **Hadron colliders (discovery machines)**
  - good background rejections > ~10^6

….with compromises?

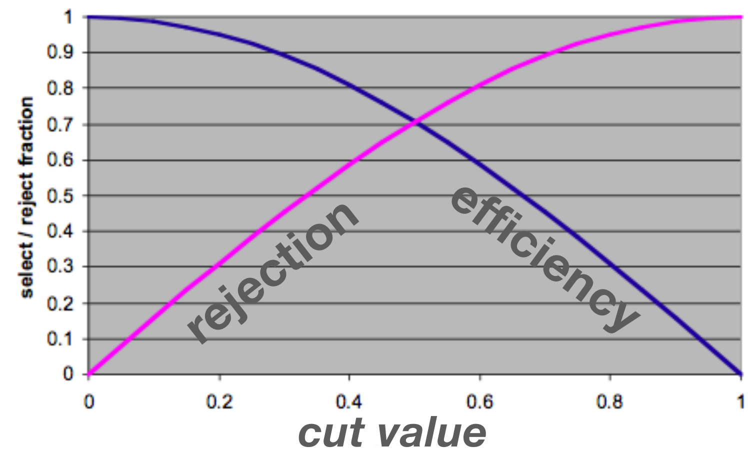
....with compromises?

➡ **What if any of the two requirements cannot be realised?**



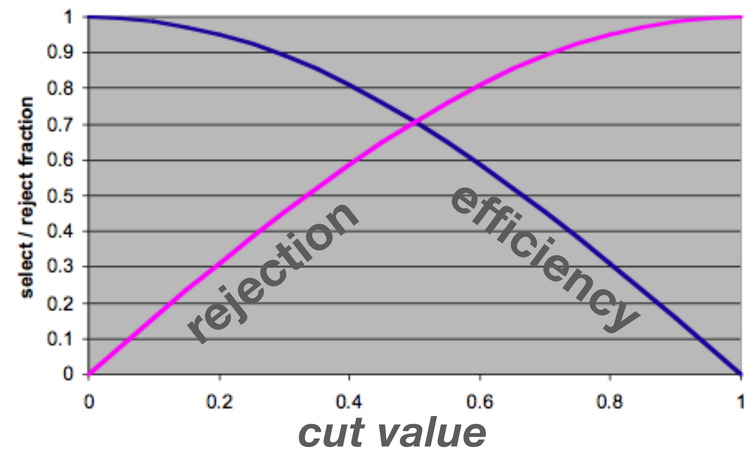
....with compromises?

- ➡ **What if any of the two requirements cannot be realised?**
- ➡ **Refine your selection!**



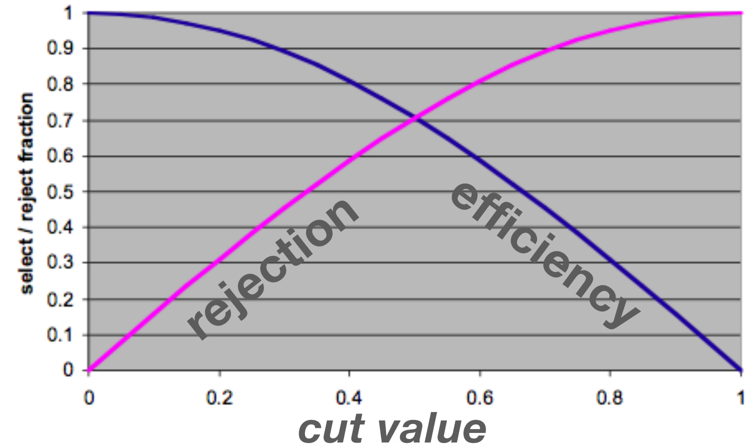
….with compromises?

- ➡ What if any of the two requirements cannot be realised?
- ➡ Refine your selection!
- ➡ Add new **parameters**
  - ➡ maybe need new detectors?

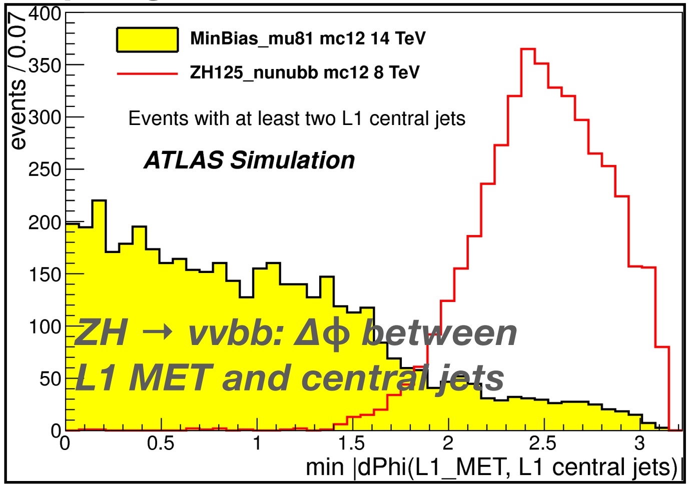


....with compromises?

- ➡ What if any of the two requirements cannot be realised?
- ➡ Refine your selection!
- ➡ Add new **parameters**
  - ➡ maybe need new detectors?
- ➡ Add more **complex** selections
  - ➡ maybe need to buy more processors?
  - ➡ or run faster algorithms?
  - ➡ maybe need new TDAQ architecture?
- ➡ **Scalability** is always an advantage

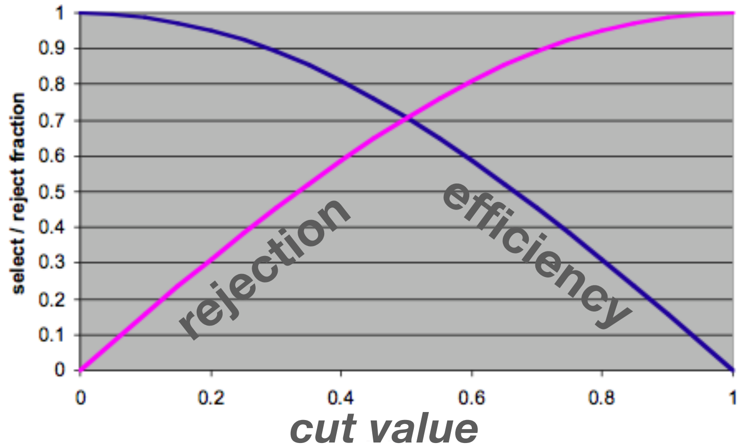


Topological cuts @L1 ATLAS

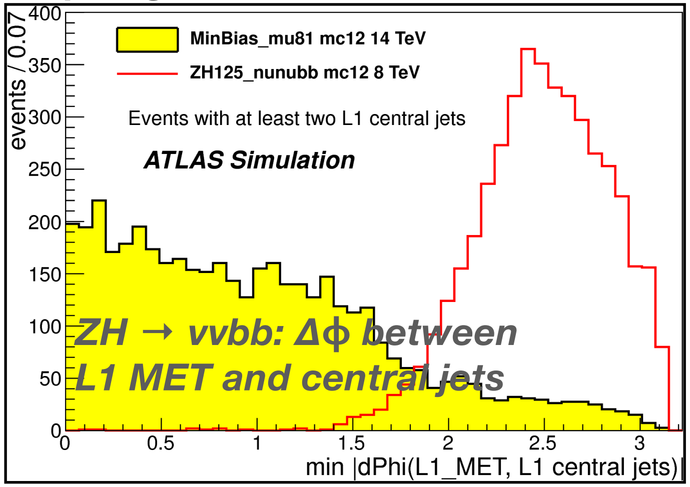


....with compromises?

- ➡ **What if any of the two requirements cannot be realised?**
- ➡ **Refine your selection!**
- ➡ **Add new parameters**
  - ➡ maybe need new detectors?
- ➡ **Add more complex selections**
  - ➡ maybe need to buy more processors?
  - ➡ or run faster algorithms?
  - ➡ maybe need new TDAQ architecture?
- ➡ **Scalability is always an advantage**
- ➡ **Whatever criteria you choose, discarded events are lost for ever!**

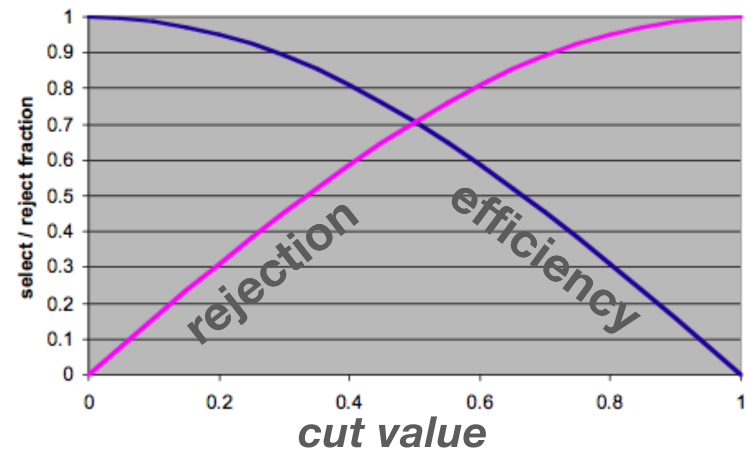


Topological cuts @L1 ATLAS

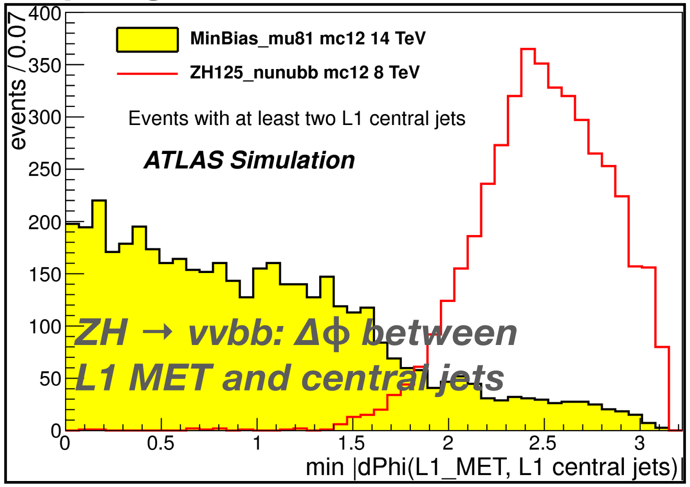


....with compromises?

- ➡ **What if any of the two requirements cannot be realised?**
- ➡ **Refine your selection!**
- ➡ **Add new parameters**
  - ➡ maybe need new detectors?
- ➡ **Add more complex selections**
  - ➡ maybe need to buy more processors?
  - ➡ or run faster algorithms?
  - ➡ maybe need new TDAQ architecture?
- ➡ **Scalability is always an advantage**
- ➡ **Whatever criteria you choose, discarded events are lost for ever!**
- ➡ **Always ask yourself: is the trigger reliable?**
  - ➡ If you don't trust your trigger, add control samples
  - ➡ and always **monitor** your trigger



Topological cuts @L1 ATLAS



Trigger efficiency is a parameter of your measurement

$$BR_{signal} = \frac{N_{candidate} - N_{bkg}}{\alpha \cdot \epsilon_{total} \cdot \sigma_{Bs} \cdot \int L dt}$$

Trigger efficiency is a parameter of your measurement

$$BR_{signal} = \frac{N_{candidate} - N_{bkg}}{\alpha \cdot \epsilon_{total} \cdot \sigma_{Bs} \cdot \int L dt}$$

$$\alpha \cdot \varepsilon_{\text{total}} = \alpha \cdot \varepsilon_{\text{Tracking}} \cdot \varepsilon_{\text{Reco}} \cdot \varepsilon_{\text{L1-Trig}} \cdot \varepsilon_{\text{L2-Trig}} \cdot \varepsilon_{\text{L3-Trig}} \cdot \varepsilon_{\text{vertex}} \cdot \varepsilon_{\text{analysis}}$$

Trigger efficiency is a parameter of your measurement

$$BR_{signal} = \frac{N_{candidate} - N_{bkg}}{\alpha \cdot \epsilon_{total} \cdot \sigma_{Bs} \cdot \int L dt}$$

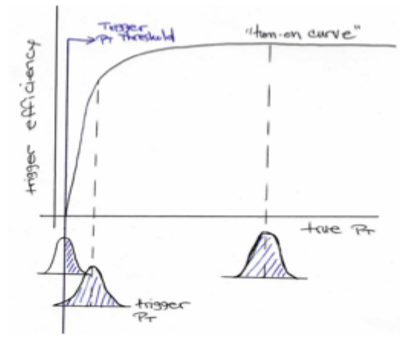
$$\alpha \cdot \varepsilon_{\text{total}} = \alpha \cdot \varepsilon_{\text{Tracking}} \cdot \varepsilon_{\text{Reco}} \cdot \varepsilon_{\text{L1-Trig}} \cdot \varepsilon_{\text{L2-Trig}} \cdot \varepsilon_{\text{L3-Trig}} \cdot \varepsilon_{\text{vertex}} \cdot \varepsilon_{\text{analysis}}$$

Trigger efficiency must be **precisely known**, since it enters the cross-sections measurements

For some precise measurements, the crucial performance parameter is not the efficiency itself, but the **systematic error** on determining it

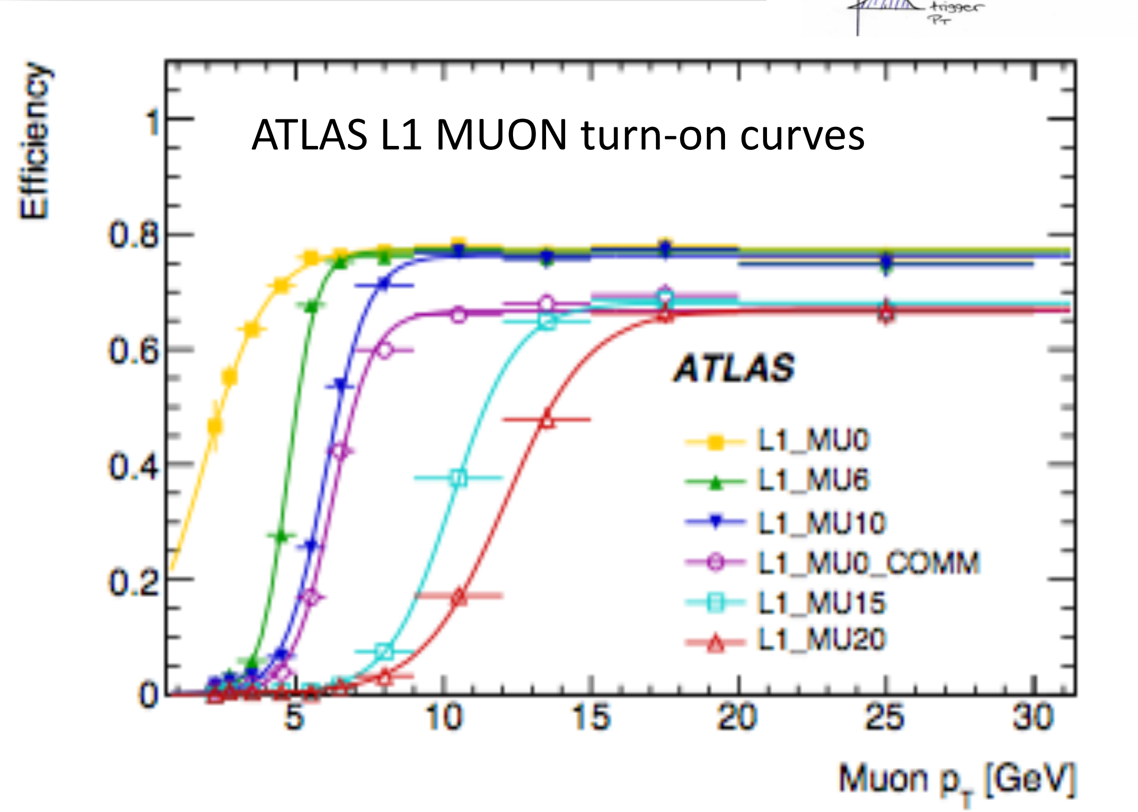
Trigger efficiency measurement

The threshold is not exactly applied as a step function. Better as an Error function, usually called trigger turn-on



- The capability of controlling the rate depends on the **resolution** on the trigger parameter

- Crucial is the **step region**, in which efficiency changes very quickly and contamination from background can be important (often abundant!)
  - If **quick**, better background suppression
  - If **slow**, can be better extrapolated and systematic error can be reduced



Trigger for precision measurements: BaBar

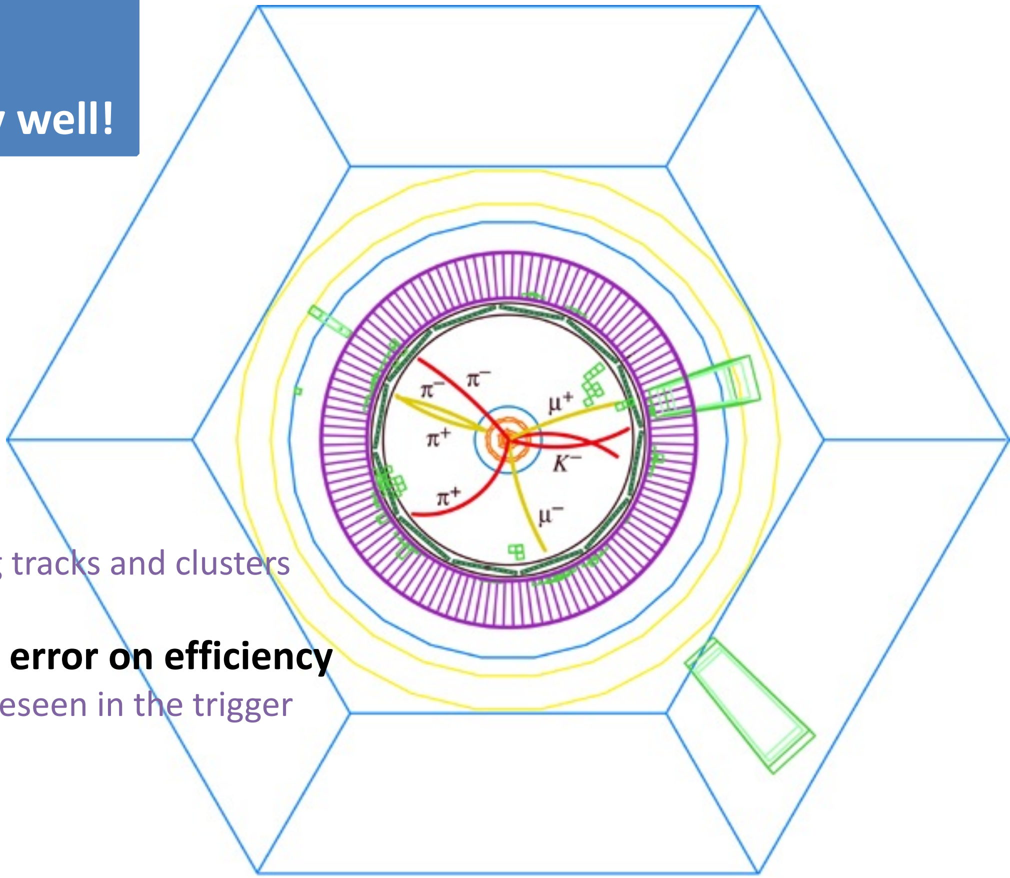
Goal: precision measurement of CP violation

Reduce systematic uncertainties

Know your detector (and your trigger) very well!

Golden event in the BaBar Detector
e+e- collision producing a B and an anti-B

Golden B (for CP violation)
Tagging B

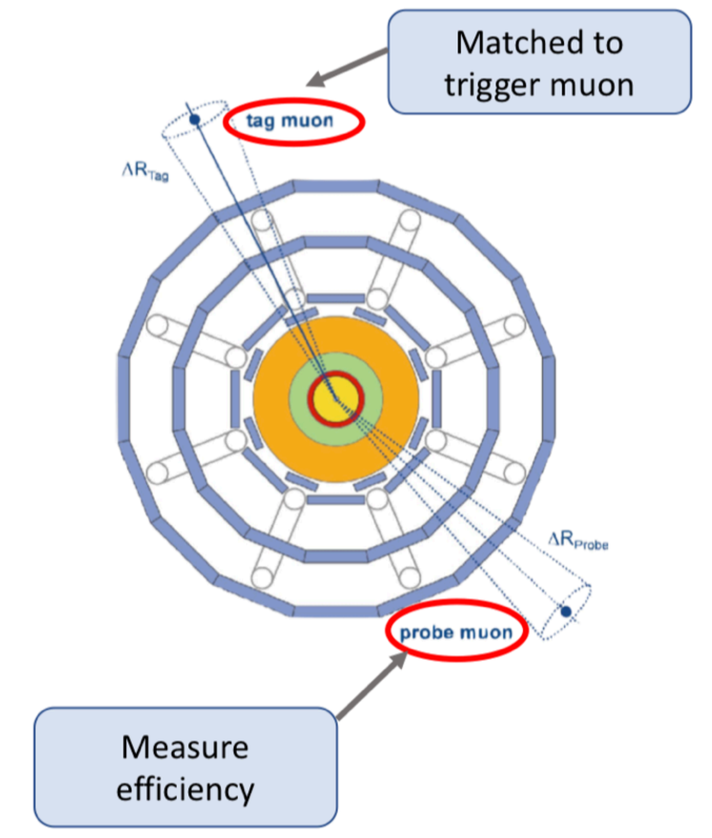


- **Multitude of trigger selections:**
  - **Charged tracks** in the drift chamber
  - **EM calorimeter clusters** with different E_T cuts
  - Particle identification capability
- **Exclusive selections based on event topology**
  - **Number of objects**, angular separation, matching tracks and clusters
- **Study both signal and background to minimise error on efficiency**
  - The selection of background samples must be foreseen in the trigger design

Methods to measure trigger efficiency

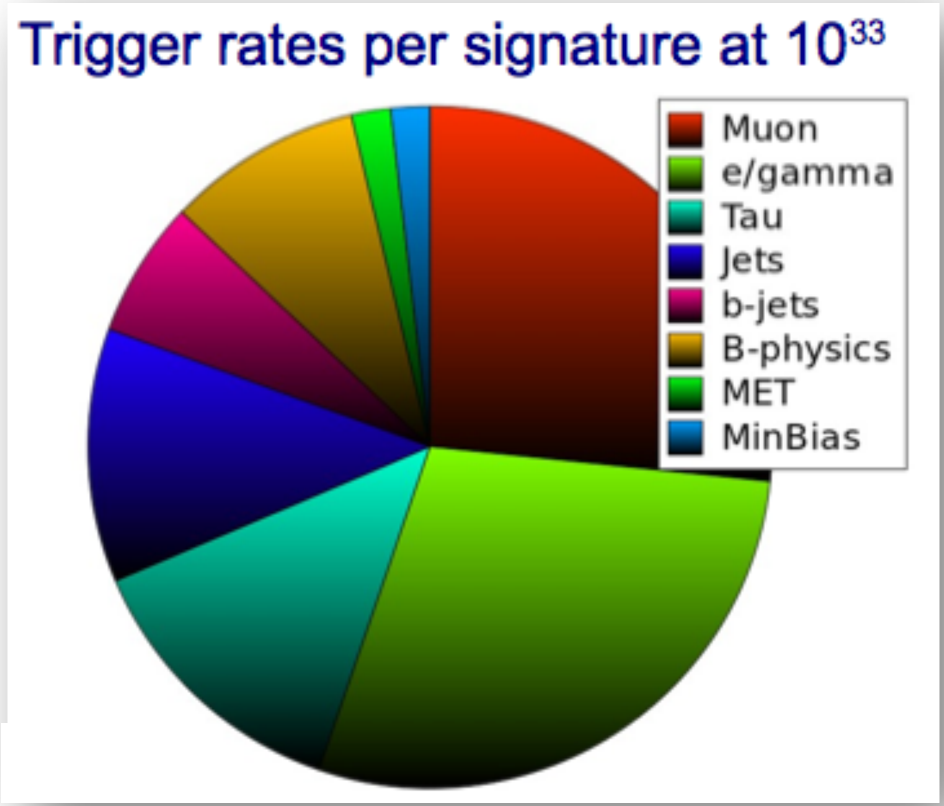
$$\text{Efficiency} = \frac{\text{number of events that passed the selection}}{\text{number of events without that selection}}$$

- **Experimental technique called "Tag-and-probe"**: trigger on one particle (tag), measure how often another (probe) passes trigger
  - exploit a well-known physics process (like Z-boson decay into leptons) to select a clean sample
  - applicable on specific signals, usually with leptons
- **Boot-strap**: use looser (prescaled) trigger to measure efficiency of higher threshold trigger
- **Orthogonal trigger**: use one trigger (e.g. muon trigger) to measure efficiency of a different, independent trigger (e.g. jet trigger)
- **Simulation/emulation**: Monte Carlo simulation suffers from reduced knowledge of background, so it is used
  - to study acceptance on signal physics processes
  - to validate online selections



Trigger strategy @ colliders: ATLAS menu

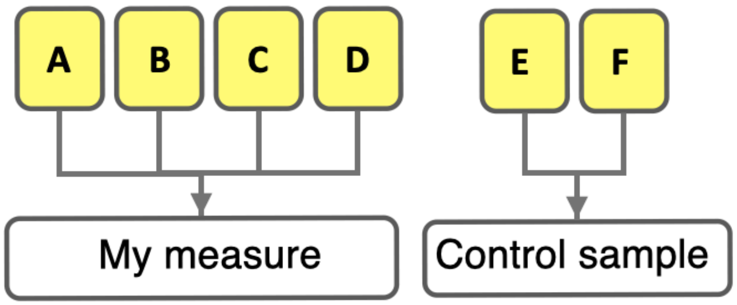
ATLAS



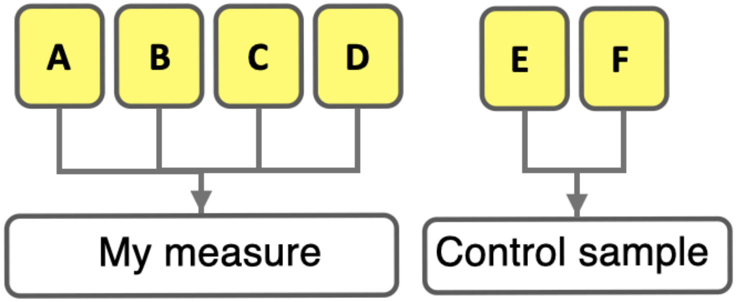
- Inclusive triggers to collect the **signal samples**
  - Single high-p_T
    - $e/\mu/\gamma$ (p_T>20 GeV)
    - jets (p_T>100 GeV)
  - Multi-object events
    - e-e, e-μ, μ-μ, e-τ, e-γ, μ-γ, etc… to further reduce the rate
- **Back-up triggers** designed to spot problems, provide control samples (often pre-scaled)
  - Jets (p_T>8, 20, 50, 70 GeV)
  - Inclusive leptons (p_T > 4, 8 GeV)
  - Lepton + jet

trigger streamers

➡ **Physics triggers** (the bulk of the events): multiple & independent



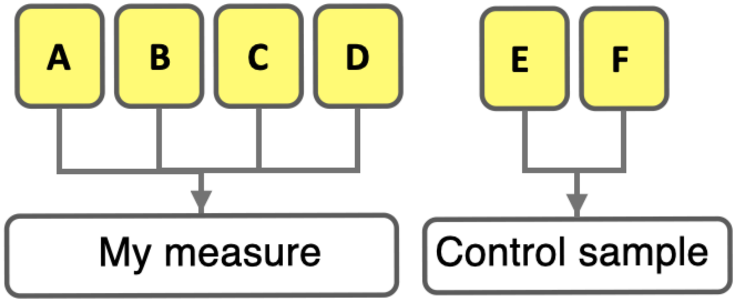
- **Physics triggers (the bulk of the events): multiple & independent**



- **Calibration triggers**
  - Detectors calibrations
  - Detectors and trigger efficiency
  - Tagging efficiency
  - Energy scale measurements

trigger streamers

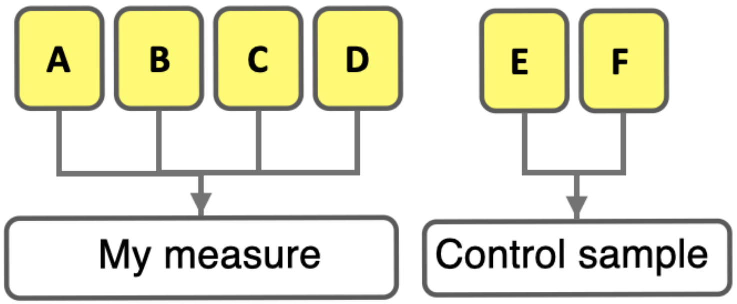
- **Physics triggers (the bulk of the events): multiple & independent**



- **Calibration triggers**
  - Detectors calibrations
  - Detectors and trigger efficiency
  - Tagging efficiency
  - Energy scale measurements
- **Background triggers**
  - Instrumental and physics background
  - Description of background
  - Understand resolutions, including the under-threshold population

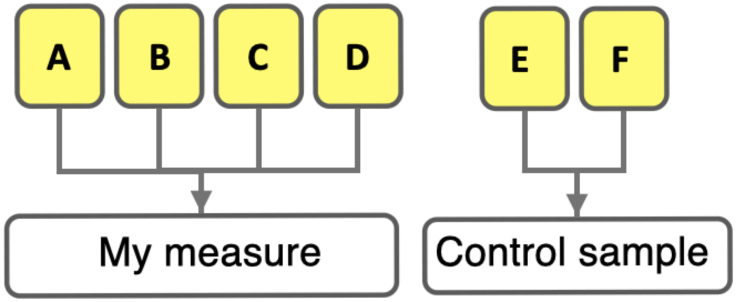
trigger streamers

- **Physics triggers** (the bulk of the events): multiple & independent



  - **Calibration triggers**
    - Detectors calibrations
    - Detectors and trigger efficiency
    - Tagging efficiency
    - Energy scale measurements
  - **Background triggers**
    - Instrumental and physics background
    - Description of background
    - Understand resolutions, including the under-threshold population
  - **Monitor triggers**
    - To monitor the trigger itself (remember, lost events are lost for ever!)

