



ISOTDAQ 2016

7th International School of Trigger and Data Acquisition

25th January - 2nd February 2016

Weizmann Institute of Science, Rehovot, Israel

Trigger architectures

F.Pastore (Royal Holloway Univ. of London)

francesca.pastore@cern.ch



Build up a trigger system

Ensure good efficiency with...

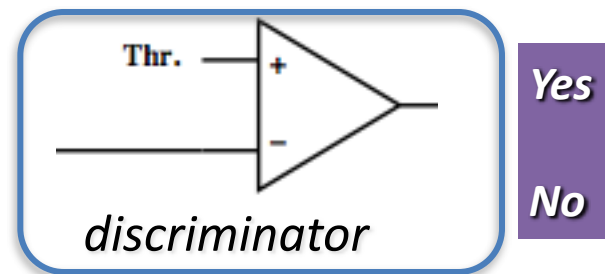
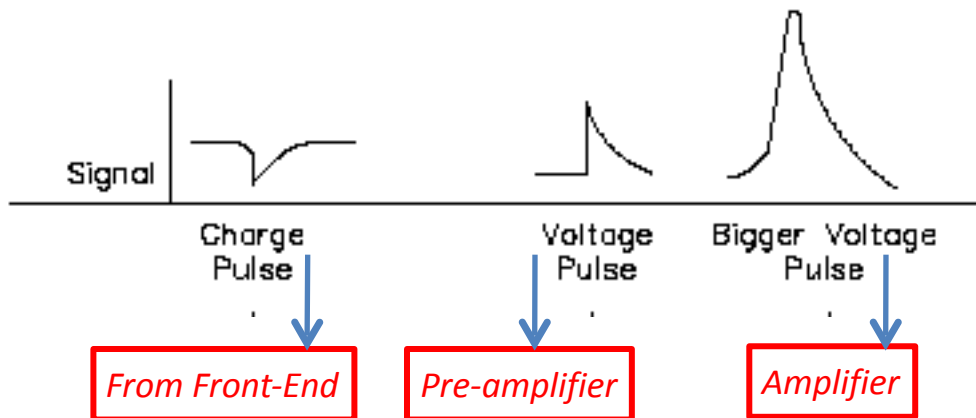


Robustness! Win against the unexpected!

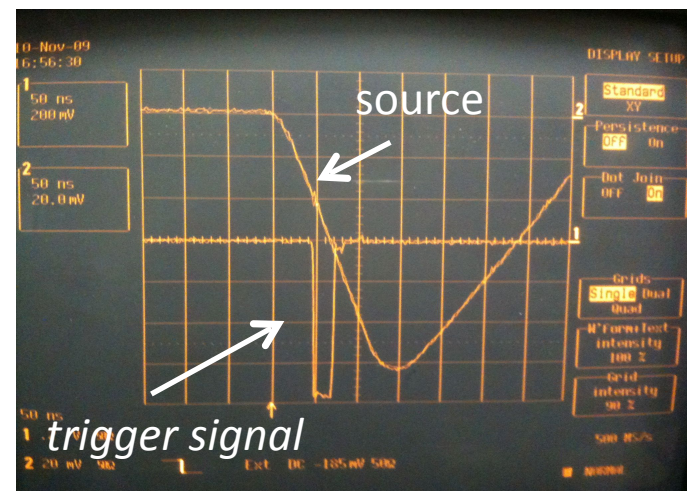
- **Flexibility: to cope changes in conditions and background**
 - Programmable thresholds, high granularity to maintain uniform performance, able to follow changes of luminosity, beam-size and vertex position, able to reach physics results also after 10 years of data taking
- **Redundancy: to make trigger 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 good rejection of the unwanted background

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,....