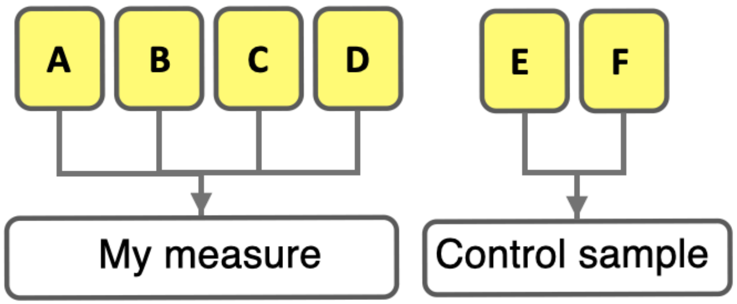
- **Physics triggers** (the bulk of the events): multiple & independent



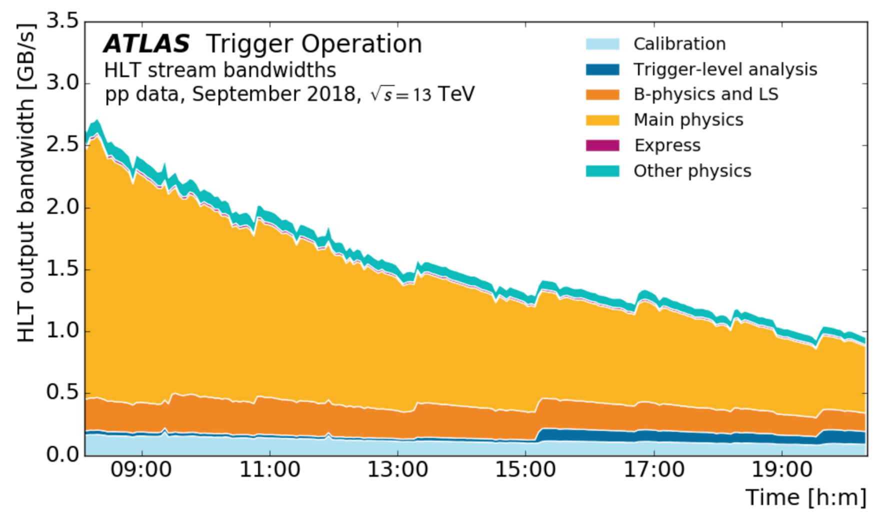
often prescaled

- **Calibration triggers**
  - Detectors calibrations
  - Detectors and trigger efficiency
  - Tagging efficiency
  - Energy scale measurements
- **Background triggers**
  - Instrumental and physics background
  - Description of background
  - Understand resolutions, including the under-threshold population
- **Monitor triggers**
  - To monitor the trigger itself (remember, lost events are lost for ever!)

- **Physics triggers (the bulk of the events): multiple & independent**



***Prescale**: record 1/N events. Useful for collecting samples of high-rate triggers without swamping the DAQ system*



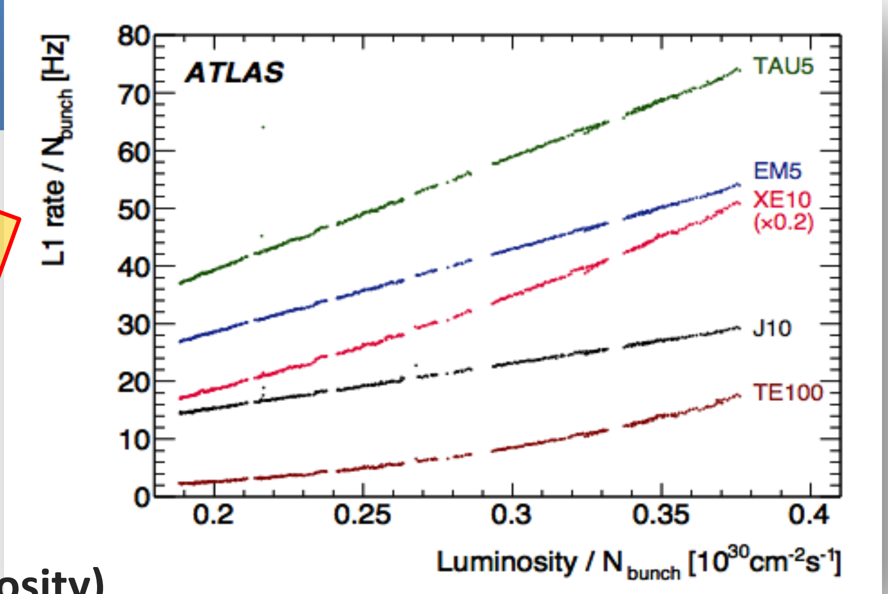
often prescaled

- **Calibration triggers**
  - Detectors calibrations
  - Detectors and trigger efficiency
  - Tagging efficiency
  - Energy scale measurements
- **Background triggers**
  - Instrumental and physics background
  - Description of background
  - Understand resolutions, including the under-threshold population
- **Monitor triggers**
  - To monitor the trigger itself (remember, lost events are lost for ever!)

Rate allocations of the trigger signatures

- ✓The target is the final allowed DAQ bandwidth
- ✓The rate allocation on each trigger based on
  - ✓Physics goals (plus calibration, monitoring samples)
  - ✓Required efficiency and background rejection
  - ✓Bandwidth consumed

Rates scale linearly with luminosity, but linearity is smoothly broken due to pile-up



Trigger Efficiency

$$R_i = L \int_{p_T\_\inf}^{p_T\_cutoff} \frac{d\sigma_i}{dp_T} \cdot \varepsilon(p_T)\, dp_T$$

Need to extrapolate trigger rates (at given Luminosity)

- For trigger design and commissioning: large samples of <u>simulated</u> data, including large cross-section backgrounds
  - 7 million of non-diffractive events used in the ATLAS trigger design
  - Large uncertainties due to detector response and background cross-sections: apply **safety factors**, then tuned with data
- During running (at colliders): but only some rates can be easily extrapolated to higher L

Priority List for >300 Hz

Chain		Unique rate L1 (Hz)	Unique rate L2 (Hz)	Unique rate EF (Hz)	*Sorted by* Problem level
EF_xe60_verytight_noMu	SUSY/Exotics	0	0	0.5	EF (pileup)
EF_j100_a4tc_EFFS_ht400	SUSY	0	0	2.5	EF
EF_4j45_a4tc_EFFS	SUSY/SM	0	0	2	EF
EF_5j30_a4tc_EFFS		0	5	3	EF
EF_j240_a10tc_EFFS	Exotics/SM	0	0	1	EF
EF_tau29_loose1_xs45_loose_noMu_3L1J10	Higgs	0	40	5	EF
EF_b10_medium_4j30_a4tc_EFFS	Top/Higgs	0	4	10	EF
EF_2mu4_BmumuX	B-physics	0	7	0.9	EF
EF_2mu4_Jpsimumu		0	6	1.7	EF
EF_mu4mu6_DiMu		0	25	6.5	EF
EF_mu4mu6_DiMu_DY20	SM	0	10	5?	EF
EF_2MUL1_l2j30_HV_allMS	Exotics	0	?	?	EF
EF_mu20i_medium	5×10^{33} prep.	0	15	3	EF
EF_mu18_MG_medium	Many	0	0	60	EF
EF_mu18_medium		0	0	60	EF
EF_e60_loose	(Exotics)	0	5	7	EF,client
EF_mu15/18/22_njX?	SUSY/??	100	10	?	EF,non-validated
EF_g22_hiptrt?	Exotics	0	?	<1?	non-validated
EF_e15_medium_xe40_noMu	SUSY/Exotics	310	70?	1.3	L2 (pileup)
EF_j55_a4tc_EFFS_xe55_medium_noMu_dphi2j30xe10		70	210	1.5	L2
EF_e10_medium_mu6_topo_medium	Higgs	1200	9	1	L1
EF_tau20_medium_e15_medium	Higgs	3700	10	1	L1
EF_xe60_tight_noMu	SUSY	680?	150?	1	L1,L2 (pileup),EF
EF_e10_medium_mu6	Higgs/SUSY	1200	75	10	L1, EF
EF_l2j30_Trackless_HV_L1MU6	Exotics	1500?	0.5	0.5	L1
Total extra rate		6500	600	100	Peak at 3×10^{33}

Build up a trigger system

Ensure good efficiency with…

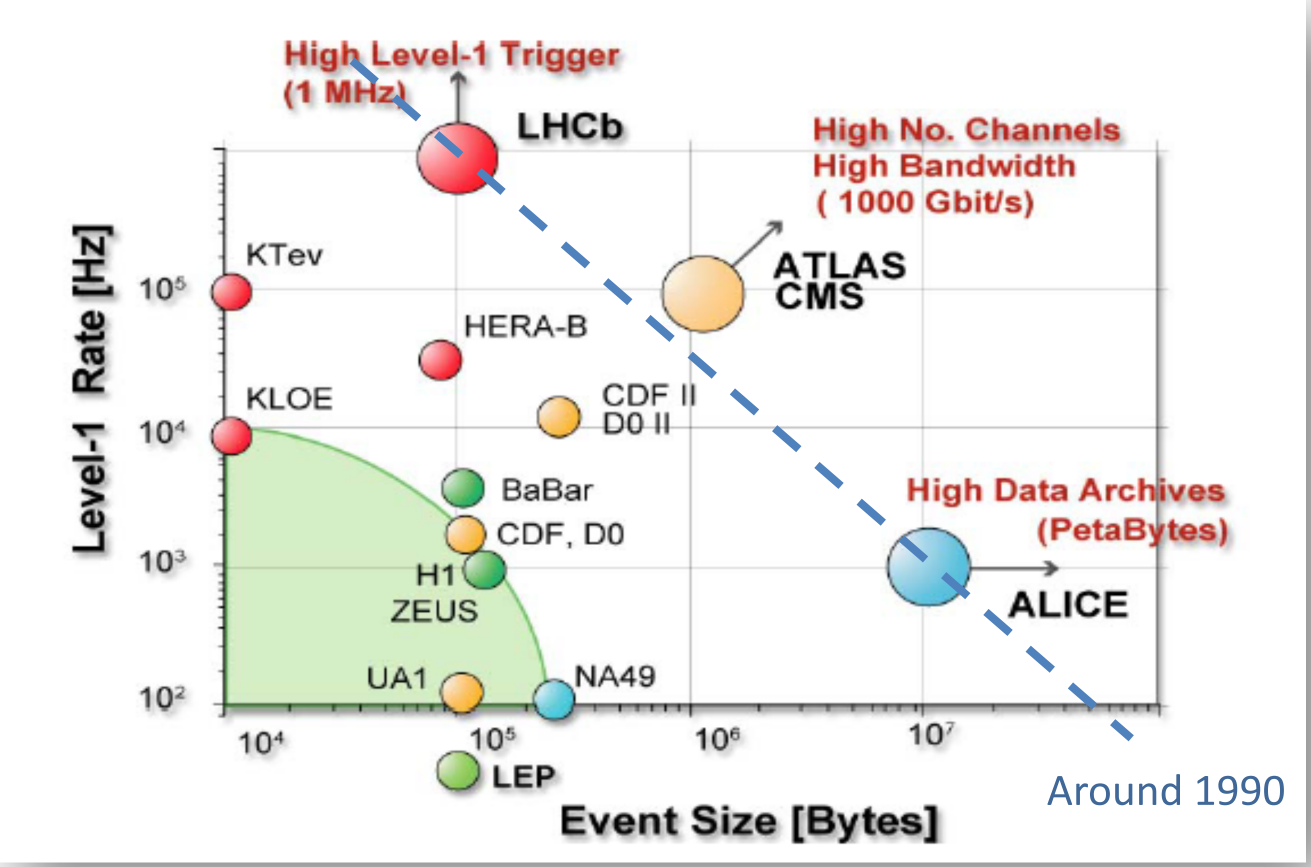
Robustness! Win against the unexpected!

- **Flexibility: to cope changes in conditions and background**
  - Programmable thresholds, high granularity to maintain uniform performance, able to follow changes of luminosity, beam-size and vertex position, able to reach physics results also after 10 years of data taking
- **Redundancy: to make trigger rates independent from the detector and the collider performance**
  - Different backgrounds can change the event shape and dimension, so the result of your trigger selection
- **Selectivity**
  - Good granularity and good resolution of the parameters to ensure rejection of the unwanted background

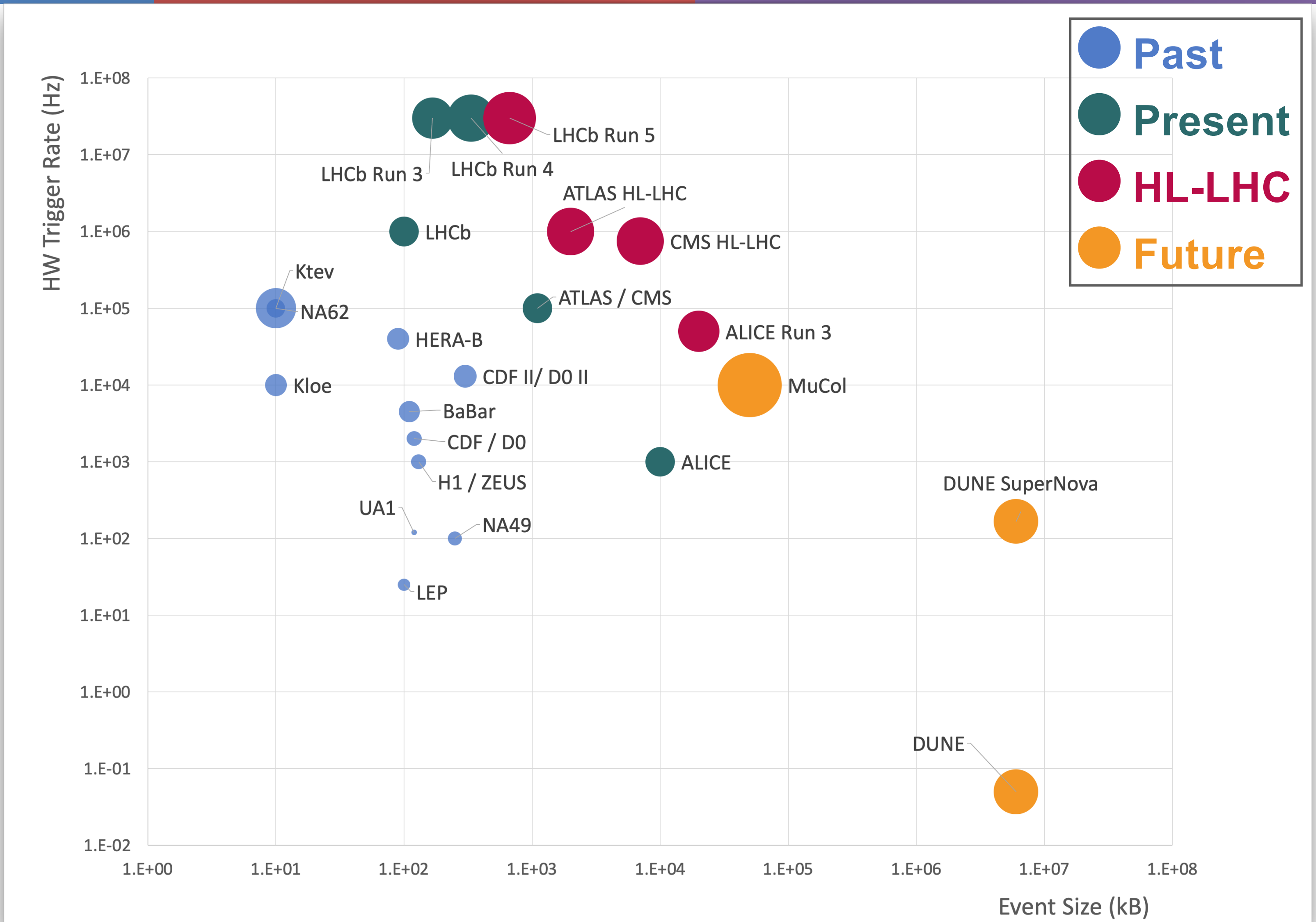
Trigger and data acquisition trends

$$R_{DAQ} = R_T^{max} \times S_E$$

As the data volumes and rates increase, new architectures need to be developed

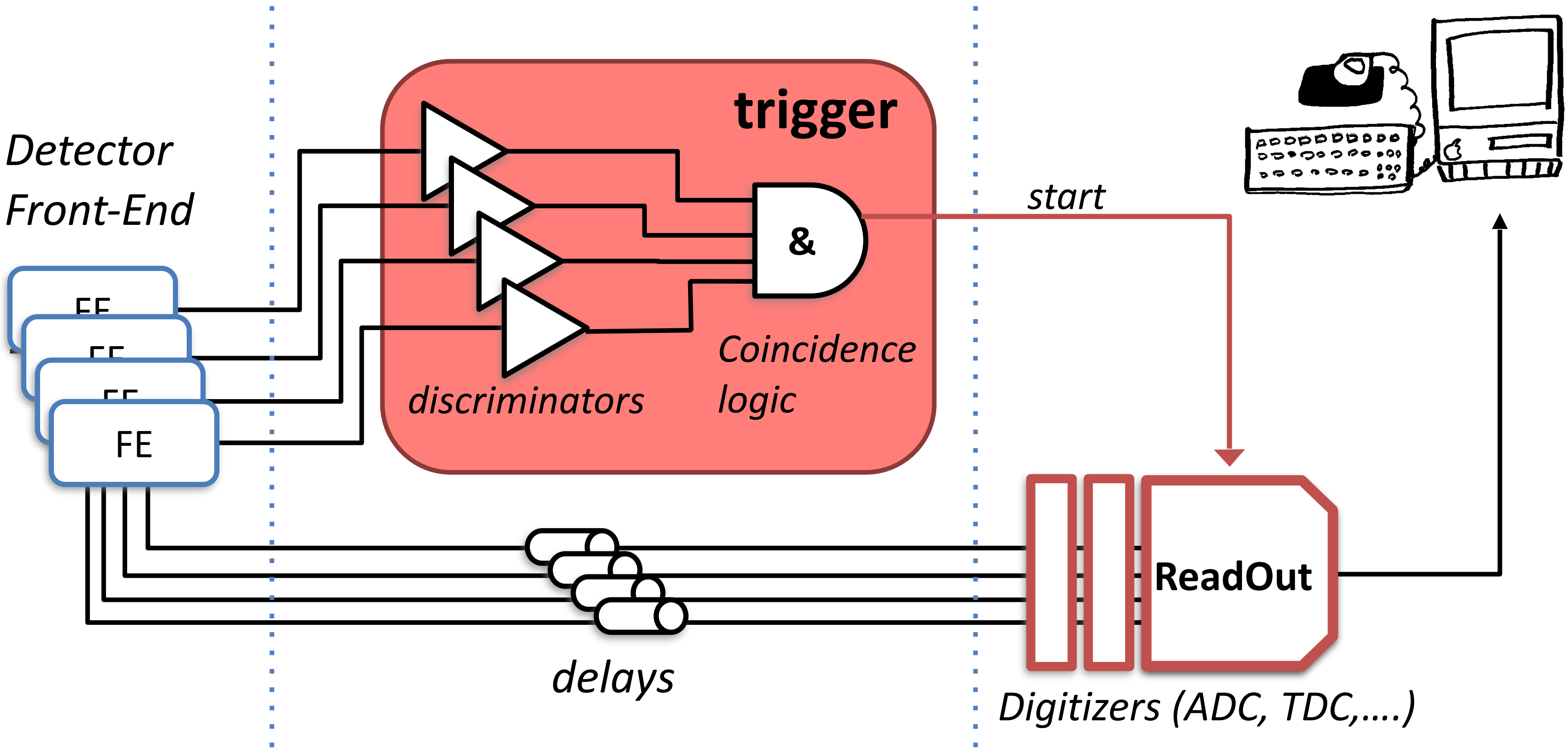


Updated figure



Courtesy of A.Cerri

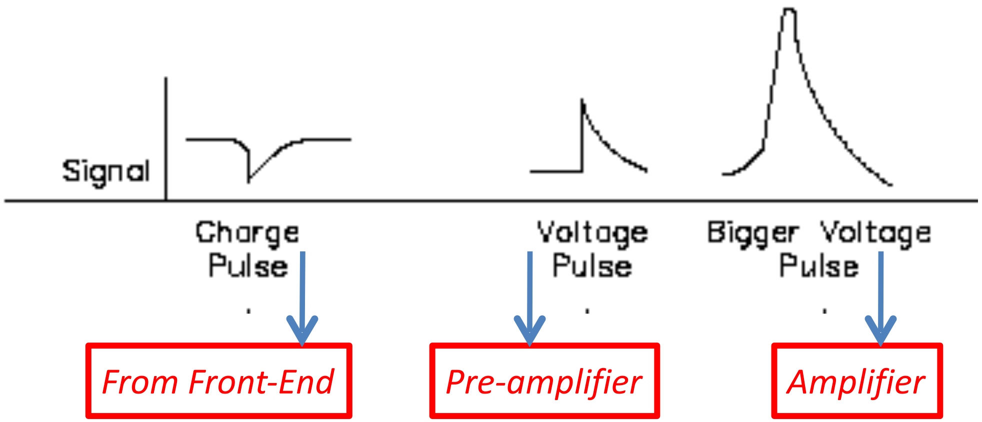
A simple trigger system

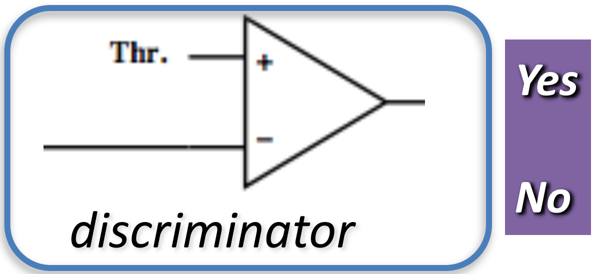


- Chose a sensor/detector specific for your selection
- Dedicate Front-End electronics also used for trigger signals

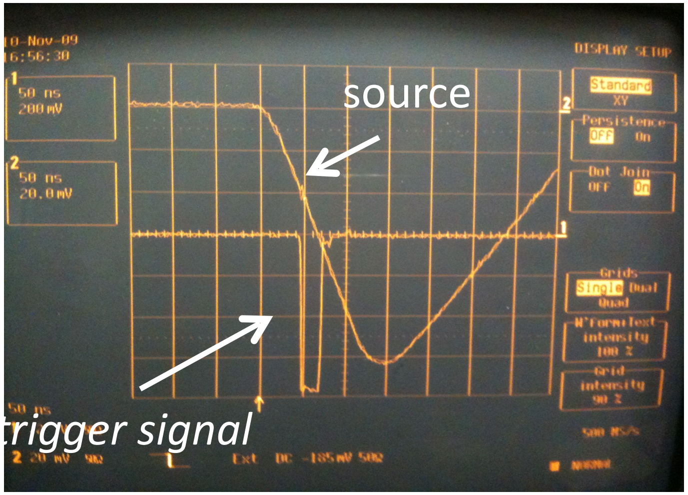
The simplest trigger system

- **Source**: signals from the Front-End of the detectors
  - Binary trackers (pixels, strips)
  - Analog signals from trackers, time of light detectors, calorimeters,….



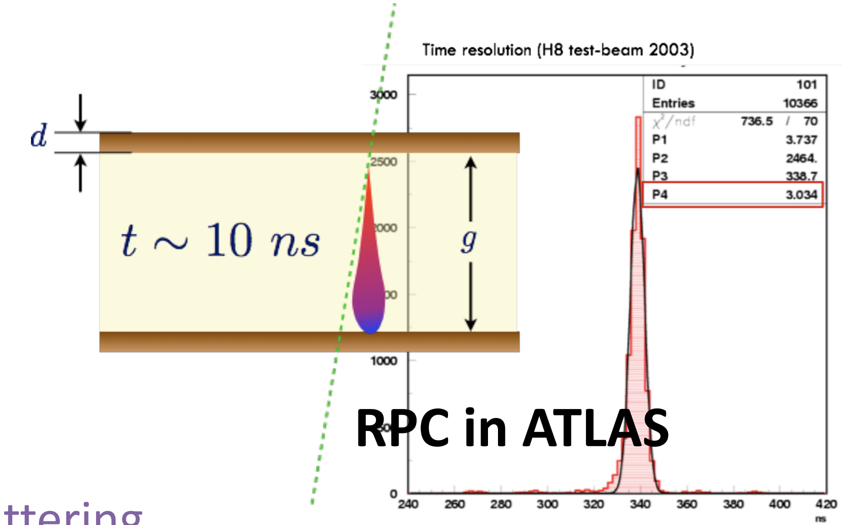


- The simplest trigger is: **apply a threshold**
  - Look at the signal
  - Apply a threshold as low as possible, since signals in HEP detectors have large amplitude variation
  - Compromise between hit efficiency and noise rate



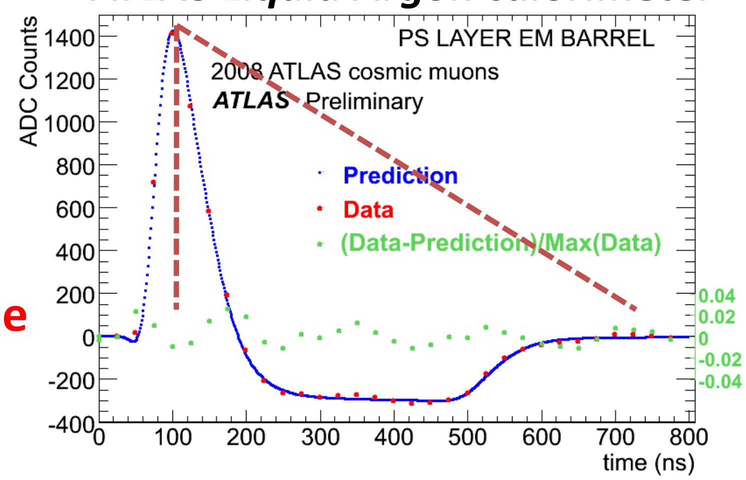
Chose your trigger detector

- Use signals from either existing detectors or dedicated “trigger detectors”
  - Organic scintillators
  - Electromagnetic calorimeters
  - Proportional chambers (short drift)
  - Cathode readout detectors (RPC,TGC,CSC)



- With these requirements
  - **Fast signal**: good time resolution and low jittering
    - Signals from slower detectors are shaped and processed to find the unique peak (peak-finder algorithms)
  - **High efficiency**
  - **(often) High rate capability**

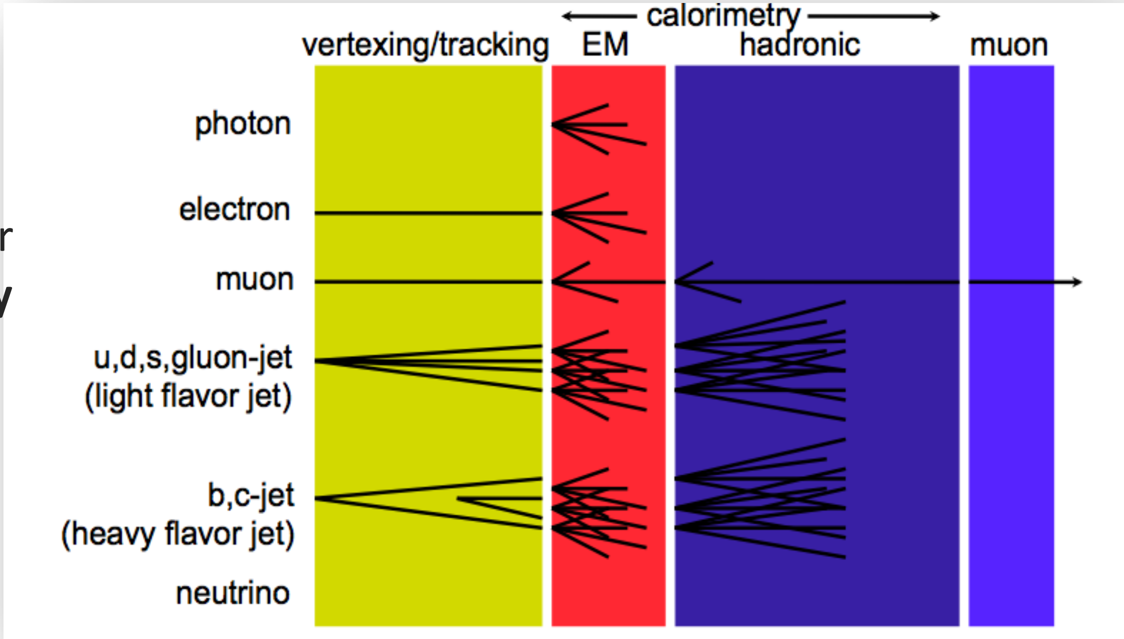
ATLAS Liquid Argon calorimeter



- **Need optimal FE/trigger electronics to process the signal (common design)**

Trigger signatures

- Can collect **many** parameters for discrimination of a given topology
  - Not only the amplitude of a signal
  - More complex quantities by software calculations (also MultiVariate Analysis)
- At <u>first</u>, use intuitive criteria: **be fast and reliable!**
  - Use clear/simple signatures
    - i.e.: apply thresholds on: muon momenta, energy deposits in the calorimeters, good quality tracks in the tracker detectors….

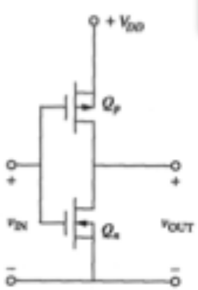


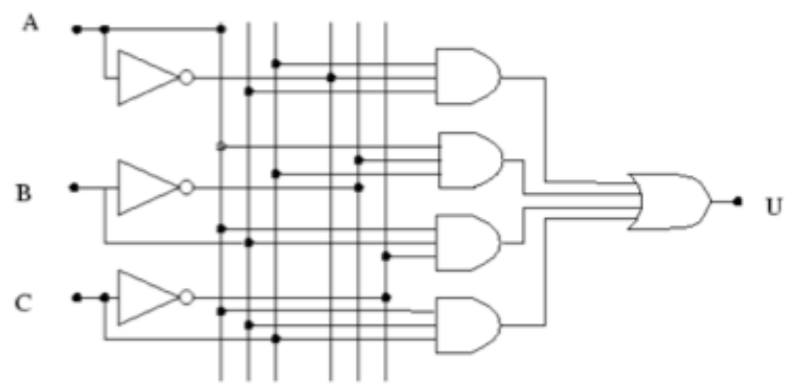
- Eventually combine more signals together following a certain trigger logic (AND/OR), giving **redundancy**

Hardware trigger logic implementation

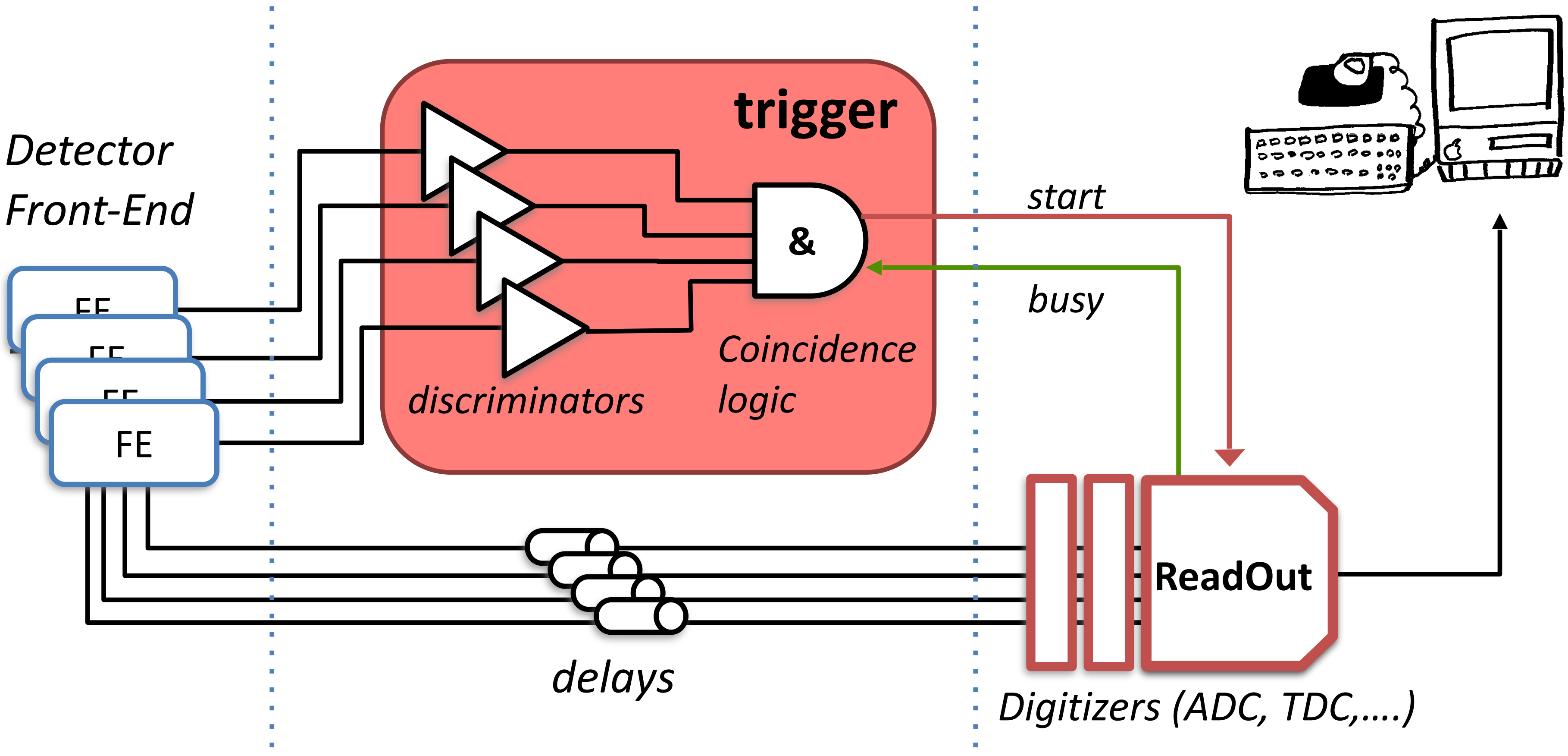
- Analog systems: amplifiers, filters, comparators, ….
- Digital systems:
  - Combinatorial: sum, decoders, multiplexers,….
  - Sequential: flip-flop, registers, counters,….
- Converters: ADC, TDC, …..

Any behaviour of our system can be described logically with a sequence of mathematical operators





A simple trigger system



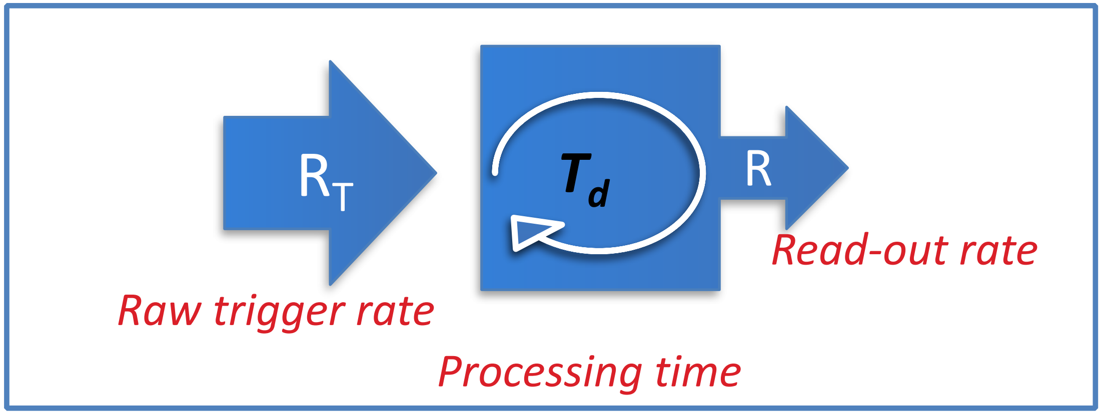
- Due to **fluctuations**, the incoming rate can be higher than the processing one
- Valid signals can be rejected due to system **busy**

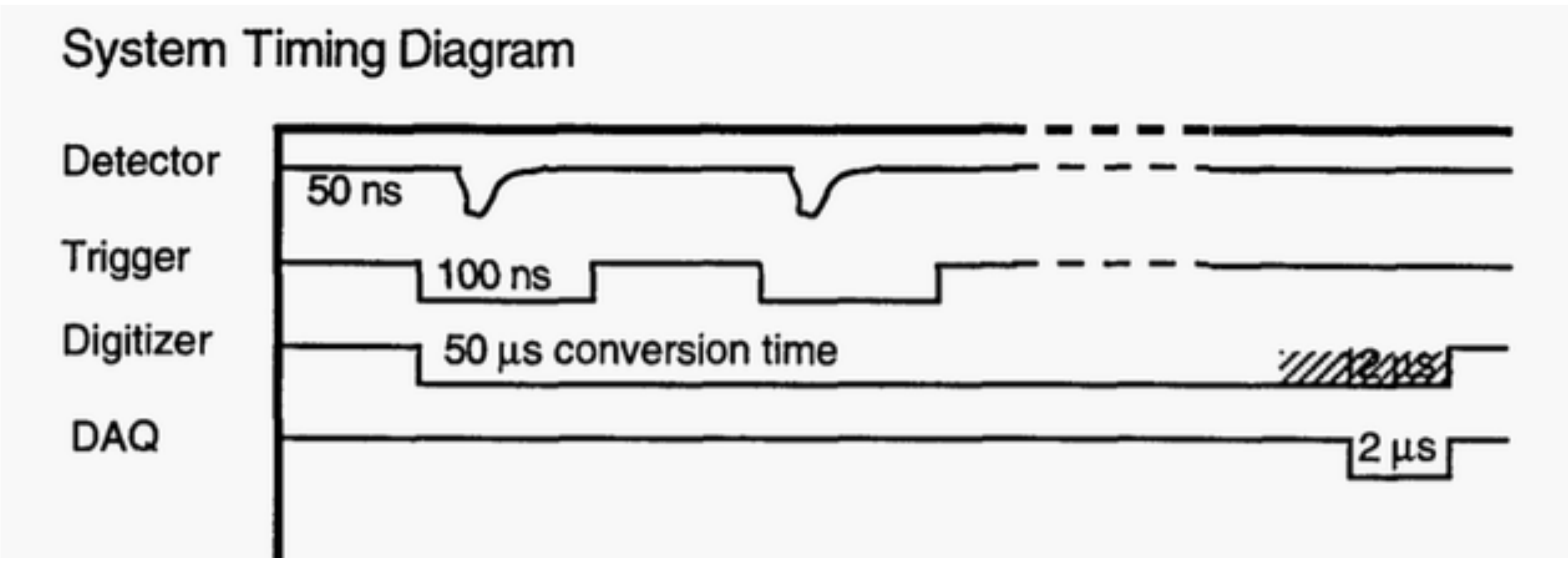
Dead-time

- The most important parameter in designing high speed **T/DAQ systems**
  - The fraction of the acquisition time in which no events can be recorded. It can be typically of the order of **few %**
- Occurs when a given step in the processing takes a **finite amount of time**
  - Readout dead-time
  - Trigger dead-time
  - Operational dead-time

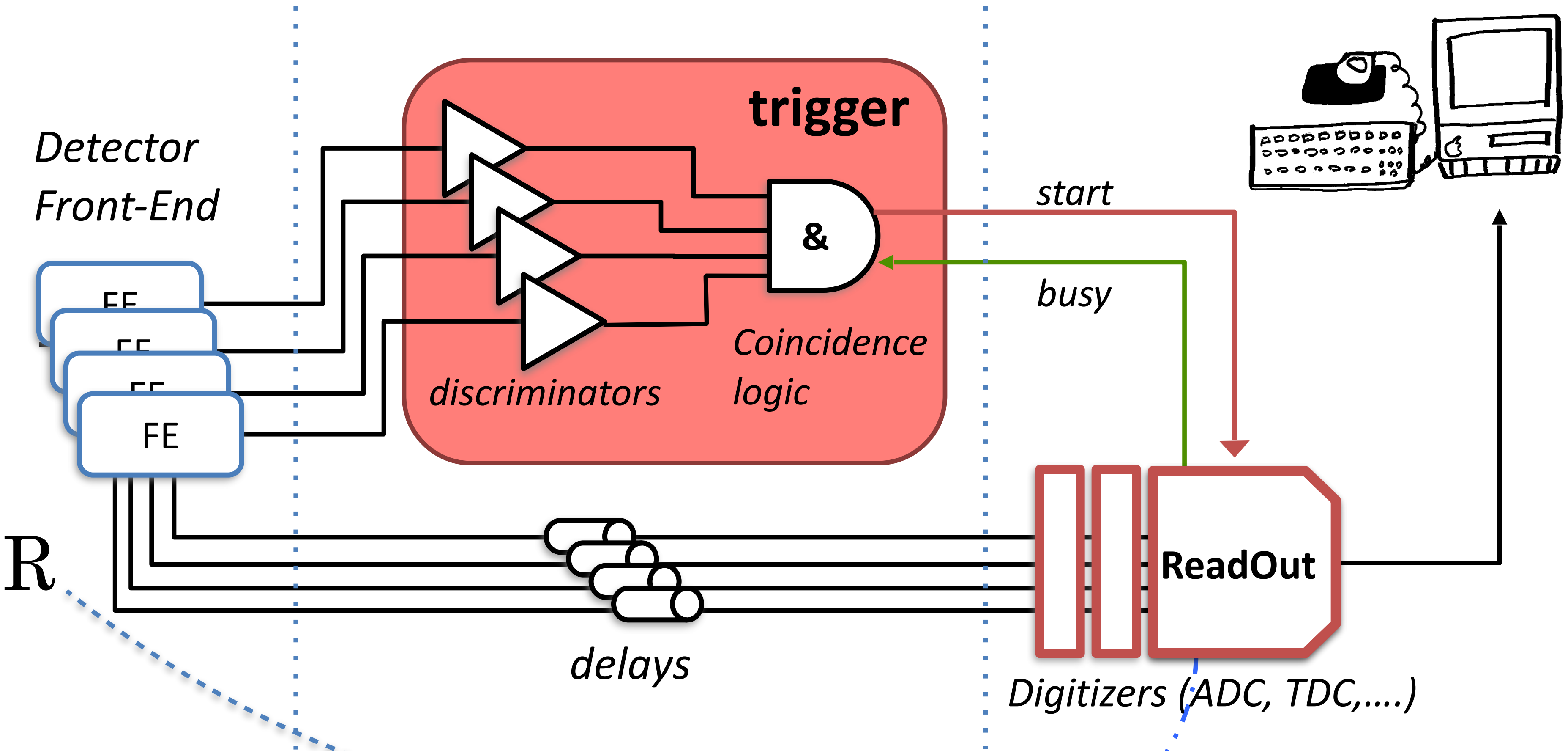
Affects efficiency!

- **Fluctuations produce dead-time!**




System Timing Diagram
Detector
50 ns
Trigger
100 ns
Digitizer
50 µs conversion time
DAQ
2 µs

A simple trigger system



$$D_t = R \cdot T_{RO}$$

Fraction of lost events due to finite readout

Dead-time and efficiency

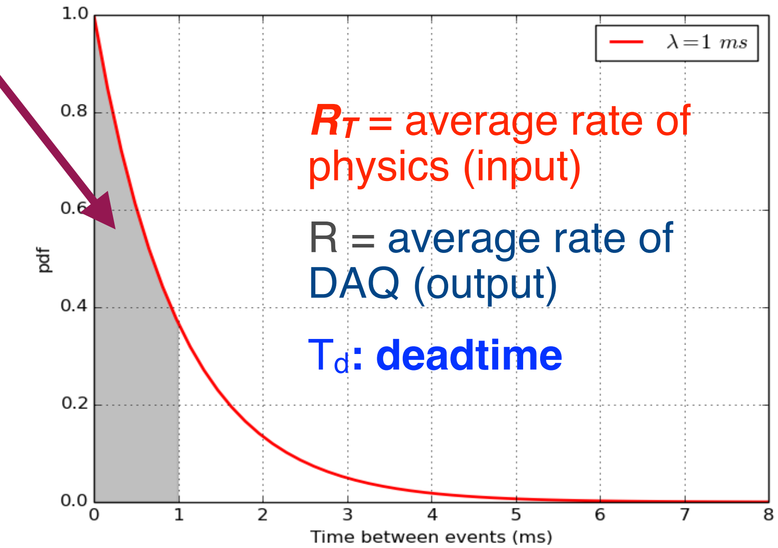
What if a trigger is created when the system is busy?

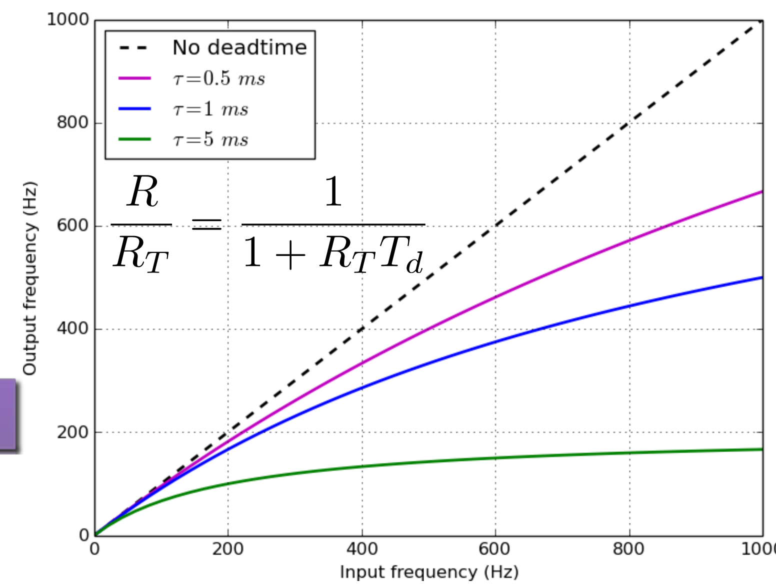
- The **busy** mechanism protects our electronics from unwanted triggers
  - During the busy time, no signals are accepted, cause of **inefficiency**
  - source of **dead-time**
- Due to **stochastic fluctuations**
  - DAQ rate always < physics rate
  - Efficiency always < 100%
- To cope with the input signal fluctuations, we have to over-design our DAQ system
  - **can we mitigate this effect?**

The trick is to make both R_T and T_d as small as possible ($R \sim R_T$)

FAST TRIGGER!

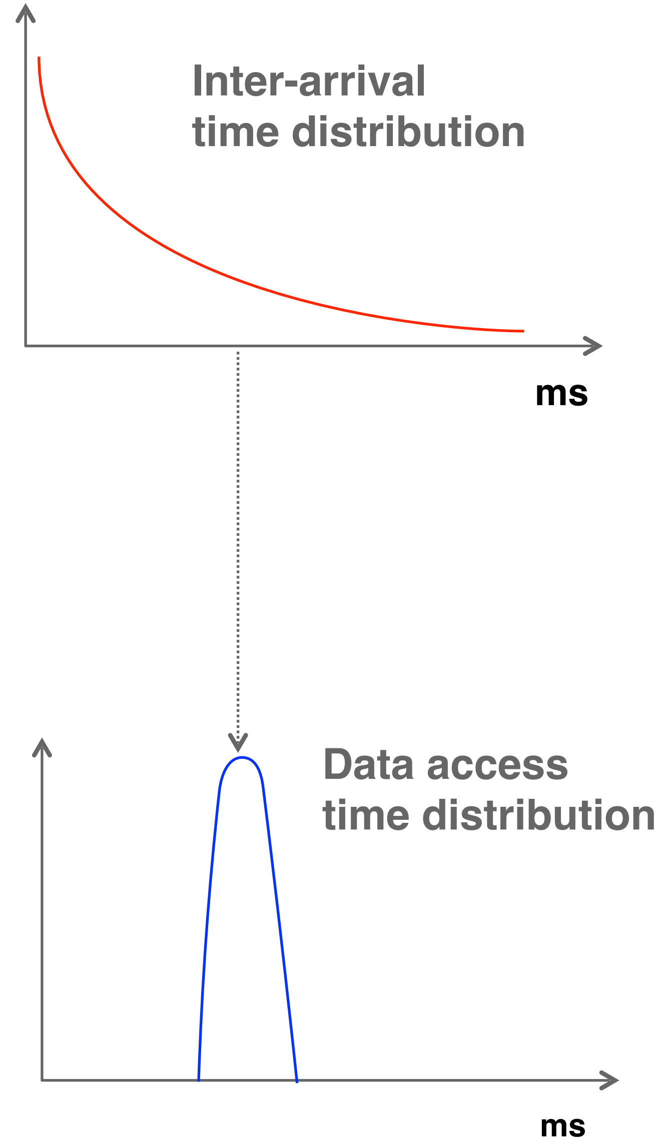
LOW INPUT RATE!