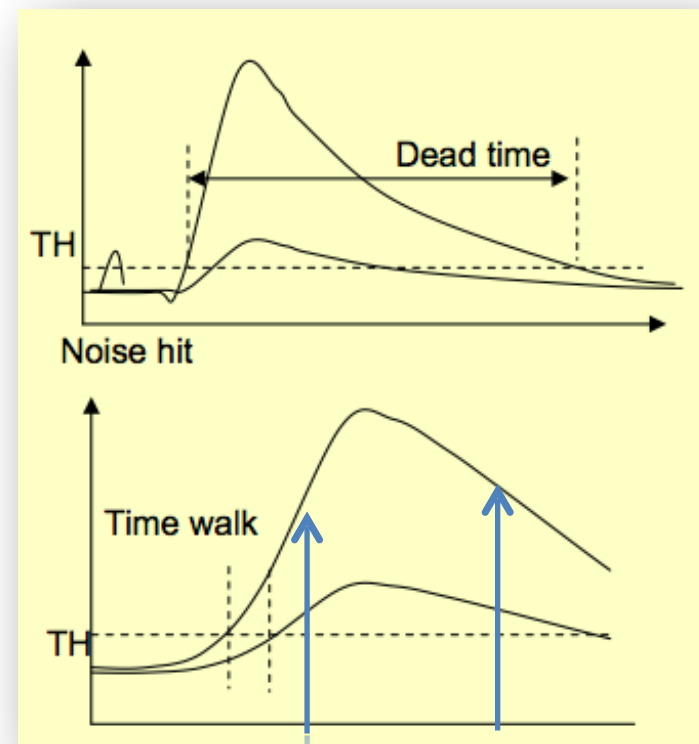
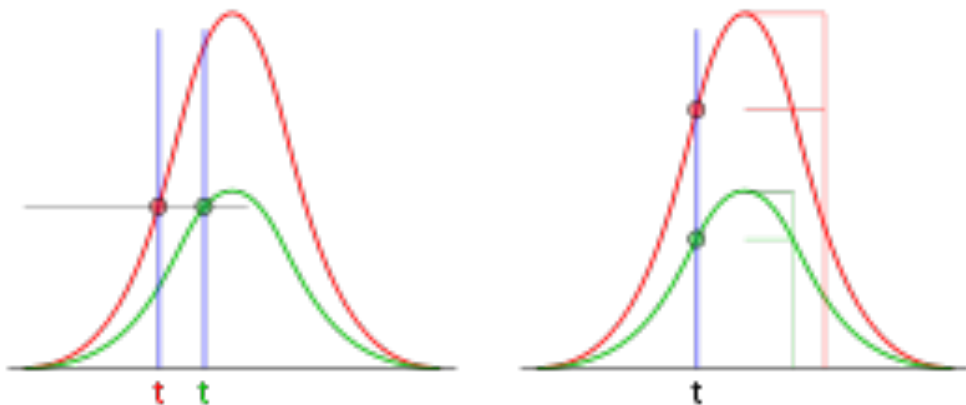
- The simplest trigger is: **apply a threshold**
 - Look at the signal
 - Apply a threshold as low as possible, since signals in HEP detectors have large amplitude variation
 - Compromise between hit efficiency and noise rate



Signals are different...

- Pulse width
 - Limits the effective hit rate
 - Must be adapted to the desired trigger rate

- Time walk
 - The threshold-crossing time depends on the amplitude of the signal
 - Must be minimised in a good trigger system