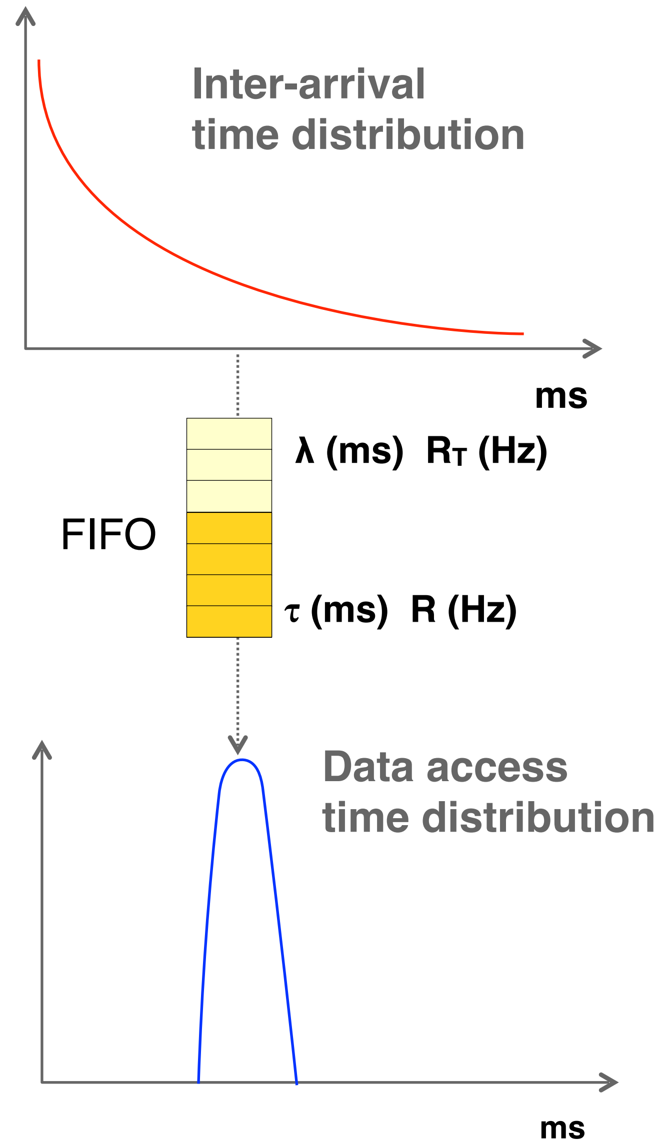
De-randomization

- What if we were able to make the system more deterministic and less dependent on the arrival time of our signals?
  - Then we could ensure that events don't arrive when the system is busy
  - This is called **de-randomization**
- How can be achieved?
  - by **buffering** the data (having a holding queue where we can slot it up to be processed)
  - Maintaining $\tau \sim \lambda$ (traffic intensity), high efficiency can be obtained even with moderate depth of FIFOs



De-randomization

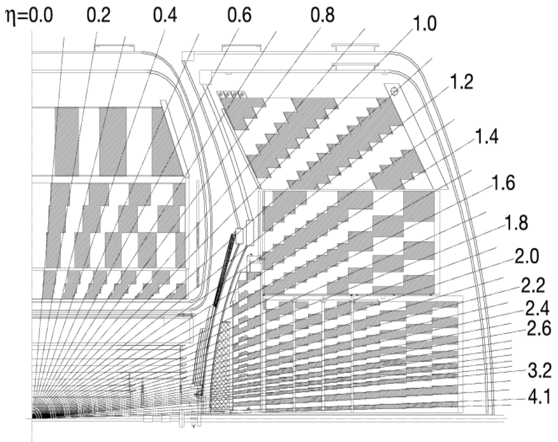
- What if we were able to make the system more deterministic and less dependent on the arrival time of our signals?
  - Then we could ensure that events don't arrive when the system is busy
  - This is called **de-randomization**
- How can be achieved?
  - by **buffering** the data (having a holding queue where we can slot it up to be processed)
  - Maintaining $\tau \sim \lambda$ (traffic intensity), high efficiency can be obtained even with moderate depth of FIFOs



How to minimise dead-time….

- **1: Parallelism**
  - Independent readout and trigger processing paths, one for each sensor element
  - Digitisation and DAQ processed in parallel (as many as affordable!)

Segment as much as you can!

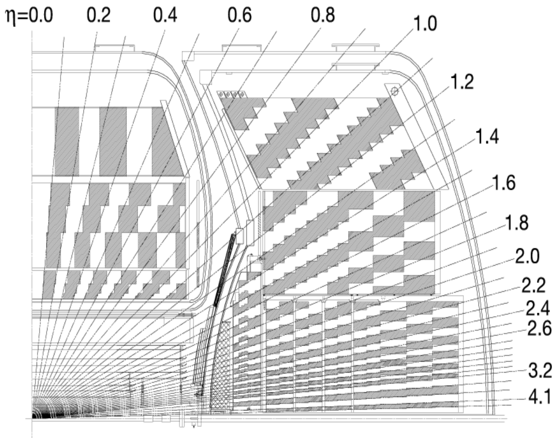


DZero calorimeters showing the transverse and longitudinal segmentation pattern

How to minimise dead-time….

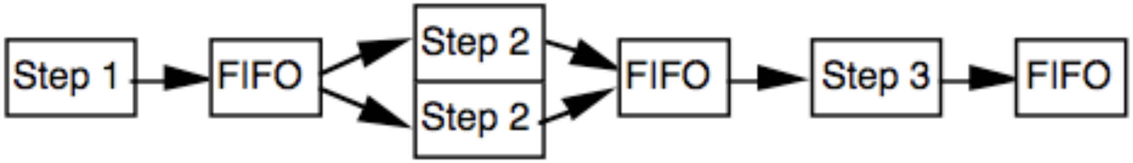
- **1: Parallelism**
  - Independent readout and trigger processing paths, one for each sensor element
  - Digitisation and DAQ processed in parallel (as many as affordable!)

Segment as much as you can!



DZero calorimeters showing the transverse and longitudinal segmentation pattern

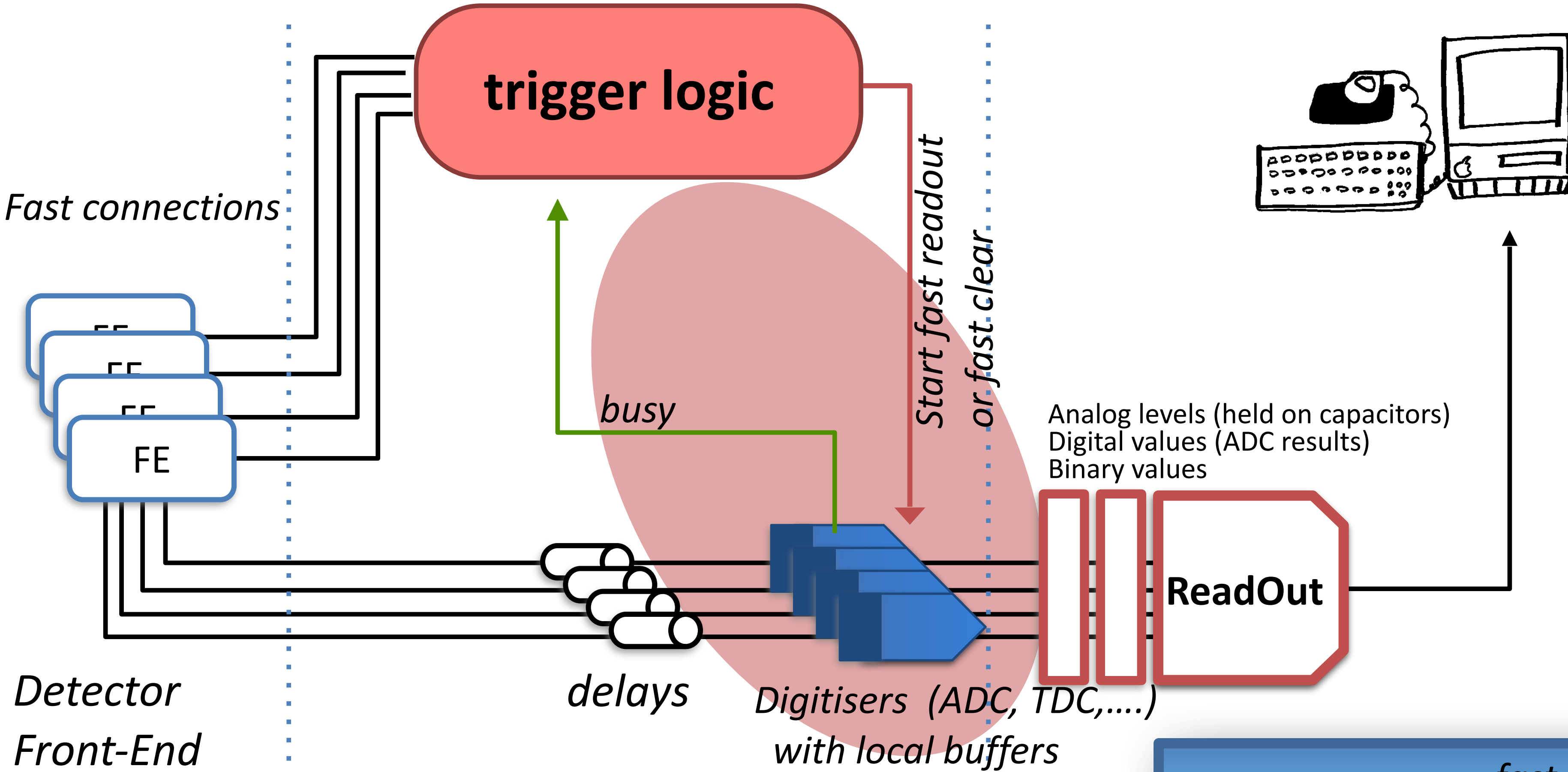
- **2: Pipeline processing with intermediate buffers,** to absorb fluctuations
  - Organise the process in different steps
  - Use local **buffers** between steps with different timing



$$\frac{R}{R_T} = \frac{1}{1 + R_T T_d}$$

Try to absorb in capable buffers

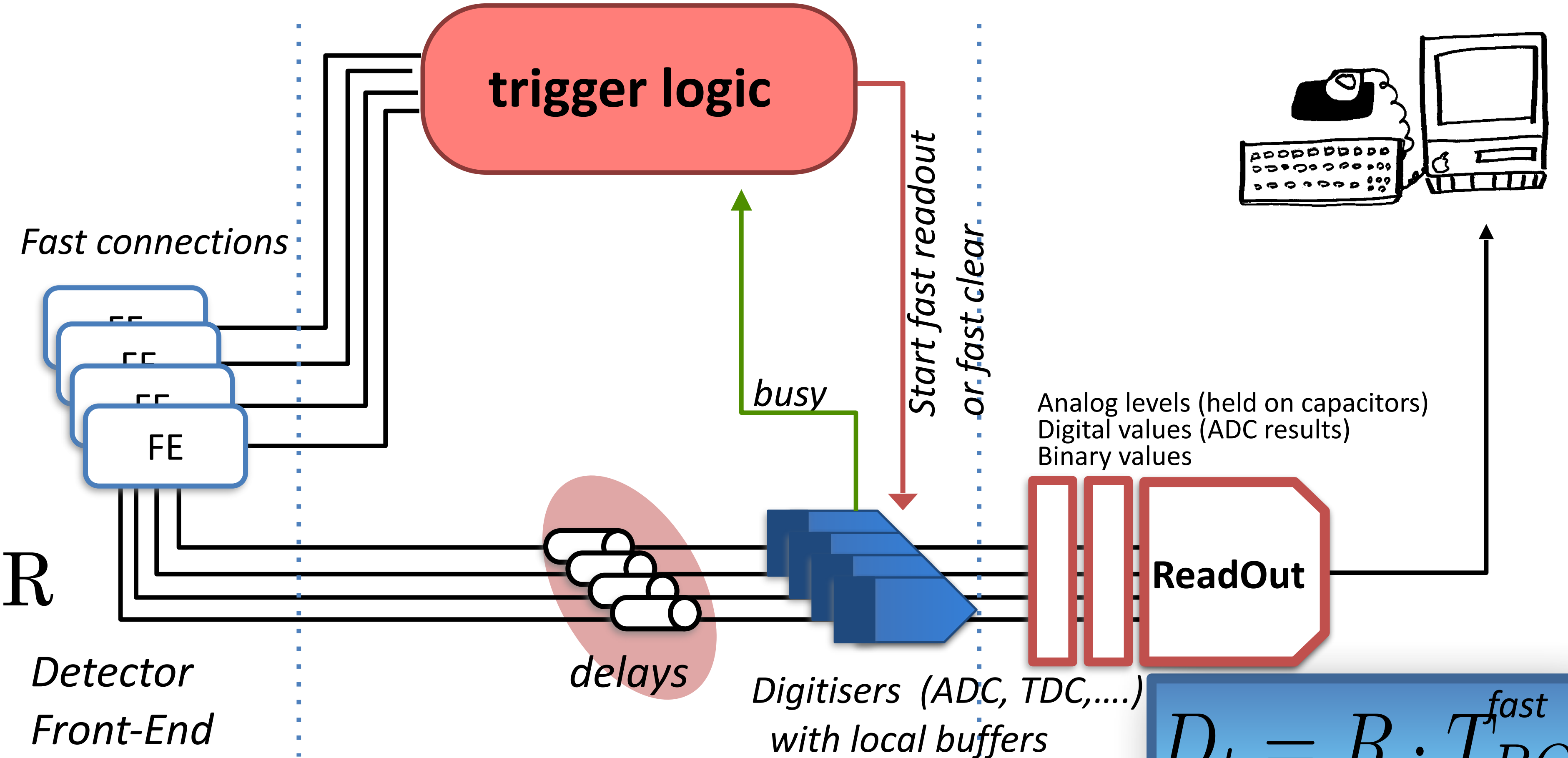
Minimising readout dead-time...



$$D_t = R \cdot T_{RO}^{fast}$$

- **Parallelism**: Use multiple digitisers
- **Pipelining**: Different stages of readout: fast local readout + global event readout (slow)

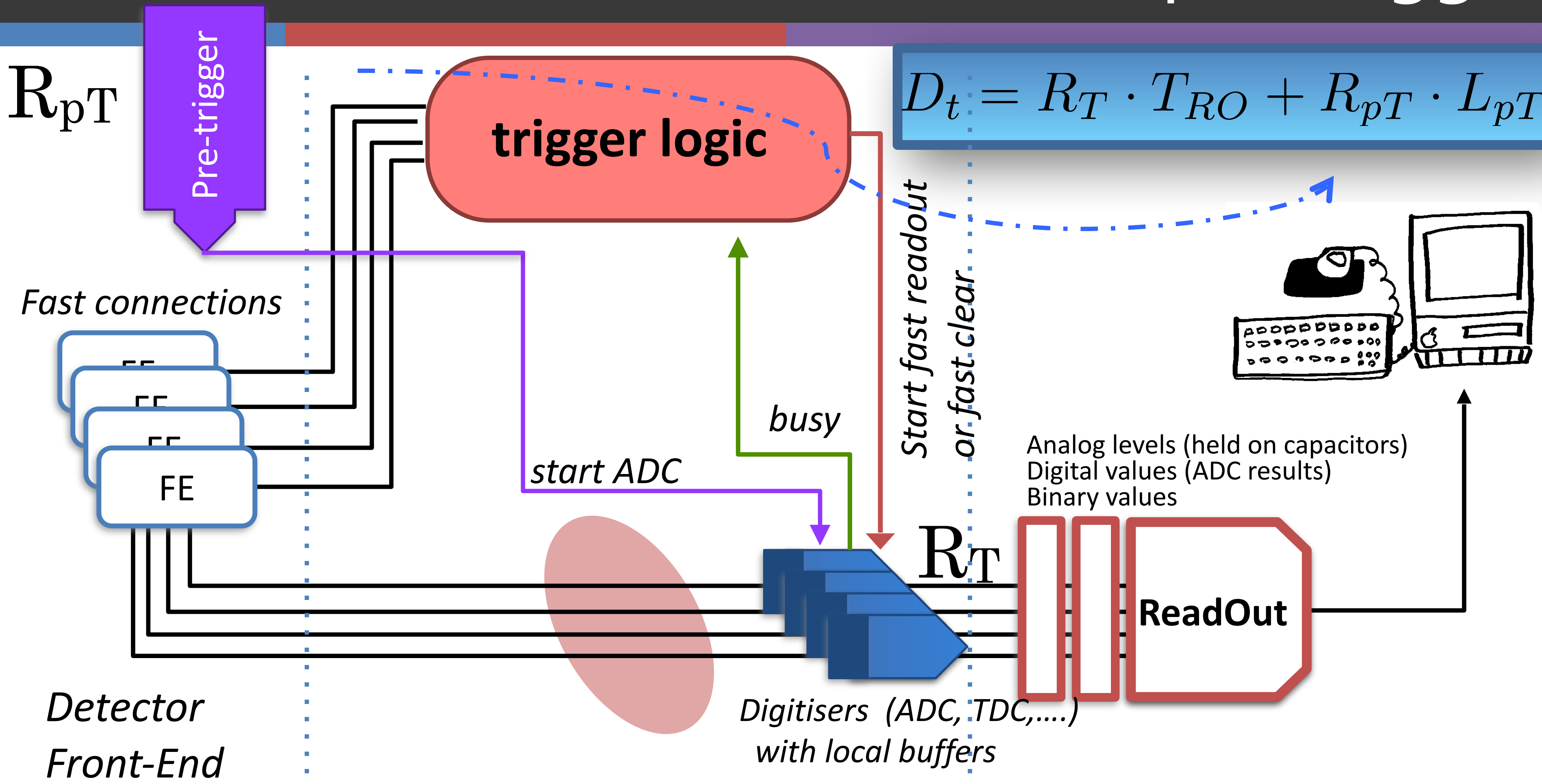
Trigger latency



$$D_t = R \cdot T_{RO}^{fast}$$

- **Time to form the trigger decision and distribute to the digitisers**
- Signals are delayed until the trigger decision is available at the digitisers
  - But more complex is the selection, longer is the latency

Add a pre-trigger



- Add **a very fast** first stage of the trigger, signalling the presence of minimal activity in the detector
  - **START the digitisers**, when signals arrive
  - The main trigger decision comes later (after the digitisation) -> can be more complex

Coupling rates and latencies

- Extend the idea… more levels of trigger, each one reducing the rate, even with longer latency

- Dead-time is the sum of the trigger dead-time, summed over the trigger levels, and the readout dead-time

$$\left(\sum_{i=2}^{N} R_{i-1} \times L_i\right) + R_N \times T_{\mathrm{LRO}}$$

i=1 is the pre-trigger

R_i = Rate after the i-th level

L_i = Latency for the i-th level

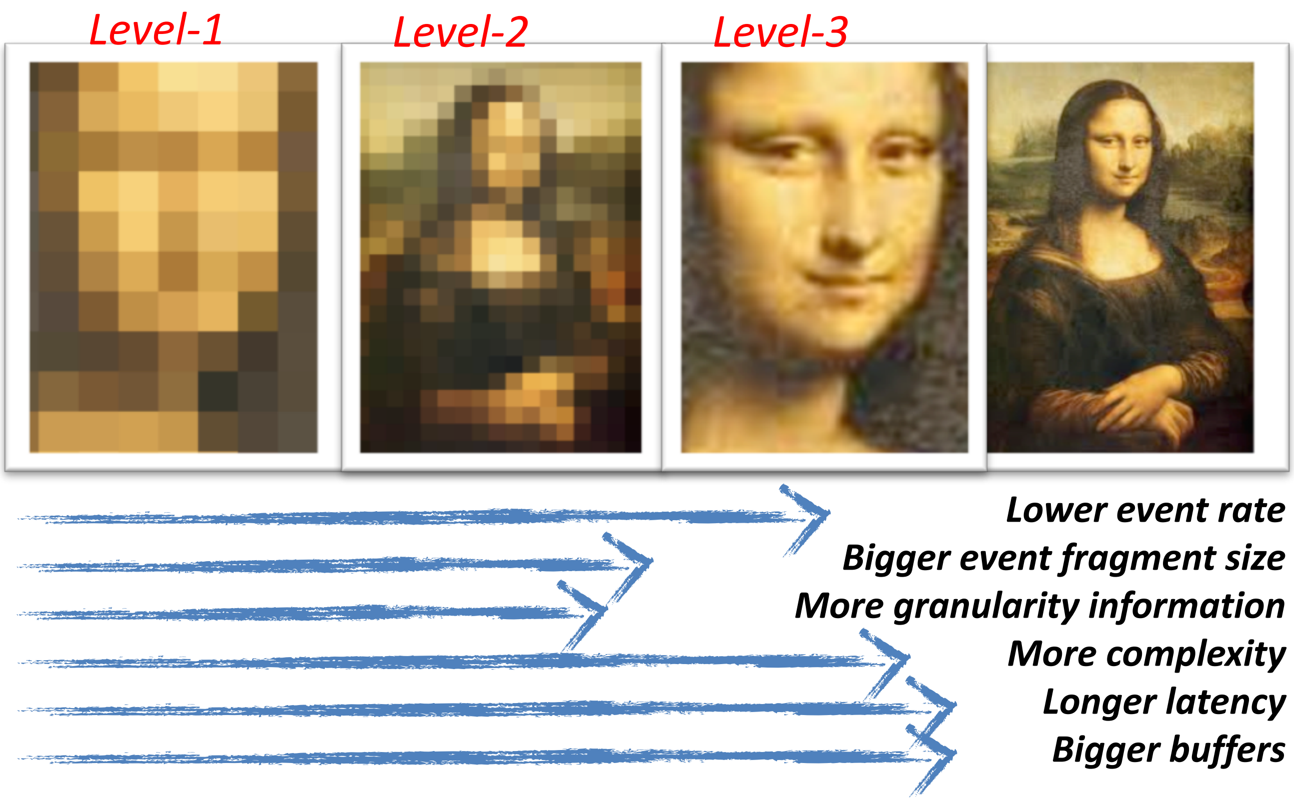
T_{LRO} = Local readout time

Readout dead-time is minimum if its input rate R_N is low!

Try to minimise each factor!

Multi-level triggers

- Adopted in large experiments with large data volume
- Successively more complex decisions are made on successively lower data rates
  - First level with short latency, working at higher rates
  - Higher levels apply further rejection, with longer latency (more complex algorithms)

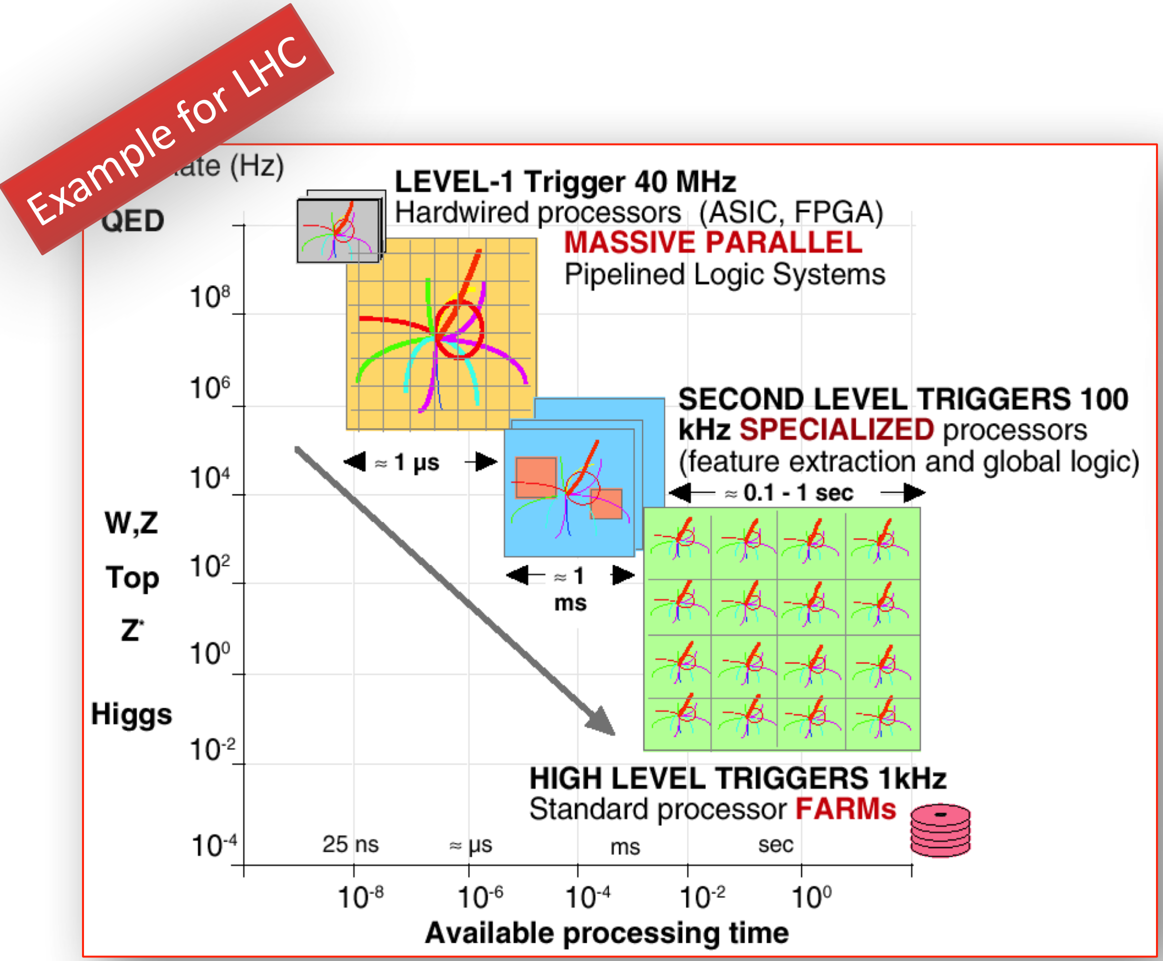


LHC experiments @ Run1

Exp.	N.of Levels
ATLAS	3
CMS	2
LHCb	3
ALICE	4

Efficiency for the desired physics must be kept high at all levels, since rejected events are lost for ever

Use of multi-level trigger



L1: Inclusive trigger

L2: Confirm L1, inclusive and semi-incl., simple topology, vertex rec.

L3: Confirm L2, more refined topology selection, near offline

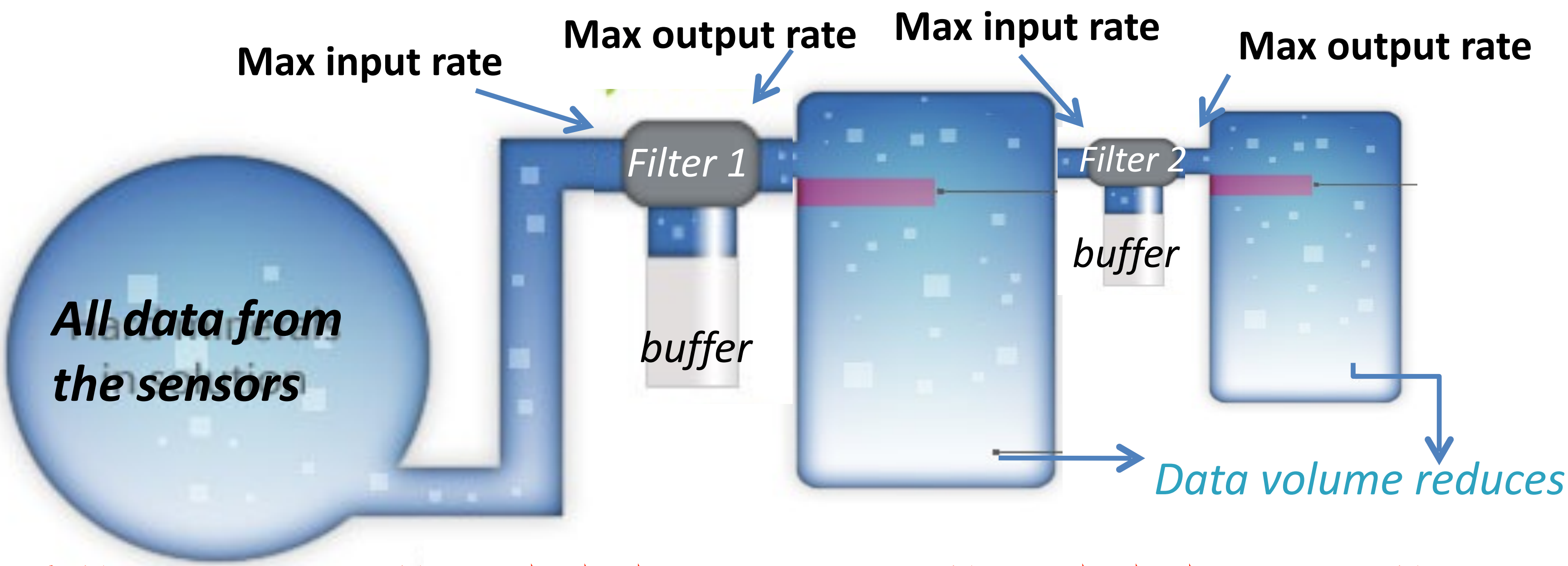
Architectural view ***Logical view***

Buffering and filtering

- **At each step**, data volume is reduced, more refined filtering to the next step
  - The input rate defines the filter **processing time** and its **buffer size**
  - The output rate limits the maximum latency allowed in the **next step**
  - Filter power is limited by the capacity of the next step

As long as the buffers do not fill up (overflow), no additional dead-time is introduced!