Leading edge

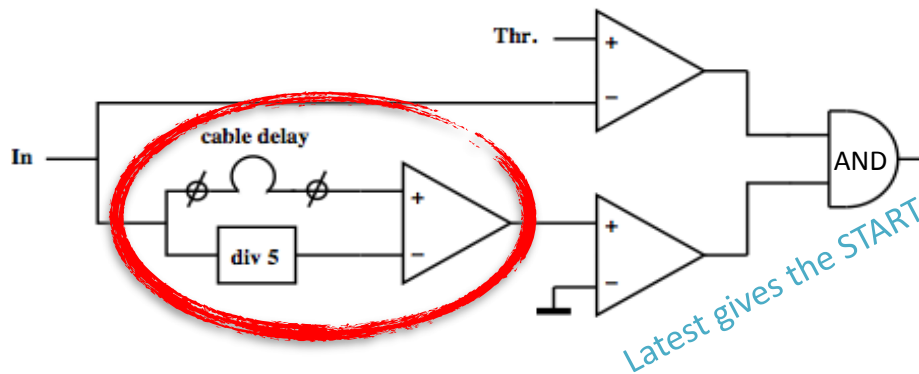
Trailing edge

- If two signals have identical rise time, at different amplitude, the time walk can be eliminated triggering when a certain fraction of the amplitude is passed
 - Good for scintillation detectors and PMT pulses mainly

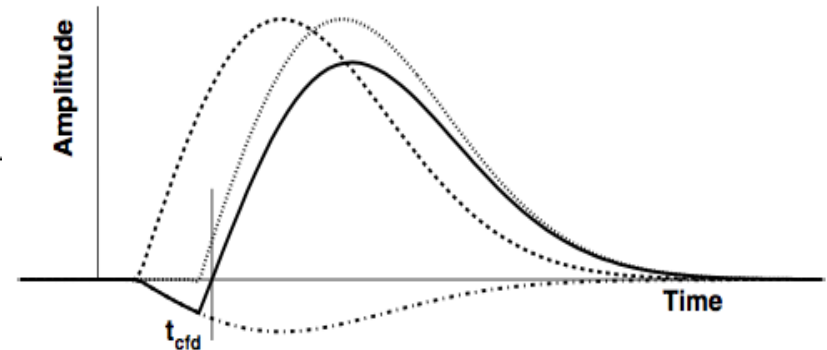
The constant fraction discriminator

If two signals have the same rising time at a fraction f
 $t(A_f) - t(A_0) = \text{constant}$

$$\rightarrow A(\text{delay}, t) - f \cdot A(t) = 0 \quad \text{at } t_{\text{CFD}}$$

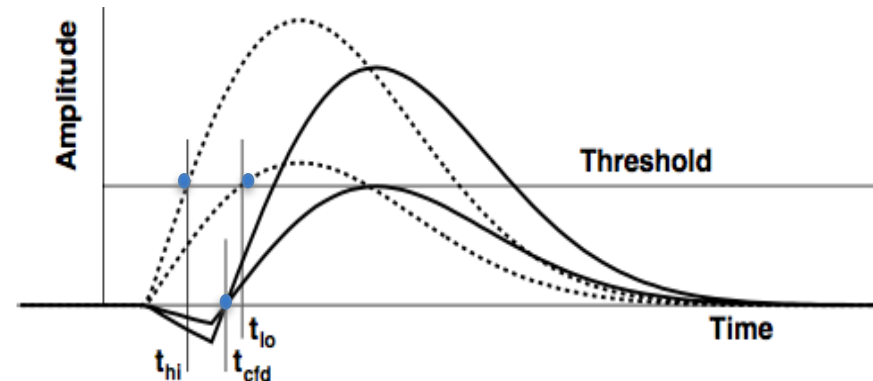


--- Input pulse
 Delayed input pulse
 - - - Attenuated inverted input
 — Bipolar pulse