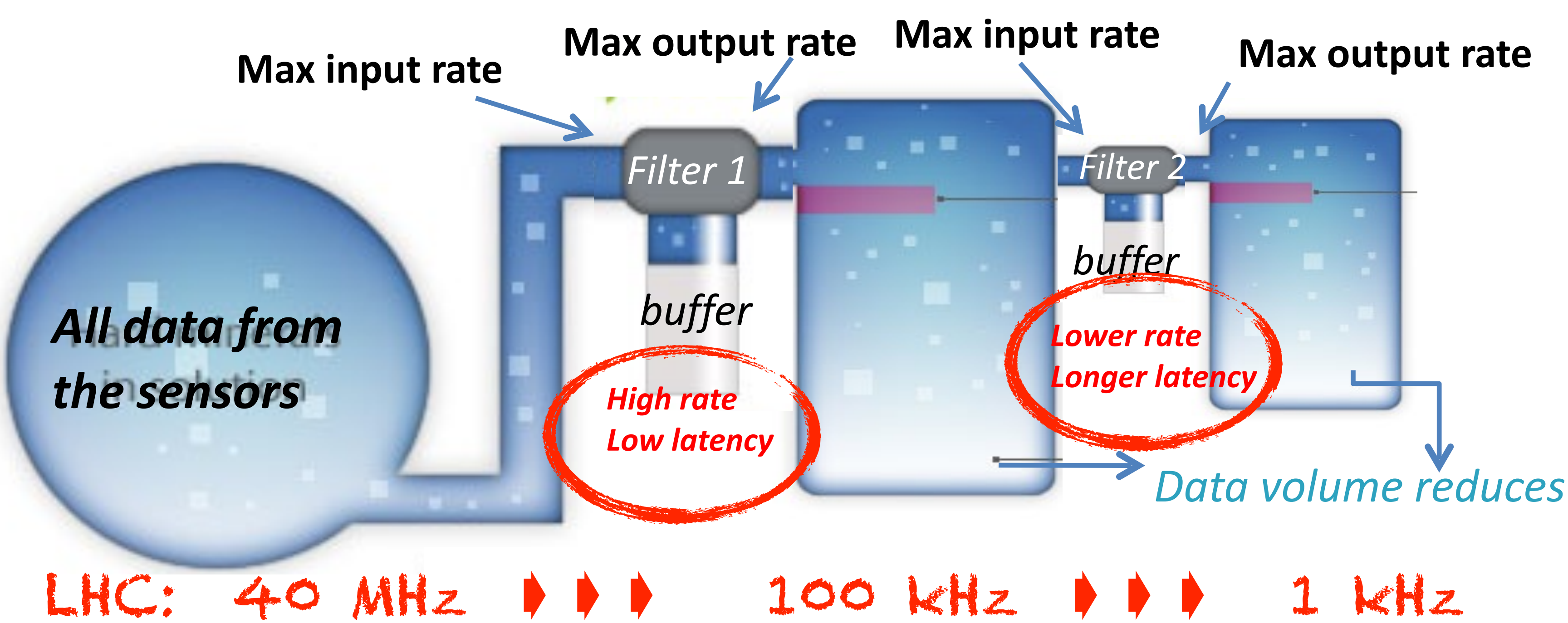
BUSY signal is still needed



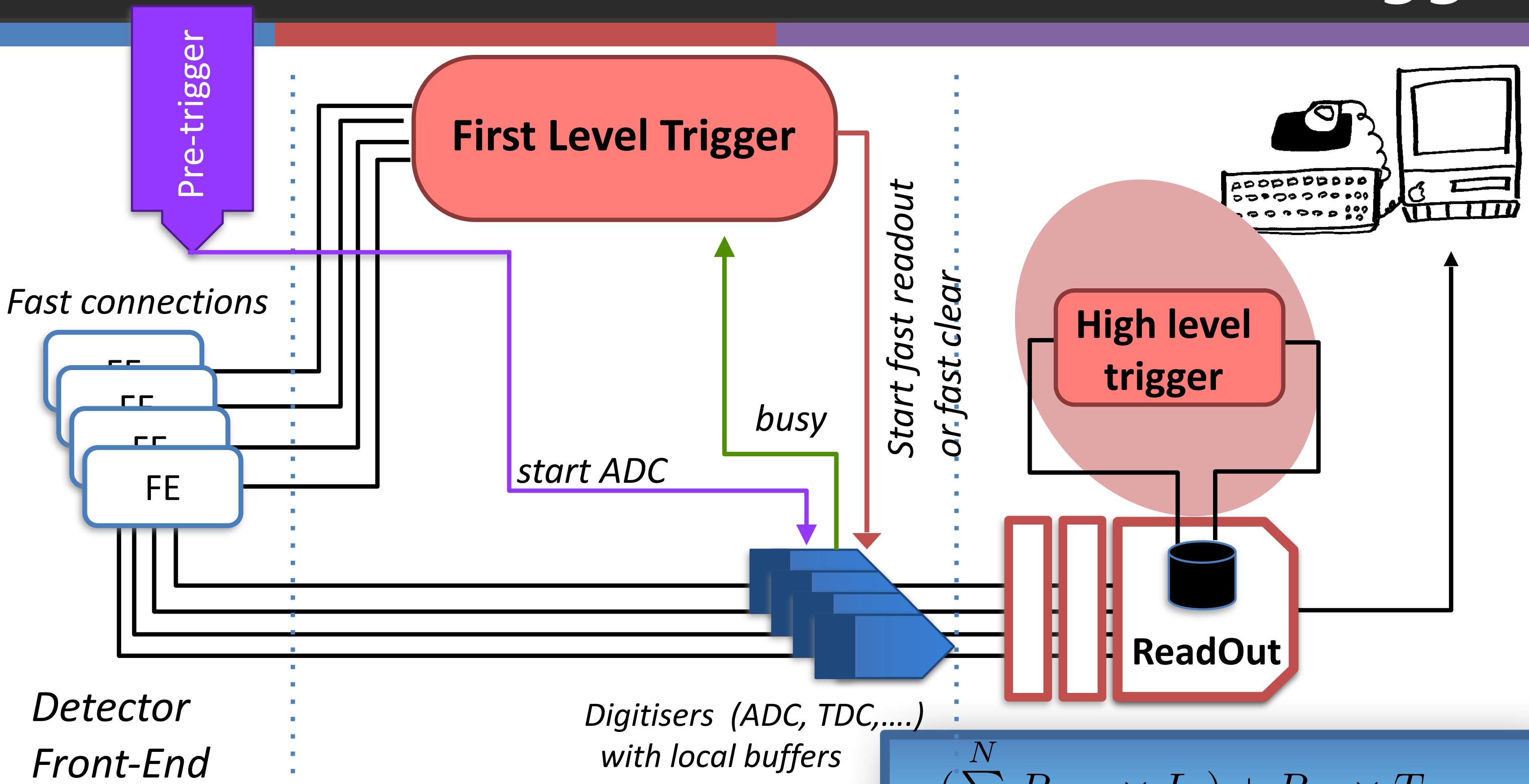
LHC: 40 MHz ▶▶▶ 100 kHz ▶▶▶ 1 kHz

Rates and latencies are strongly connected

- If the rate after filtering is higher than the capacity of the next step
  - Add filters (tighten the selection)
  - Add better filters (more complex selections)
  - Discard randomly (pre-scales)



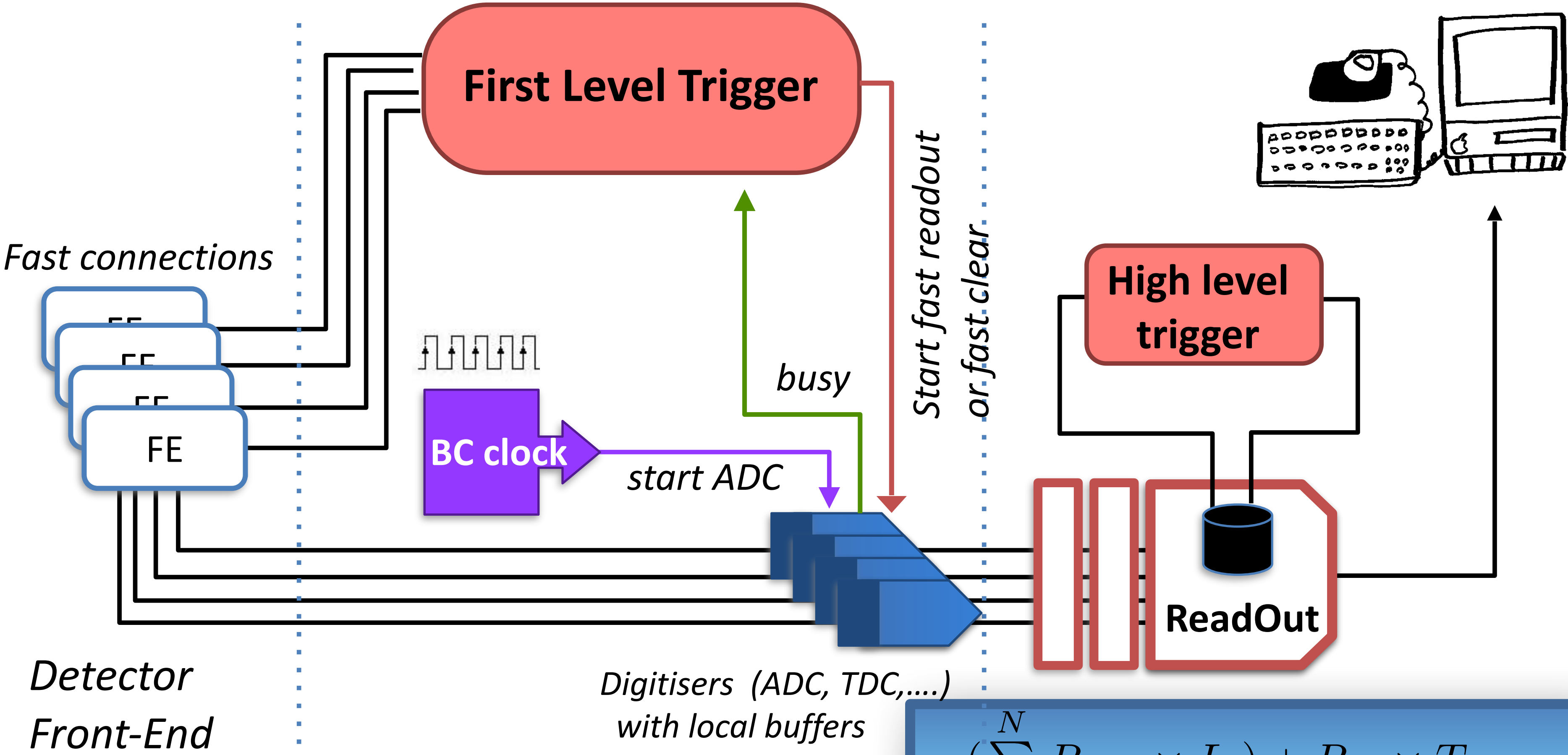
Schema of a multi-level trigger



$$(\sum_{i=2}^{N} R_{i-1} \times L_i) + R_N \times T_{LRO}$$

- Different levels of trigger, accessing different buffers
- The pre-trigger starts the digitisation

Schema of a multi-level trigger @ colliders



$$(\sum_{i=2}^{N} R_{i-1} \times L_i) + R_N \times T_{LRO}$$

- The BC clock can be used as a pre-trigger
  - First-level trigger is **synchronous** to the collision clock: can use the time between two collisions to make its decision, without dead-time

Three types of trigger

Three types of trigger

- **Global:**
  - an external system identifies the “interesting” event, all the readout data is collected for that **event identifier**

Three types of trigger

- **Global:**
  - an external system identifies the “interesting” event, all the readout data is collected for that **event identifier**
- **Local:**
  - local trigger decision to readout data on the **local front-end modules**, readout collects fragments corresponding to that trigger

Three types of trigger

- **Global:**
  - an external system identifies the “interesting” event, all the readout data is collected for that **event identifier**
- **Local:**
  - local trigger decision to readout data on the **local front-end modules**, readout collects fragments corresponding to that trigger
- **Continuous readout:**
  - front-end sends data continuously to the readout, at a fixed rate, regardless the data content. **Data size and rate are constant in size**. Readout cannot group fragments relative to an event

By Rick Harrison (license)

not really a photo, almost a movie

Three types of trigger

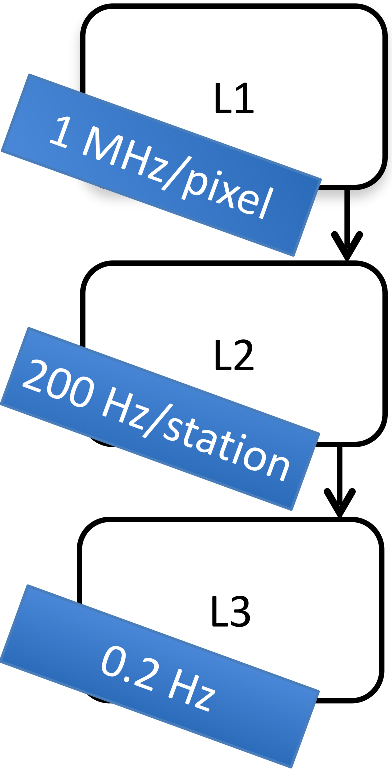
- **use cases:**
  - **Colliders**: normally use global trigger: if something interesting has been seen somewhere, take all the data corresponding to that bunch crossing
  - **Large distributed telescopes**: often use local trigger: readout data for the portions of the detector that have seen something
  - **Very slow detectors**: sometimes use continuous readout: sample the analogue signals at a fixed rate and let the downstream DAQ decide whether there were any interesting signals

By Rick Harrison (license)

not really a photo, almost a movie

Simple signatures: Auger observatory

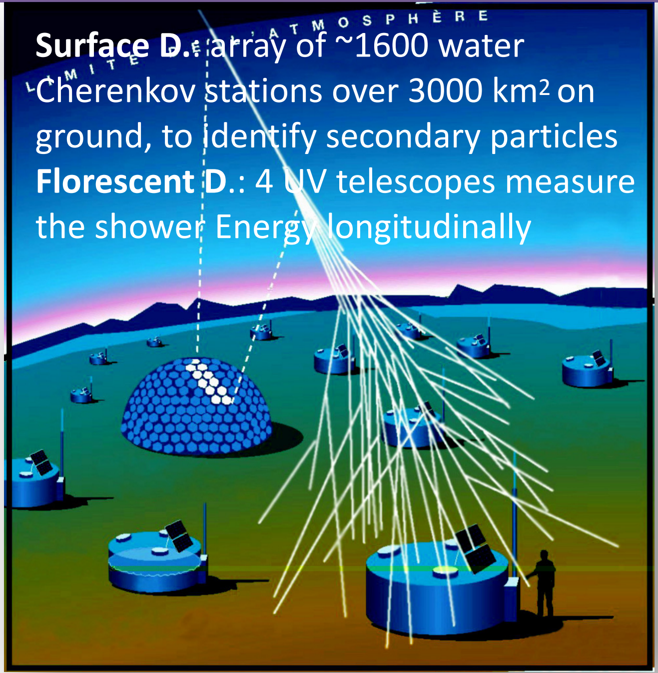
- **Detect air showers generated by cosmic rays above 10^{17} eV**
  - Expected rate $< 1/km^2$/century. Two large area detectors
- On each detector, a 3-level trigger operates at a wide range of primary energies, for both vertical and very inclined showers

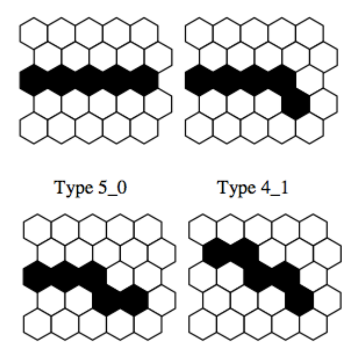


L1: (local) decides the pixel status (on/off)
- ADC counts > **threshold**
- ADC with 100 ns (**time** resolution)
- ADC values stored for **100 μs in buffers**
- **Synchronised** with a signal from a GPS clock

L2: (local) identifies track segments
- Geometrical criteria with recognition algorithms on programmable patterns

L3: (central) makes spatial and temporal correlation between L2 triggers



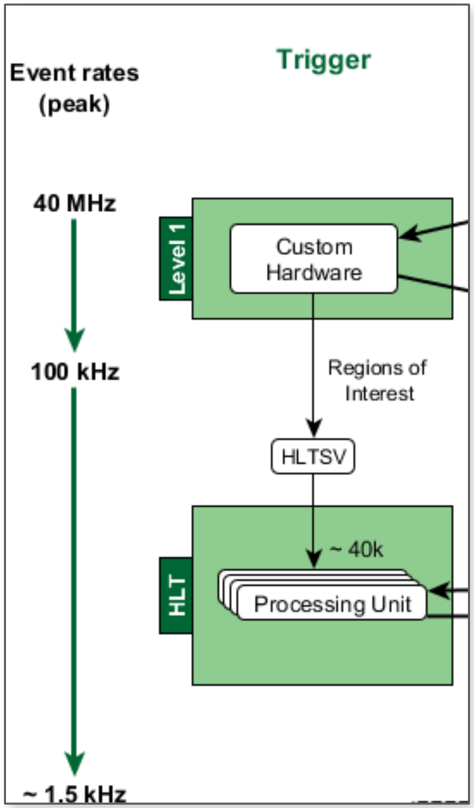


Example of L2 patterns

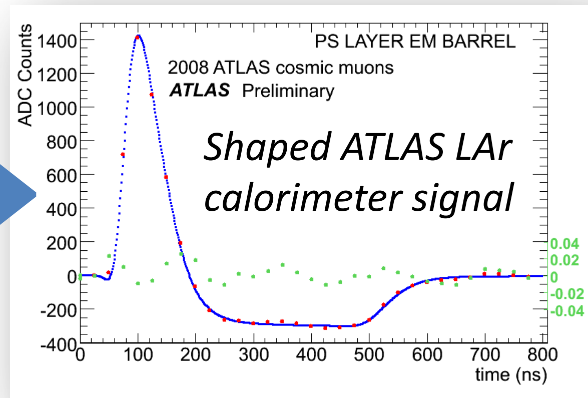
One event ~ 1MB → 0.2 MB/s bandwidth for the DAQ system

Multiple signatures: ATLAS calorimeter trigger

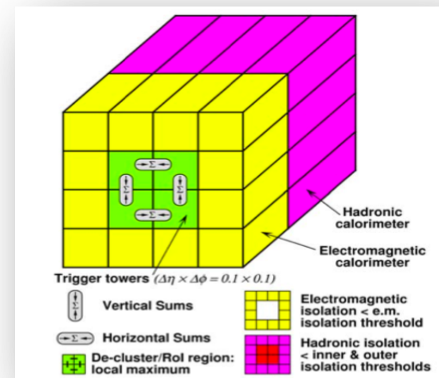
- **Identify high energy e, γ, τ, jets, missing E_T, ΣE_T**

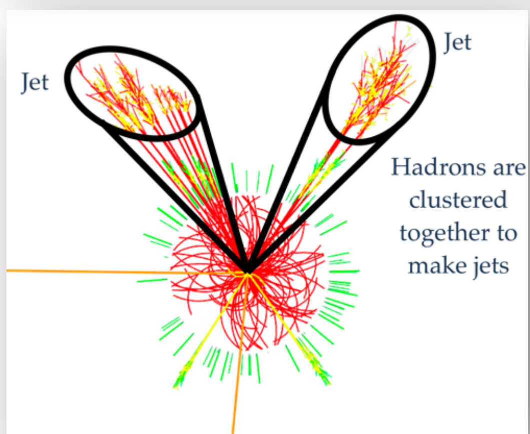


- **Dedicated Front-End electronics**
  - Each cell sends shaped analog signals



- **Level-1 hardware trigger (2.5 us)**
  - **ASICs** for simple cluster algorithms, with programmable E_T thresholds

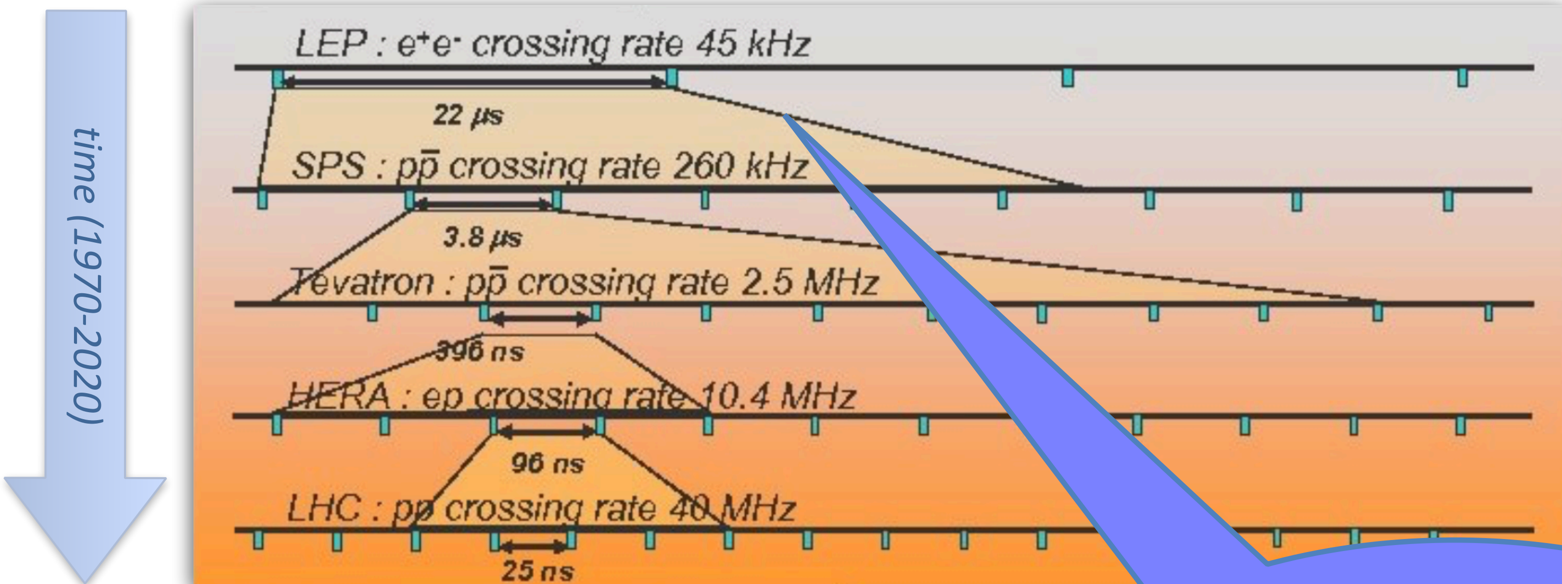




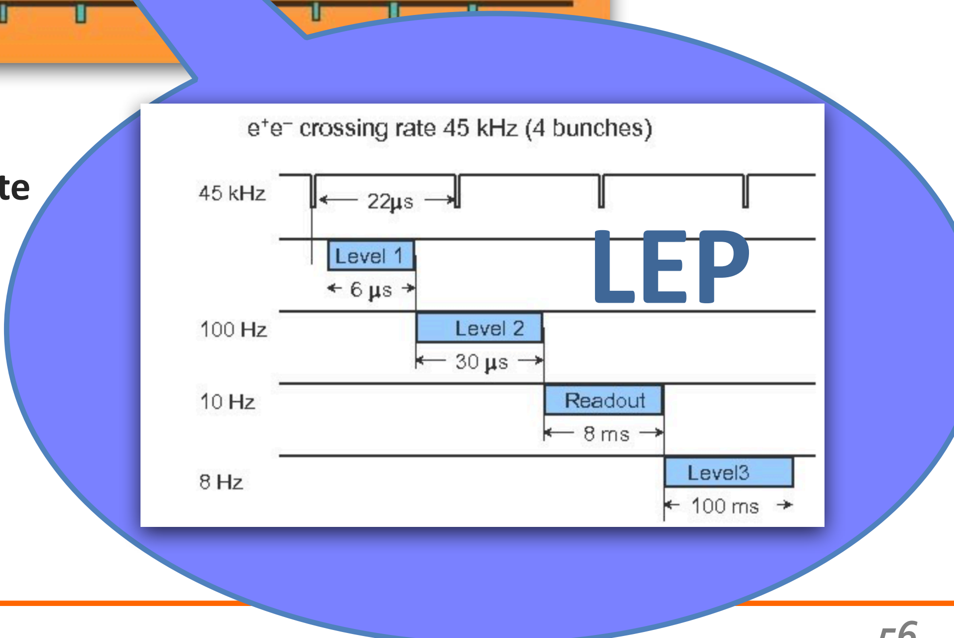
- **High-Level triggers in software (~1 s)**
  - electron/jet separation using
    - Cluster shapes
    - **Topological** variables, **tracking** information
    - **Isolation** criteria

Trigger processing

In a Synchronous level-1 trigger @ colliders



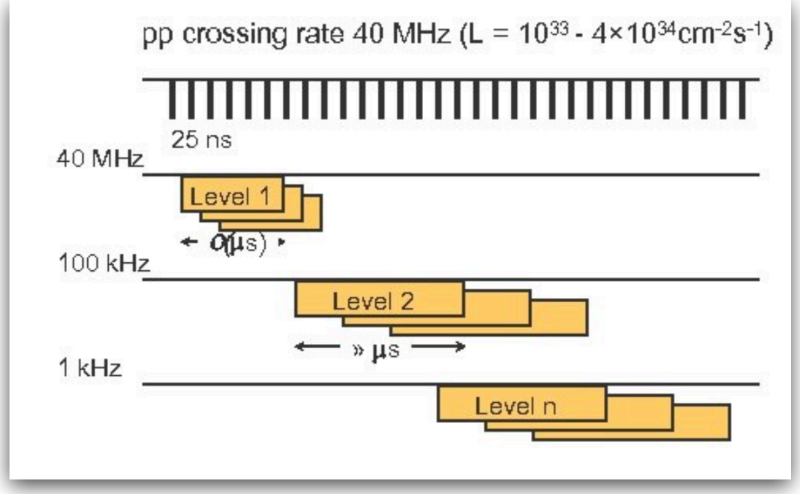
Bunch crossing clocks (BC)

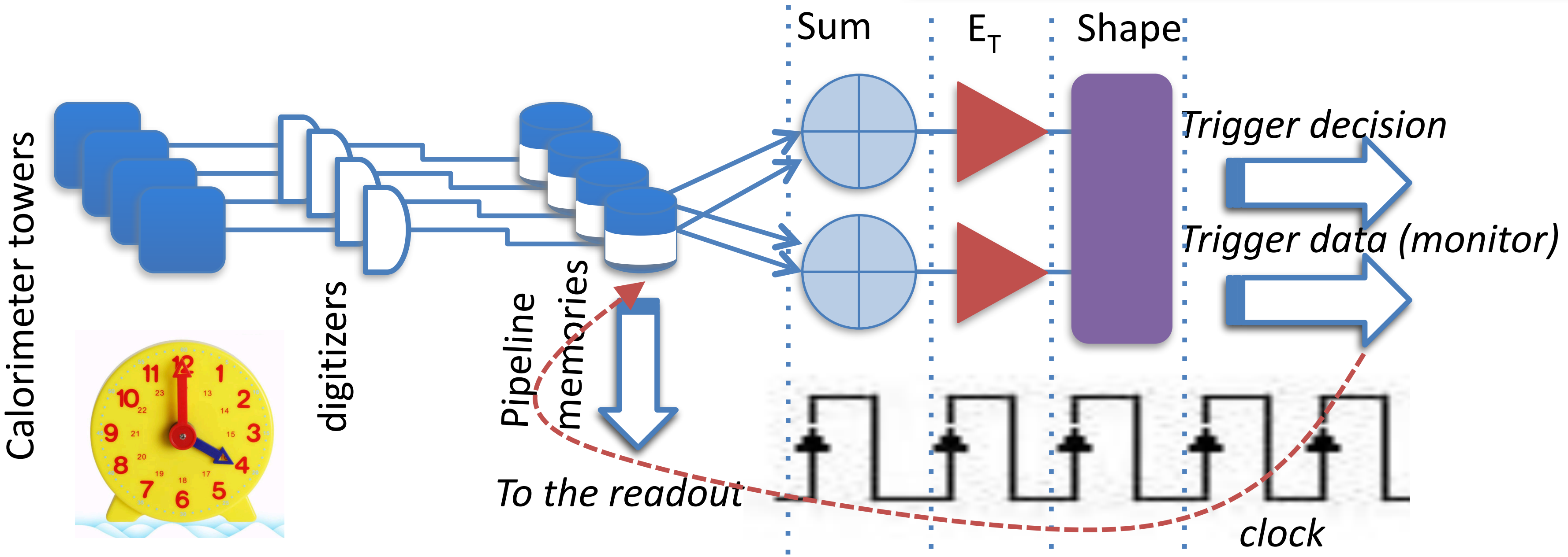


- **In colliders high luminosity is driven by high clock rate**
  - **can take trigger decision between two collisions?**
- **@LEP (BC interval 22 μs):**
  - Level-1 trigger latency < bunch-interval
  - no event overlap
  - most electronics outside the detector

Level-1 pipeline trigger

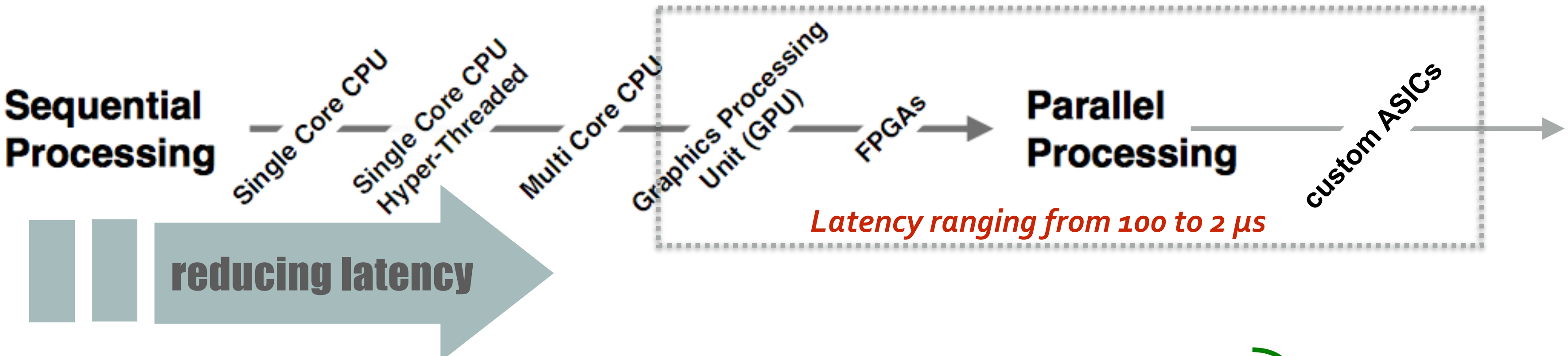
- **@LHC (25ns): L1 latency (few μs) exceeds bunch interval**
  - events overlap
  - signals pileup (in the detectors)
  - large detectors (large distance)
    - Latency = processing + data transmission time
- **Front-end buffer pipelines @ fixed trigger latency**
  - Each processor **concurrently** processes many events
  - Divide processing in steps, one per BC clock





Trigger (co-)processors

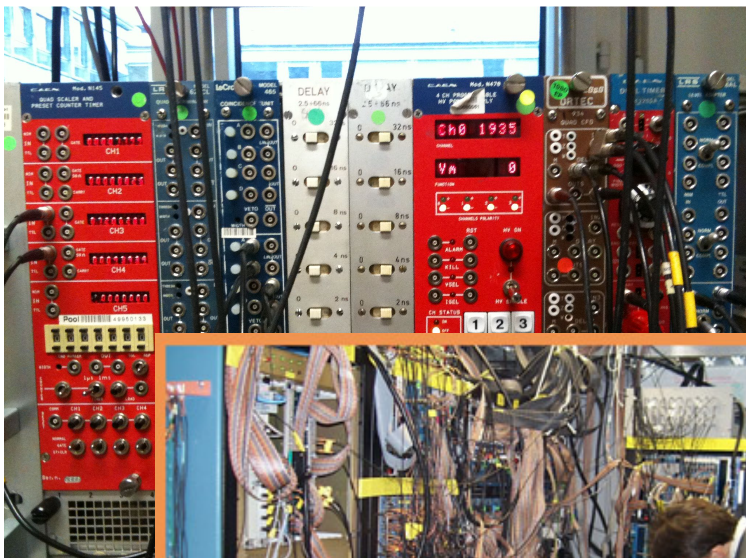
- General requirements
  - Fast processing
  - Flexible/programmable algorithms
  - Data compression and formatting
  - Monitor and automatic fault detection



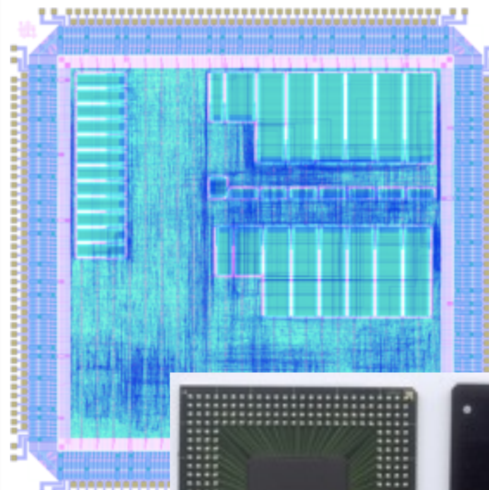
- **Microprocessors** (CPUs, GPGPUs, ARMs, DSP=digital signal processors..) — *need instructions*
  - Available on the market or specific, programmed only once
- **Programmable logic devices** (FPGAs, CAMs,…) — *already learned the task*
  - More operations/clock cycle, but costly and difficult software developing

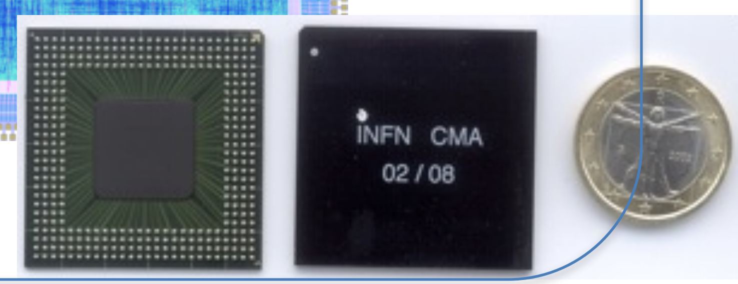
Choose your L1 trigger system

- Modular electronics
  - Simple algorithms
  - Low-cost
  - Intuitive and fast use

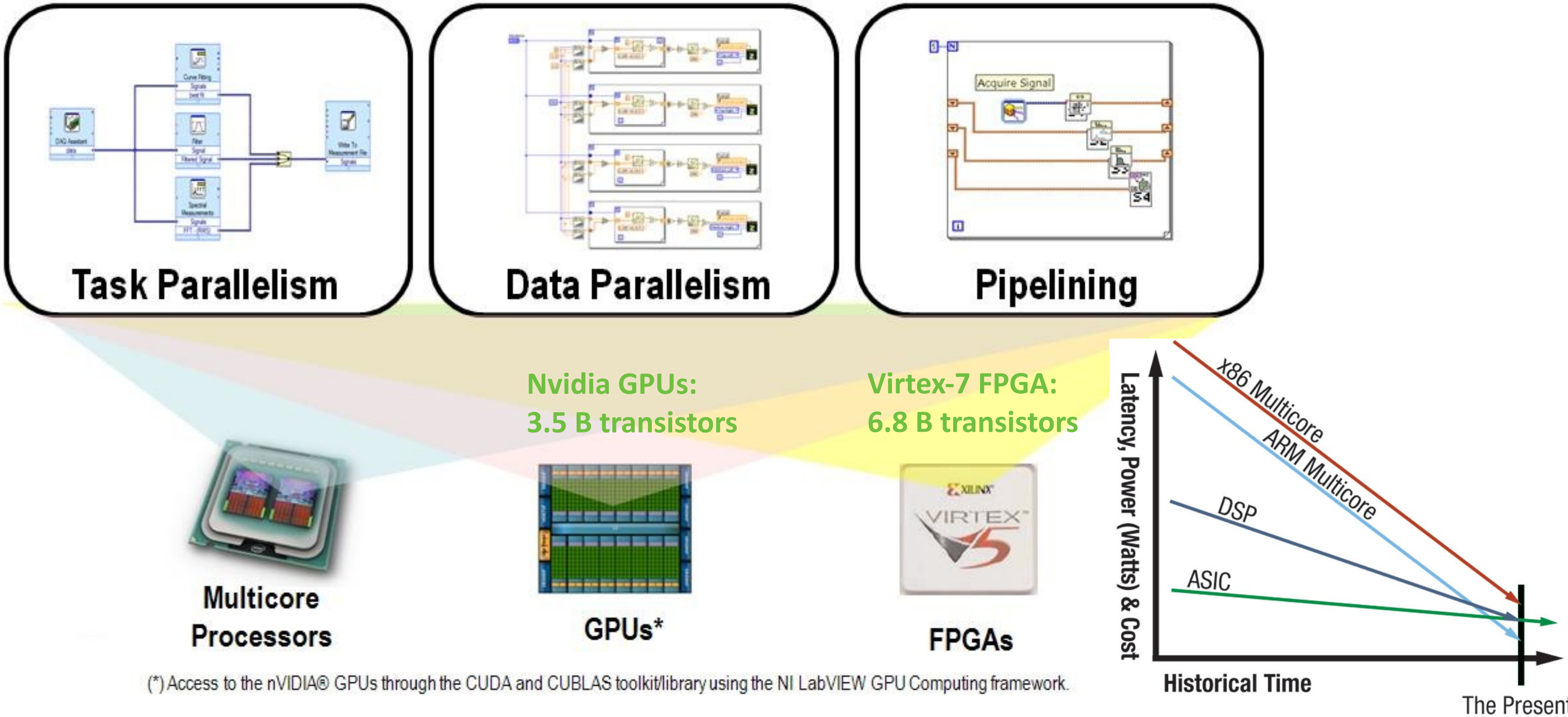


- Digital integrated systems
  - Highly complex algorithms
  - Fast signals processing
  - Specific knowledge of digital systems





Trends: combined technology



(*) Access to the nVIDIA® GPUs through the CUDA and CUBLAS toolkit/library using the NI LabVIEW GPU Computing framework.

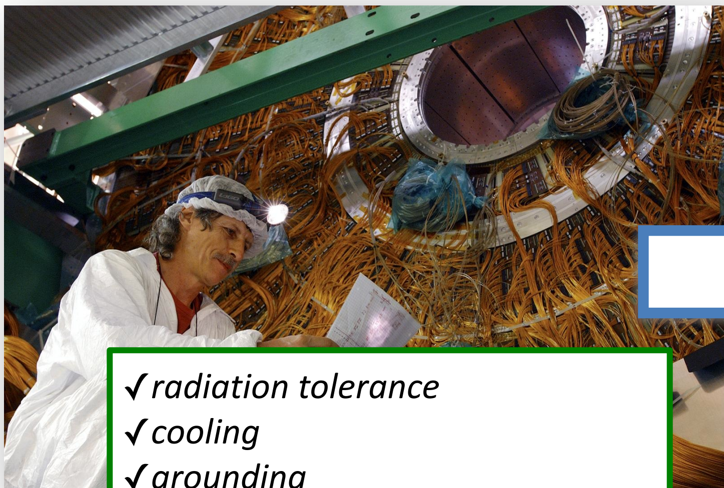
The right choice can be combining the best of both worlds by analysing which strengths of FPGA, GPU and CPU best fit the different demands of the application.

- Using standard interface (ethernet), can profit of standard tools and development time is reduced

Fast data movement

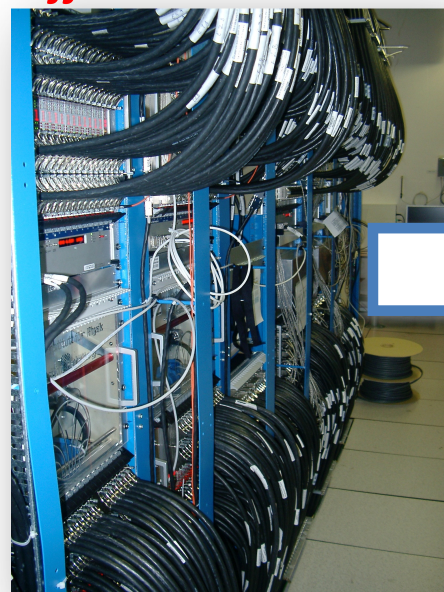
- Faster data processing are placed on-detector (close or joined to the FE)
- **Intermediate crates are good separation between FE (long duration) and PCs**

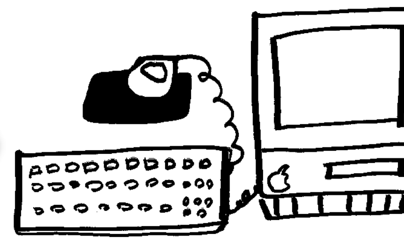
On-detector



- ✓*radiation tolerance*
- ✓*cooling*
- ✓*grounding*
- ✓*operation in magnetic field*
- ✓*very restricted access*

Off-detector





- **High-speed serial links**, electrical and optical, depending on distance
  - Low cost low-power LVDS links, @400 Mbps, < 10m
  - Optical GHz-links for longer distances (up to 100 m)

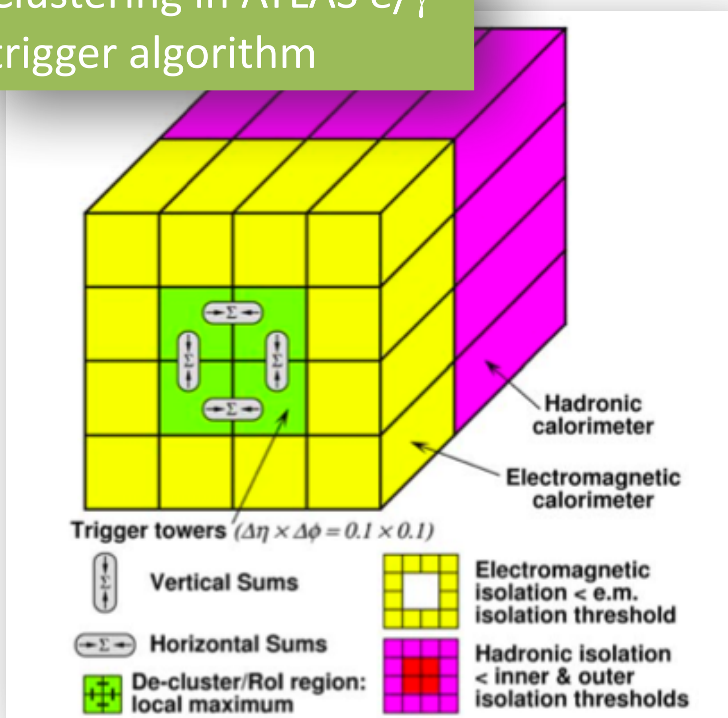
- **High density backplanes** for data exchanges in crates
  - High pin count, with point-to-point connections up to 160 Mbit/s
  - Large boards preferred

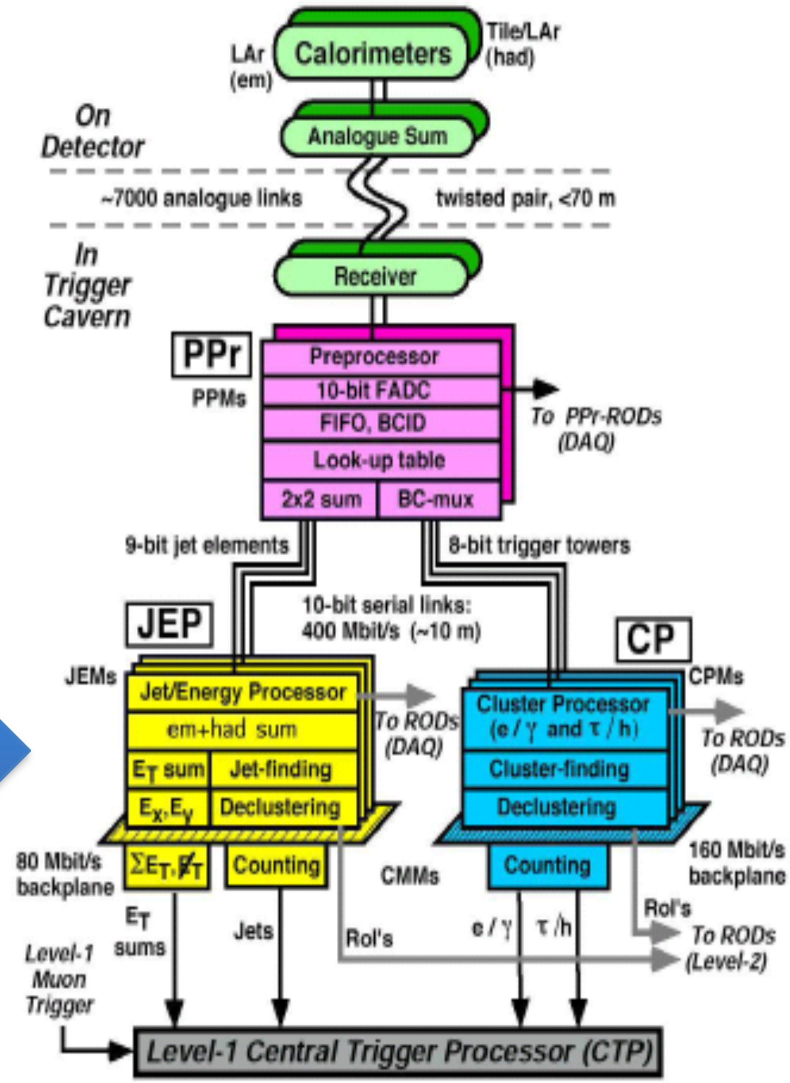
Example: ATLAS calorimeter trigger

- Cluster Processor (CP)
- Jet/Energy Processor (JEP)

- Programmable FPGAs and ASICs on custom boards
- Total of 5000 digital links @ 400 Mb/s

Clustering in ATLAS e/γ trigger algorithm



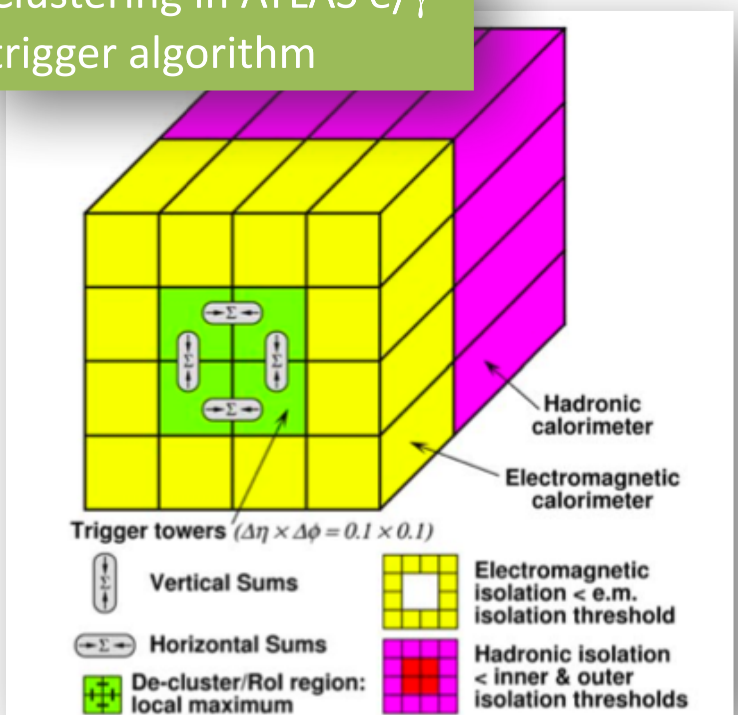


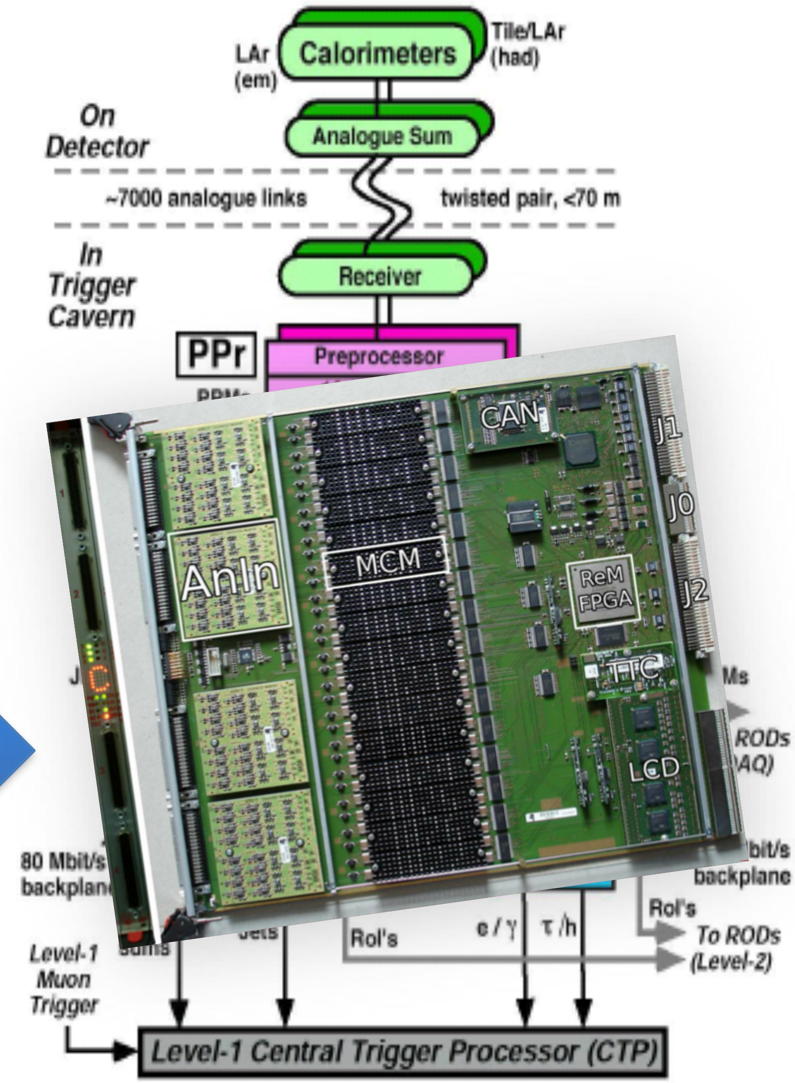
Example: ATLAS L1 calorimeter trigger

- Cluster Processor (CP)
- Jet/Energy Processor (JEP)

- Programmable FPGAs and ASICs on custom boards
- Total of 5000 digital links @ 400 Mb/s

Clustering in ATLAS e/γ trigger algorithm





Event building and Event filtering

	Levels	L1 rate (Hz)	Event size	Readout bandwidth	Event filter output
LEP	2/3	1 kHz	100 kB	few 100 kB/s	~5 Hz
ATLAS	2/3	100 kHz (L2: 10 kHz)	1.5 MB	30 GB/s (incremental Event Building)	~1000 Hz
CMS	2	100 kHz	1.5 MB	100 GB/s	~1000 Hz

cannot store on disk at this rate!

After the L1-trigger selection, data rates are reduced, but can be still massive!