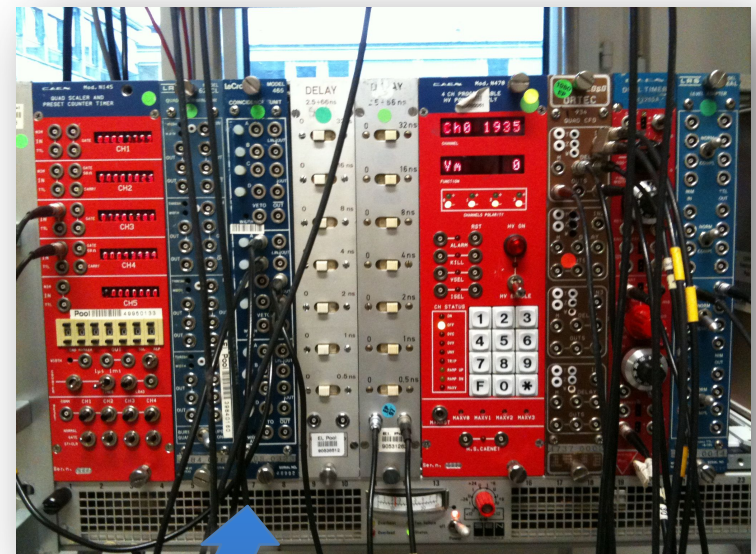
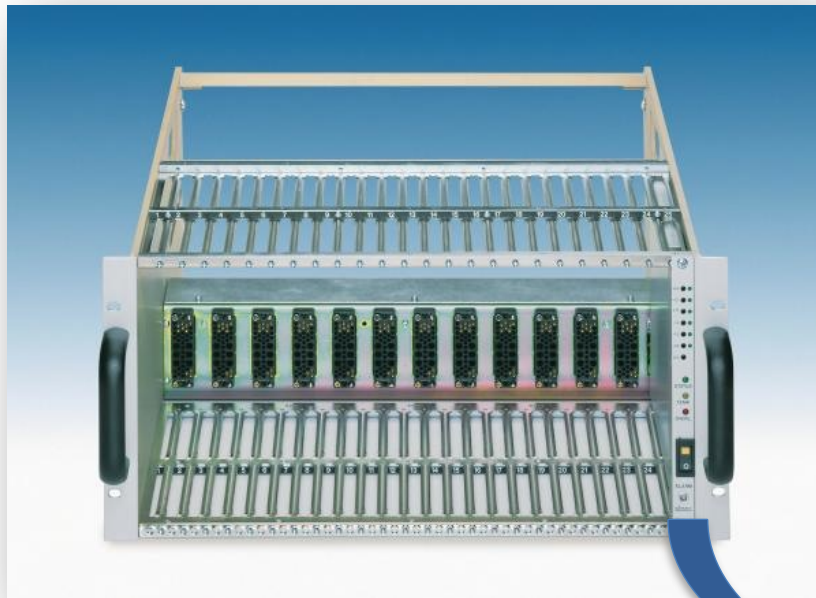
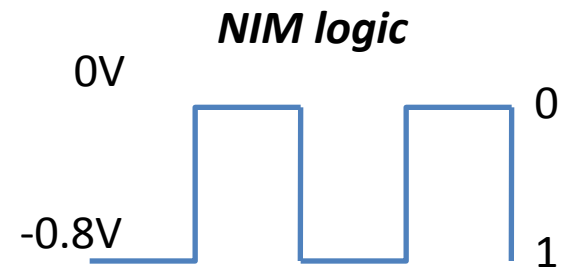
The output of the CFD fires when the bipolar pulse changes polarity