Event building and Event filtering

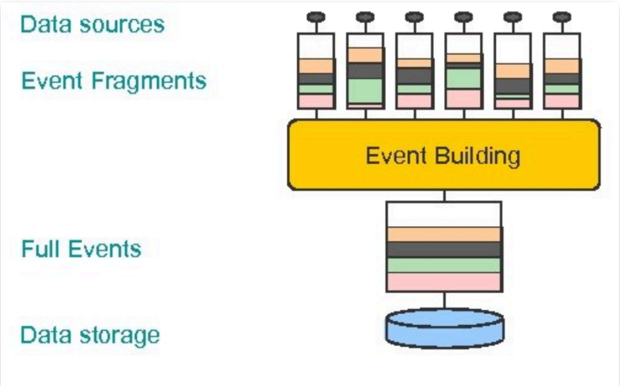
	Levels	L1 rate (Hz)	Event size	Readout bandwidth
LEP	2/3	1 kHz	100 kB	few 100 kB/s
ATLAS	2/3	100 kHz (L2: 10 kHz)	1.5 MB	30 GB/s (incremental Event Building)
CMS	2	100 kHz	1.5 MB	100 GB/s

Event filter output
~5 Hz
~1000 Hz
~1000 Hz

cannot store on disk at this rate!

After the L1-trigger selection, data rates are reduced, but can be still massive!

- **Readout Network** to collect data from Front-End buffers
  - LEP: 40 MB/s VME bus was able to support the bandwidth
  - LHC: **latest technologies** in processing, high-speed network, optical data transmission

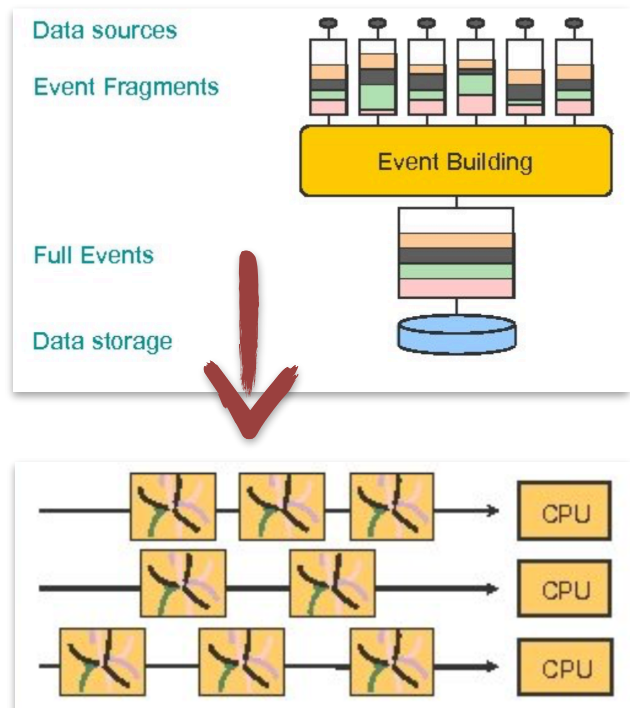


Event building and Event filtering

	Levels	L1 rate (Hz)	Event size	Readout bandwidth	Event filter output
LEP	2/3	1 kHz	100 kB	few 100 kB/s	~5 Hz
ATLAS	2/3	100 kHz (L2: 10 kHz)	1.5 MB	30 GB/s (incremental Event Building)	~1000 Hz
CMS	2	100 kHz	1.5 MB	100 GB/s	~1000 Hz

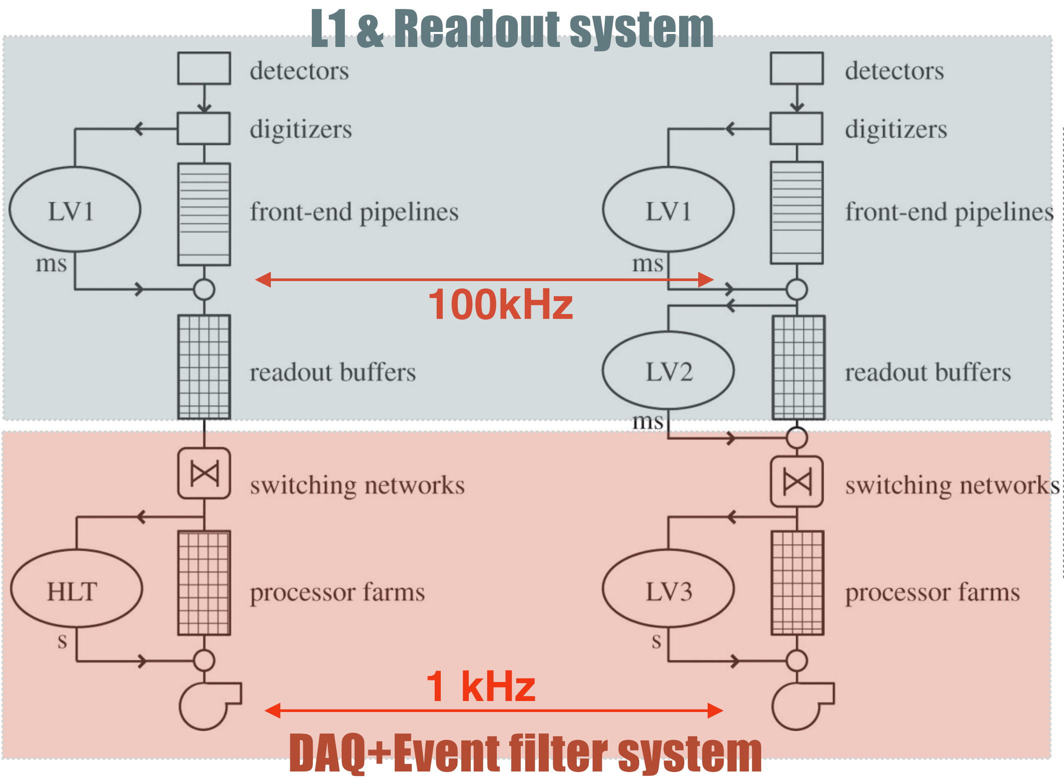
cannot store on disk at this rate!

After the L1-trigger selection, data rates are reduced, but can be still massive!



- **Readout Network** to collect data from Front-End buffers
  - LEP: 40 MB/s VME bus was able to support the bandwidth
  - LHC: **latest technologies** in processing, high-speed network, optical data transmission
- **Event Building and Filter farms** on networks
  - **farm processing**: one event per processor (larger latency, but scalable)
  - additional networks regulates the CPU assignment
  - commercial products: PCs (linux based), Ethernet protocols, standard LAN, configurable devices

Choosing the T/DAQ architecture



ATLAS/CMS Example

- **1 MB/event at 100 kHz for O(100 ms) Event filter latency**
  - Network: 1 MB*100 kHz = **100 GB/s**
  - EF farm: 100 kHz*100 ms = **O(10^4) CPU cores**
- Can add intermediate steps (level-2) to reduce resources, at cost of complexity (at ms scale)

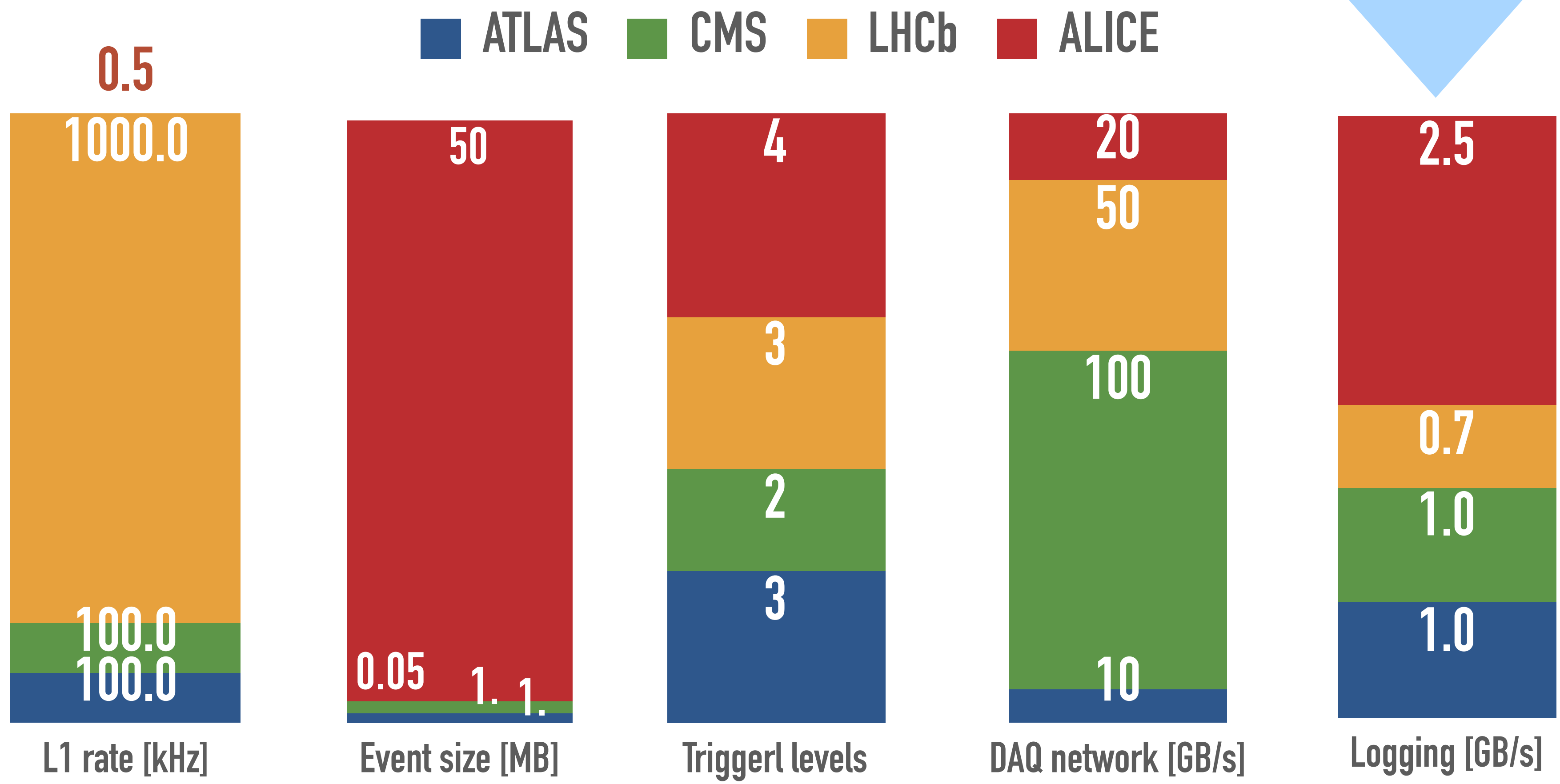
See S.Cittolin, DOI: 10.1098/rsta.2011.0464

- **High data rates can be held with different approaches**
  - Network-based event building (LHC example: CMS)
  - Seeded reconstruction of partial data (LHC example: ATLAS)

Comparing LHC experiments design

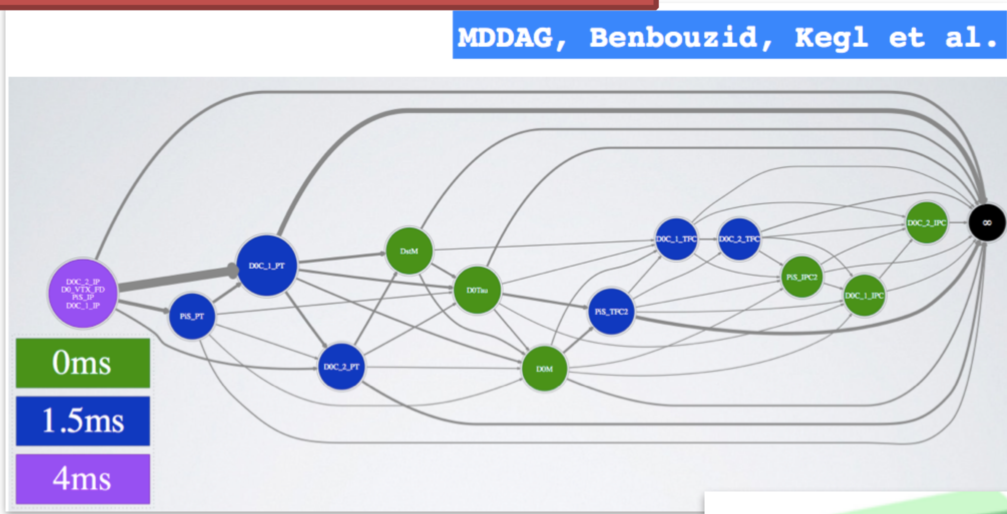
The 4 LHC experiments share the CERN budget for computing resources, which is the constrain between trigger and DAQ rate

Design in 2009:
allowed storage and processing resources: ~1GB/s



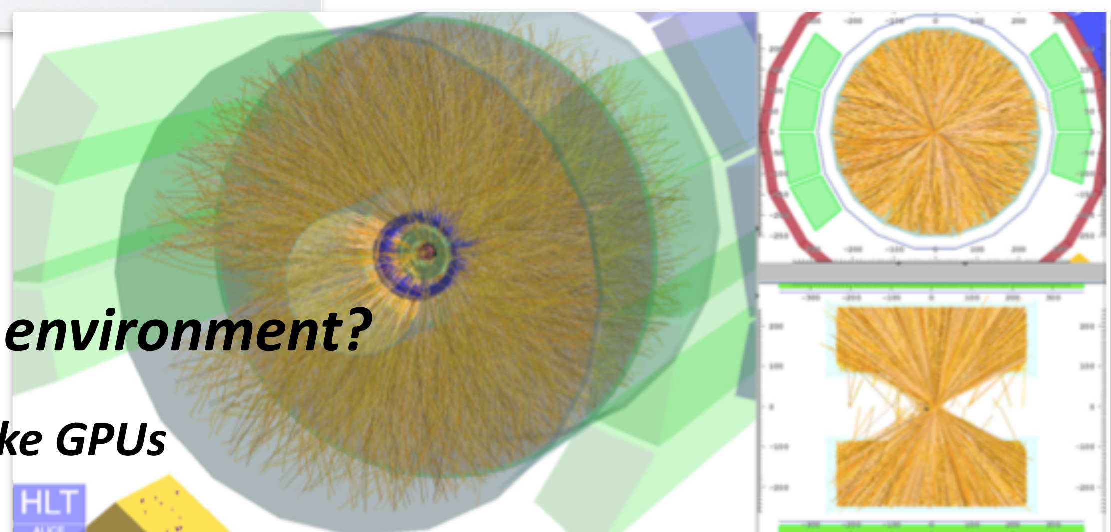
Can we use any algorithms online?

Latency is the constraint!



Multivariate analysis?

Yes, recently included in both software and hardware (FPGA) processing

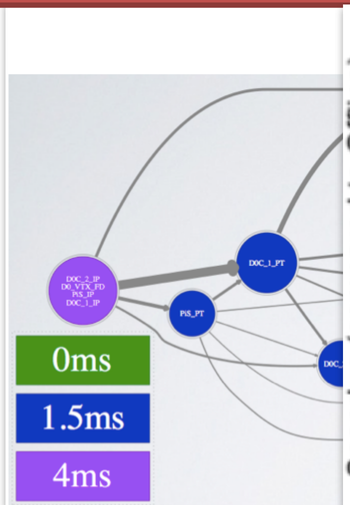


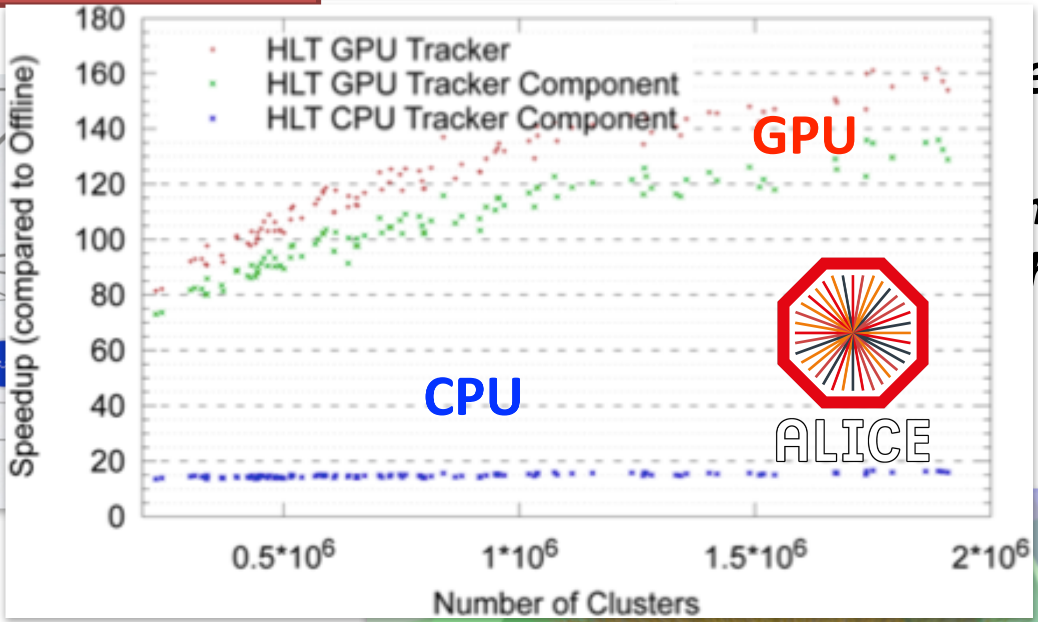
Pattern recognition in dense environment?

Yes, with the help of co-processors like GPUs

Can we use any algorithms online?

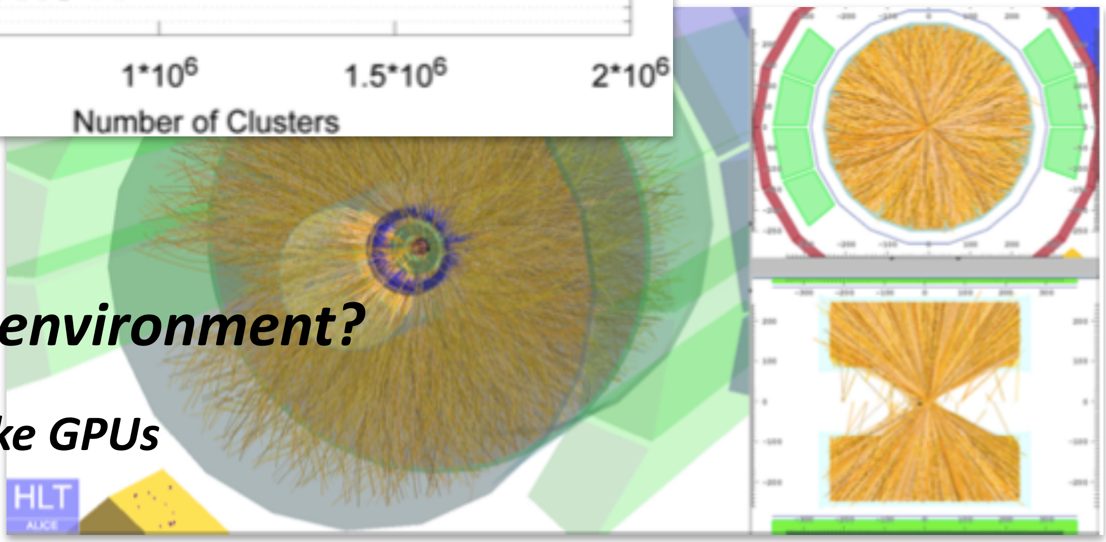
Latency is the constraint!





...e analysis?

...ncluded in both ...hardware (FPGA)

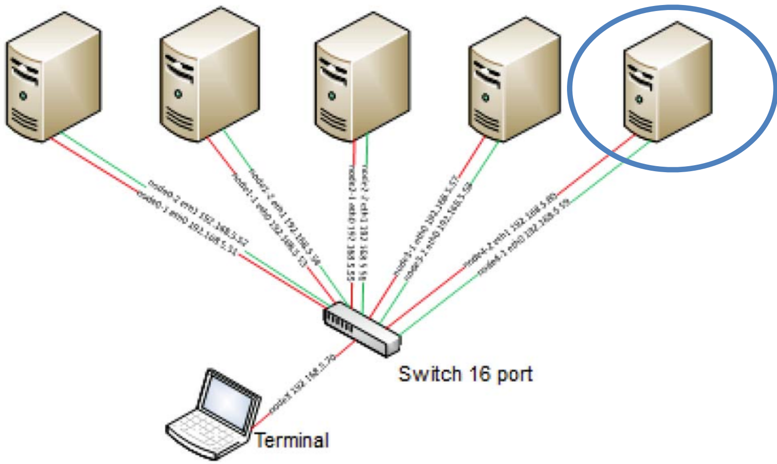


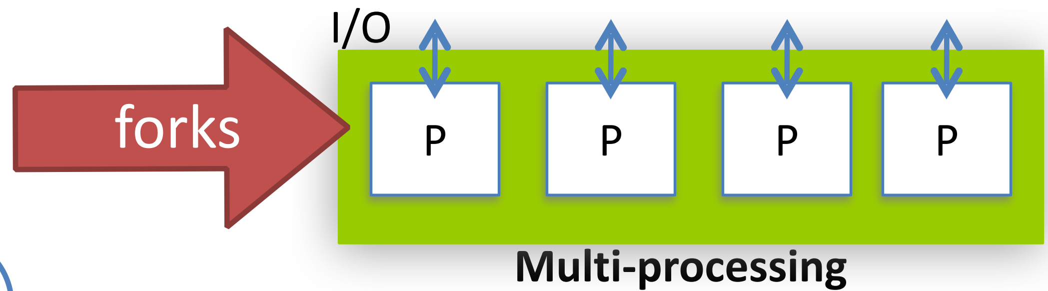
Pattern recognition in dense environment?

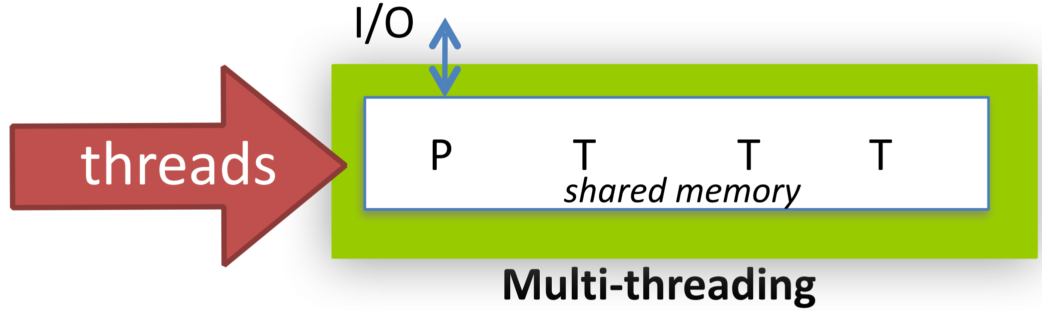
Yes, with the help of co-processors like GPUs

online software: design principles

Multi-core processors







- **Early rejection:** alternate feature extraction with hypothesis testing
- **Partial event data processing**
- **Fast reconstruction:** same as offline (easy maintenance and higher efficiency)
- **Increase parallelism to exploit all CPU resources (all cores!)**
  - event-level and algorithm-level concurrency

What's up for the future of software trigger?

what we want

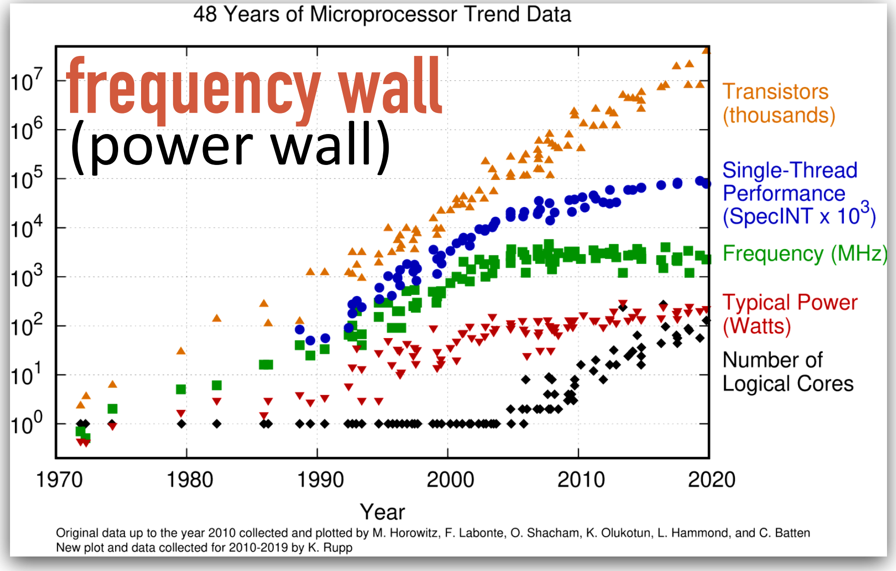
$$R = \mu \cdot f_{BC} = \sigma_{in} \cdot L$$

- ➡ Linear increase of digitisation time
- ➡ Factorial increase of reconstruction time
- ➡ Larger events, lots of more memory