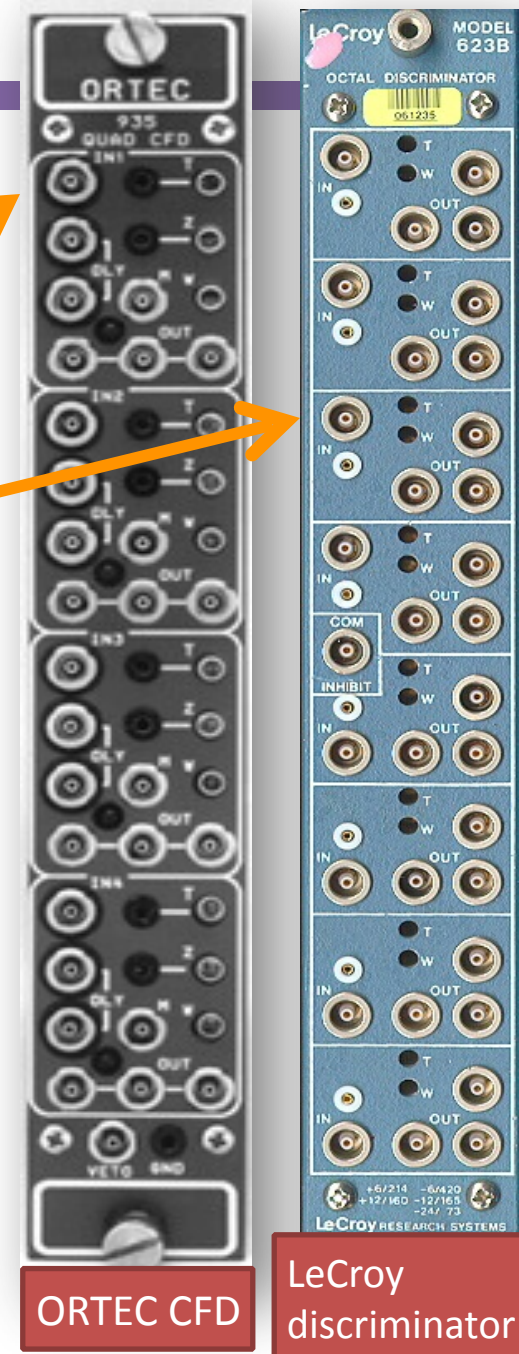
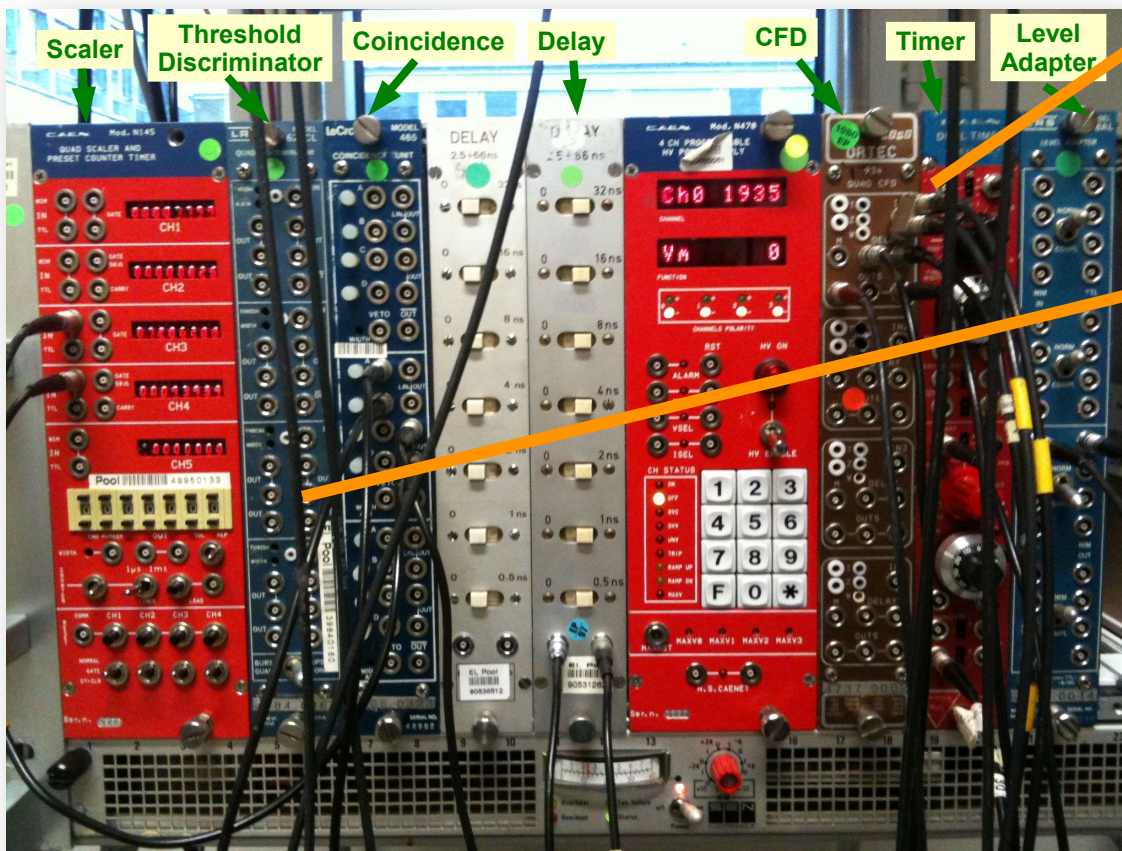
- Attenuation and delay (configurable) applied before the discrimination determines t_{CFD}
- If the delay is too short, the unit works as a normal discriminator because the output of the normal discriminator fires later than the CFD part