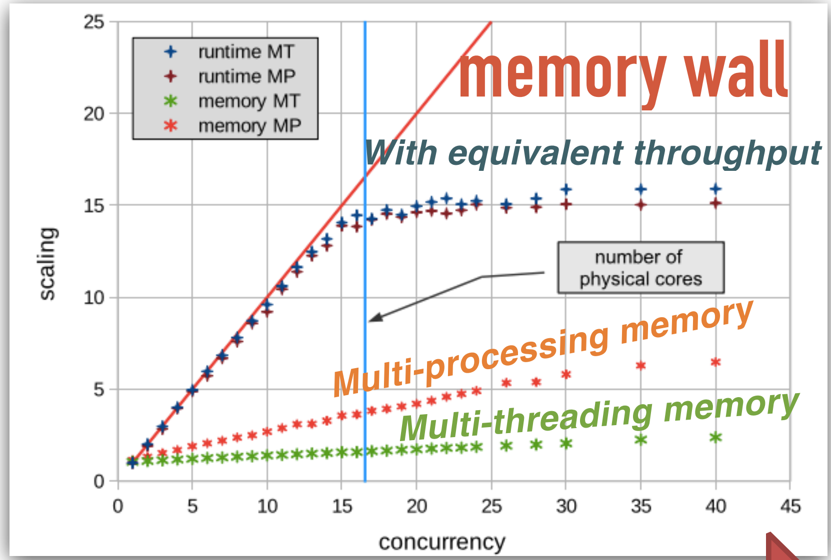
Higher rates requires more processing/memory

what we have

Technology evolution



Throughput/memory scaling for a tracking demonstrator

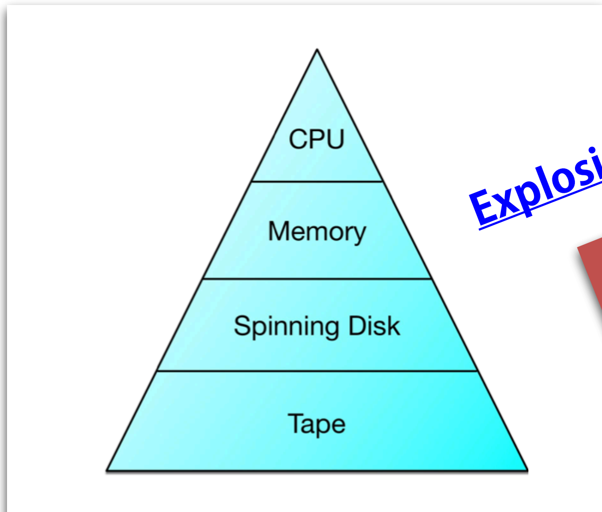


Limits on the capabilities of serial processing

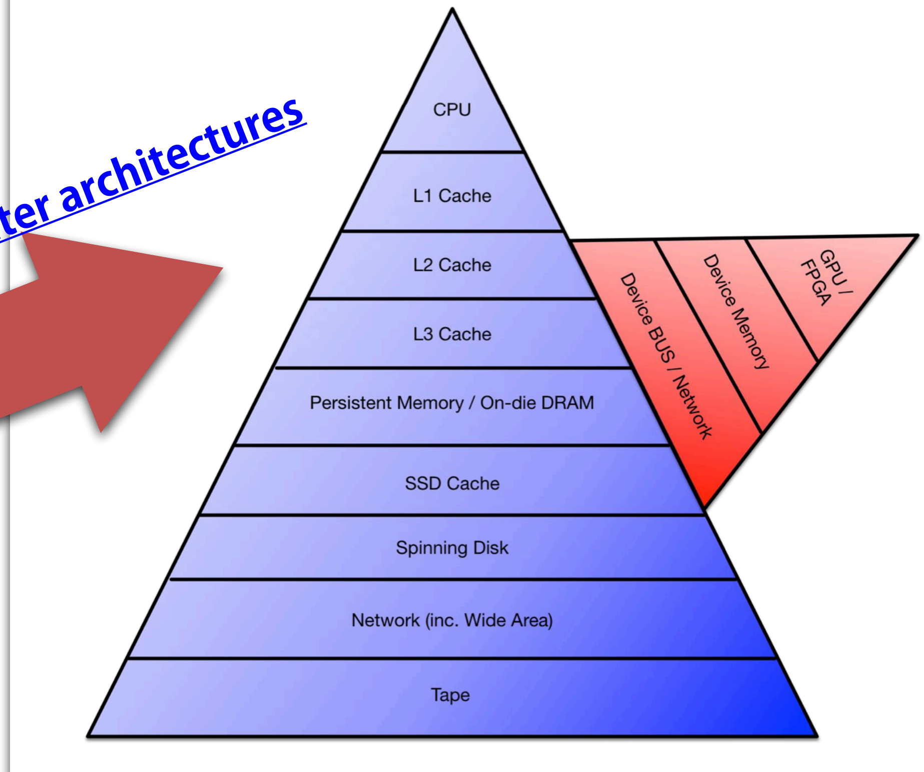
(Trigger) software evolution to break walls

"We're approaching the limits of computer power – we need new programmers now"
John Naughton, Guardian

See LP-2022 slides from Graeme Stewart



Explosion of novel computer architectures



==> Exploiting CPU hardware in new architectures
- more complicated programming (vectorisation, memory sharing…)
- Exploit more efficiently instruction level parallelism (ILP)

==> Move towards concurrent processing
- and delegate accelerators for specific tasks

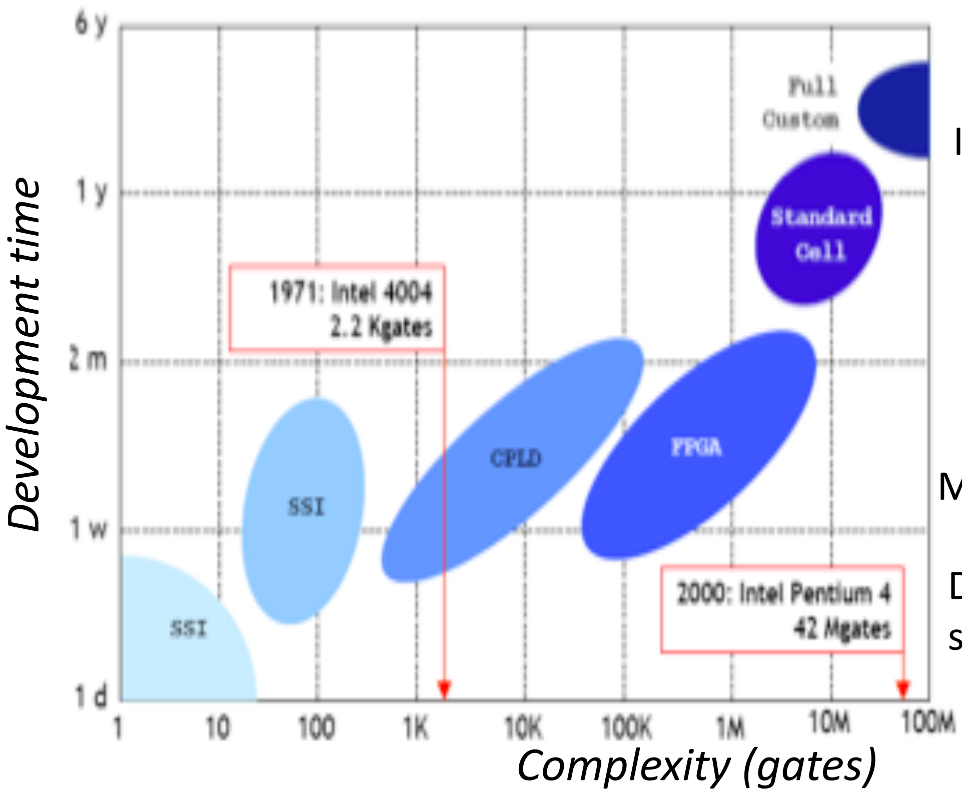
Concluding remarks

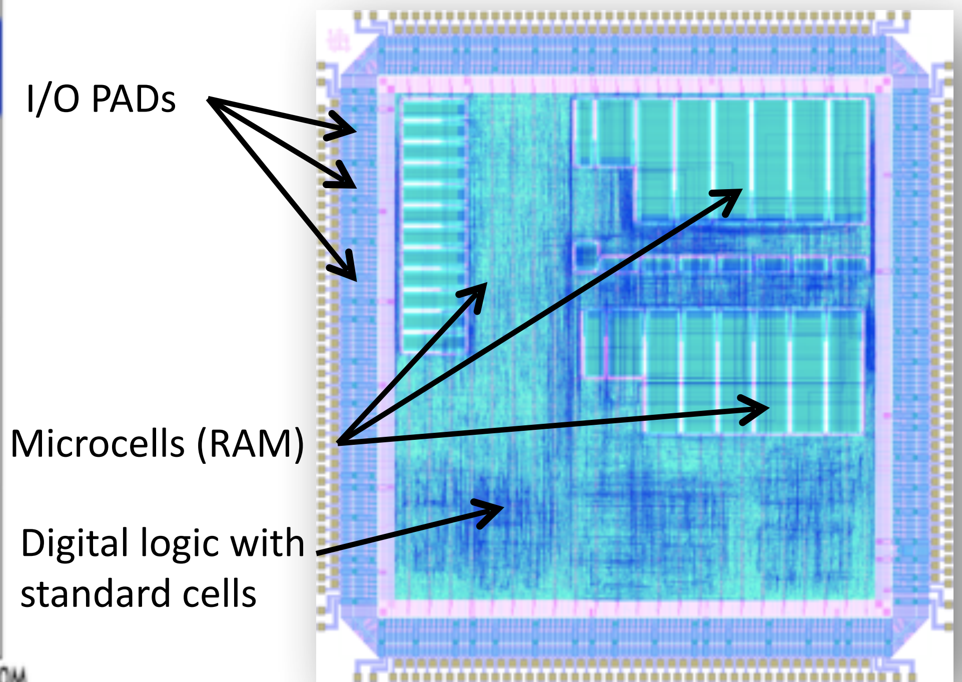
- The trigger strategy is a trade-off between physics requirements and affordable systems and technologies
  - A good design is crucial – then the work to maintain optimal performance is easy
- Here we just reviewed the main trigger requirements coming from physics
  - High efficiency – rate control
  - Perfect knowledge of the trigger selection on signal and background
  - Flexibility and redundancy
- Microelectronics, networking, computing expertise are required to build an efficient trigger system
  - But being always in close contact with the physics measurements we want to study

Back-up slides

Custom trigger processors?

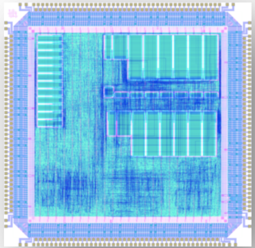
- Application-specific integrated circuits (**ASICs**): optimized for fast processing (Standard Cells, full custom)
  - Intel processors, ~ GHz
- Programmable ASICS (like Field-programmable gate arrays, **FPGAs**)
  - Easily find processors @ 100 MHz on the market (1/10 speed of full custom ASICs)



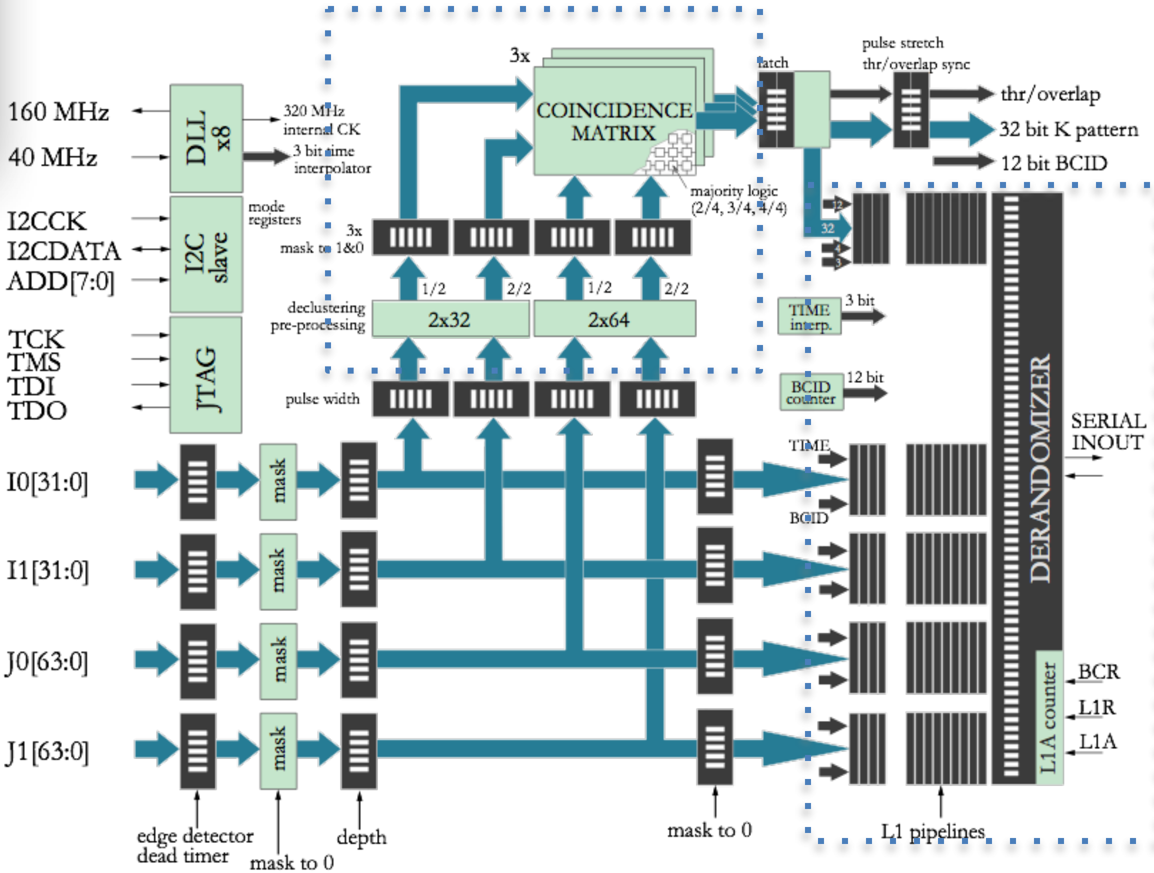


Layout of the CM ASIC

Example: logic of a trigger ASIC



Trigger part



Trigger output

Readout output

Digital inputs from detector

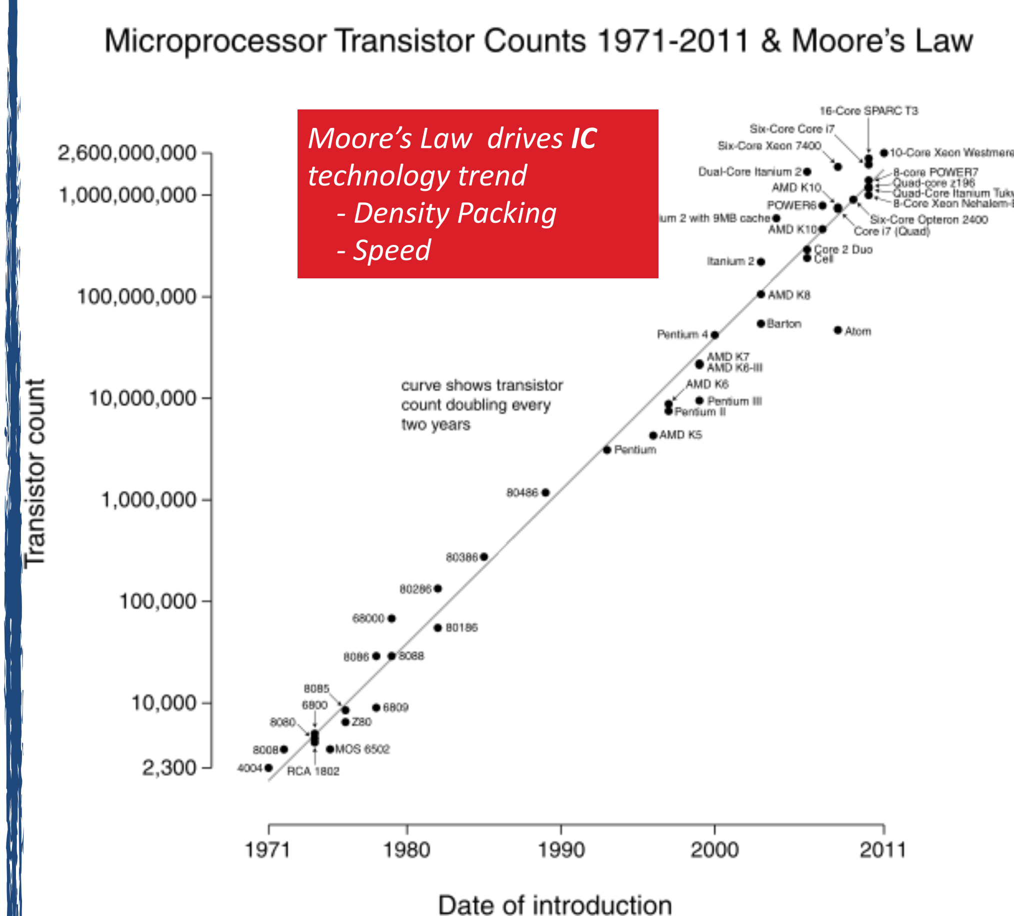
Input: synchronization signals

Readout part

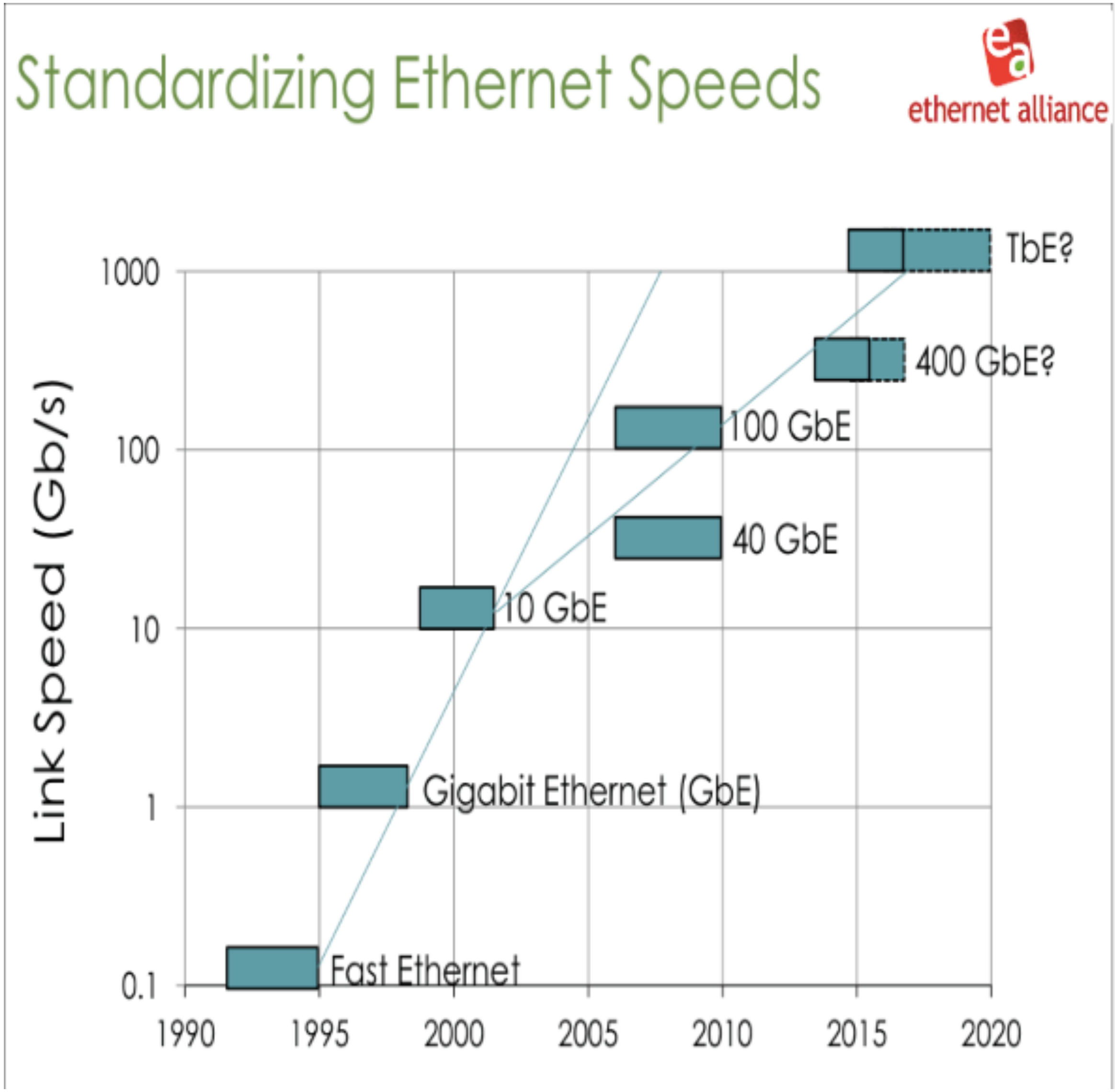
Coincidence Matrix ASIC for Muon Trigger in the Barrel of ATLAS

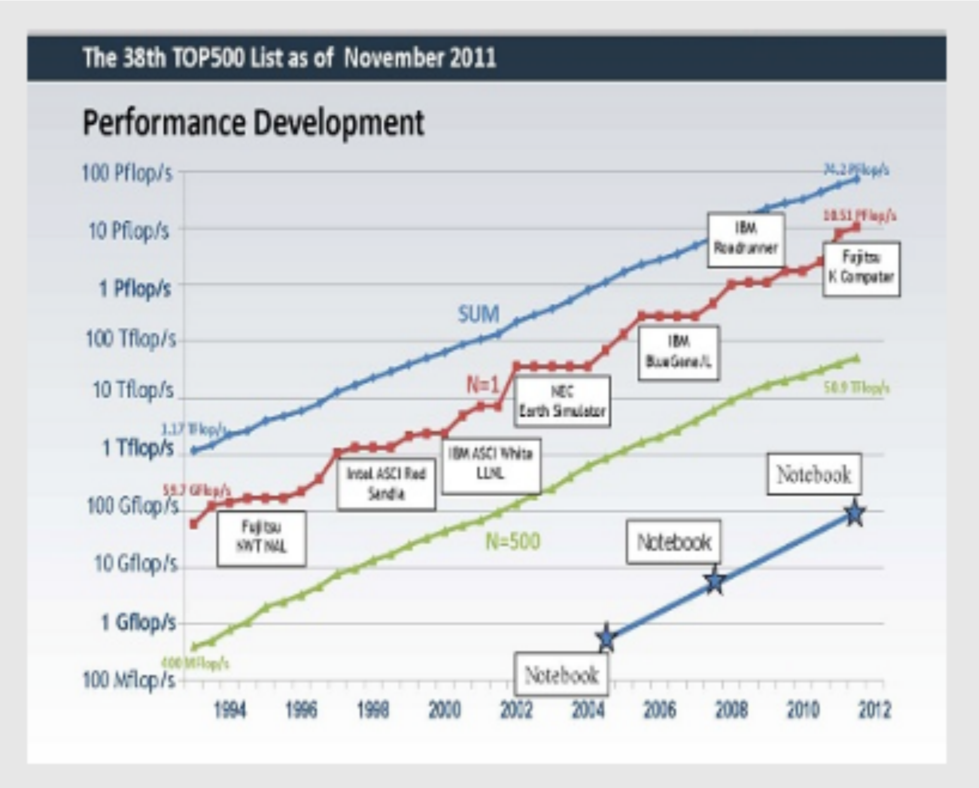
Trends in processing technology

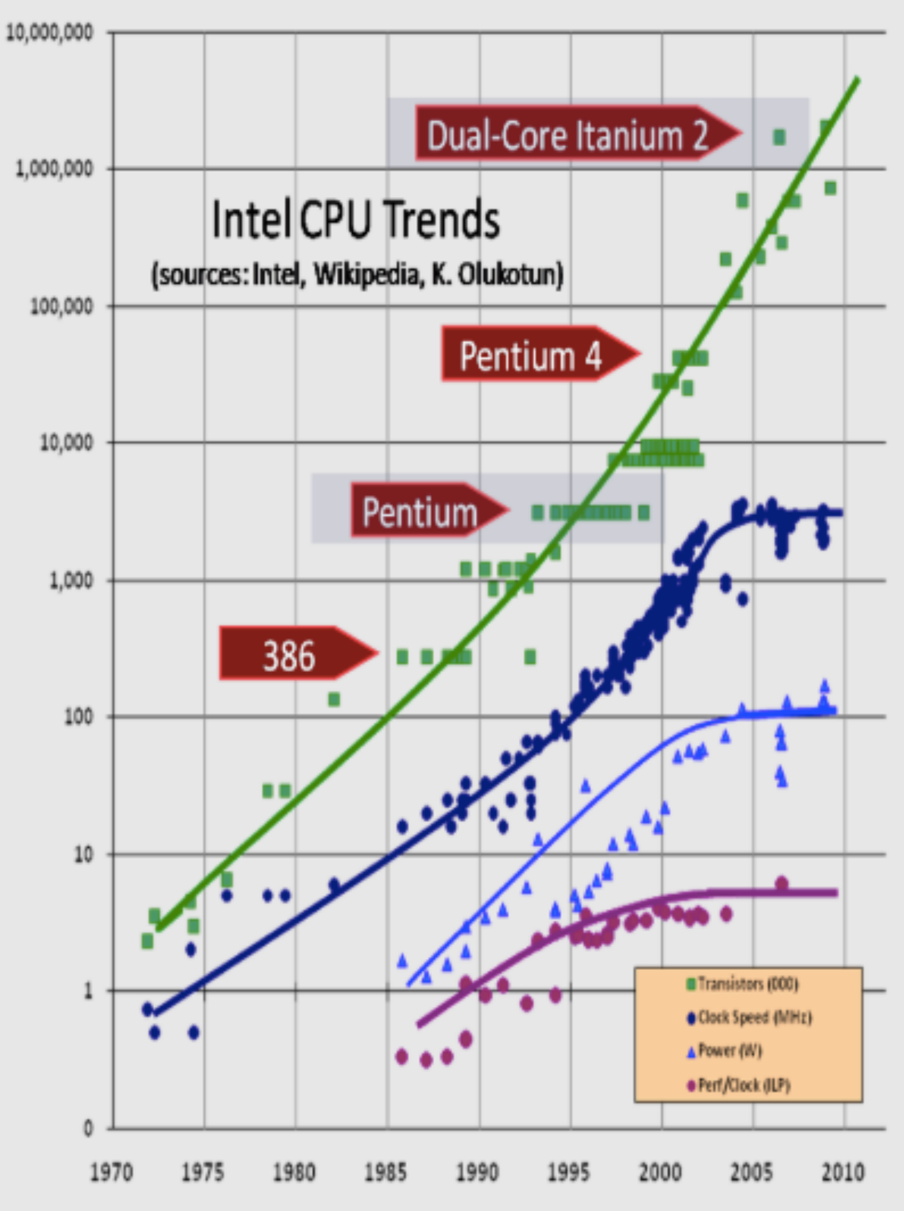
- Request of higher complexity → higher chip density → **smaller structure size** (for transistors and memory size): **32 nm → 10 nm**
  - Nvidia GPUs: 3.5 B transistors
  - Virtex-7 FPGA: 6.8 B transistors
  - 14 nm CPUs/FPGAs in 2014
- For FPGAs, smaller feature size means higher-speed and/or less power consumption
- Multi-core evolution
  - Accelerated processing GPU+CPU
  - Needs increased I/O capability
- Moore's law will hold at least until 2020, for FPGAs and co-processors as well
- Market driven by cost effective components for Smartphones, Phablets, Tablets, Ultrabooks, Notebooks ….
- Read also: http://cern.ch/go/DFG7



Moore's Law: the number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years (Wikipedia)


Standardizing Ethernet Speeds
ethernet alliance
Link Speed (Gb/s)
1000
100
10
1
0.1
1990
1995
2000
2005
2010
2015
2020
TbE?
400 GbE?
100 GbE
40 GbE
10 GbE
Gigabit Ethernet (GbE)
Fast Ethernet


The 38th TOP500 List as of November 2011
Performance Development
100 Pflop/s
10 Pflop/s
1 Pflop/s
100 Tflop/s
10 Tflop/s
1 Tflop/s
100 Gflop/s
10 Gflop/s
1 Gflop/s
100 Mflop/s
SUM
N=1
N=500
IBM Roadrunner
Fujitsu K Computer
IBM BlueGene/L
NEC Earth Simulator
IBM ASCI White LLNL
Intel ASCI Red Sandia
Fujitsu NWT NAL
Notebook
Notebook
Notebook
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012


Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)
Dual-Core Itanium 2
Pentium 4
Pentium
386
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
0
1970
1975
1980
1985
1990
1995
2000
2005
2010
Transistors (000)
Clock Speed (MHz)
Power (W)
Perf/Clock (ILP)