And now build your own trigger system

- A simple trigger system can start with a NIM crate
- Common support for electronic modules, with standard impedance, connections and logic levels: negative
 - **-16 mA into 50 Ohms = -0.8 Volts**

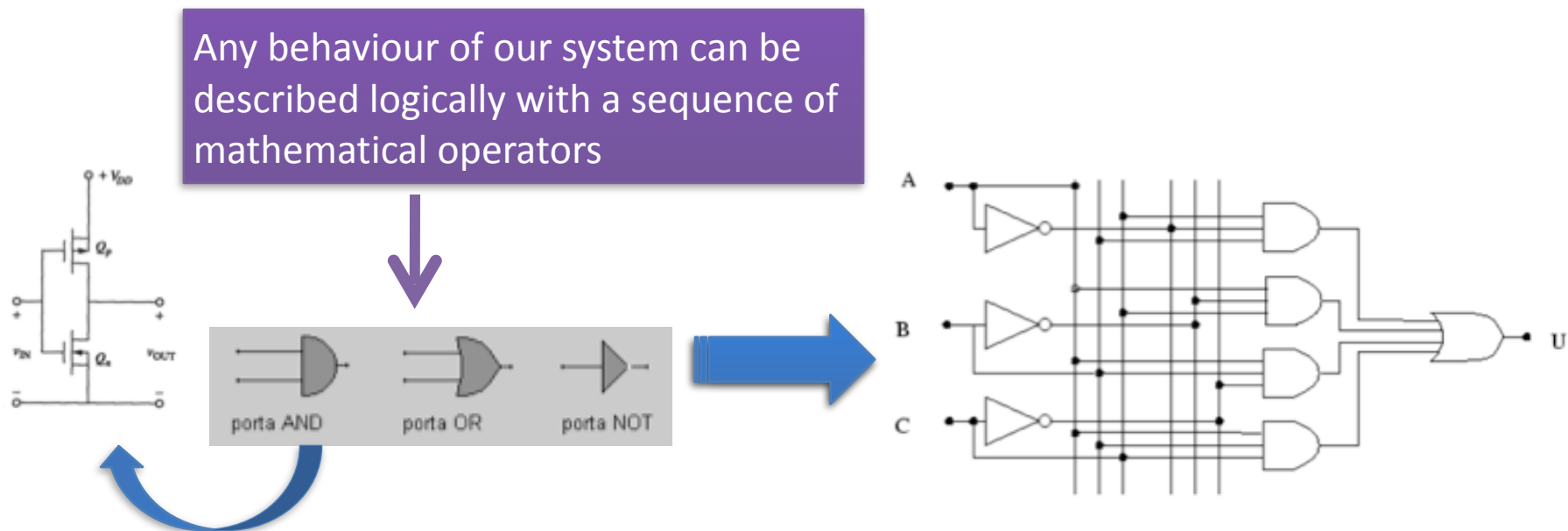


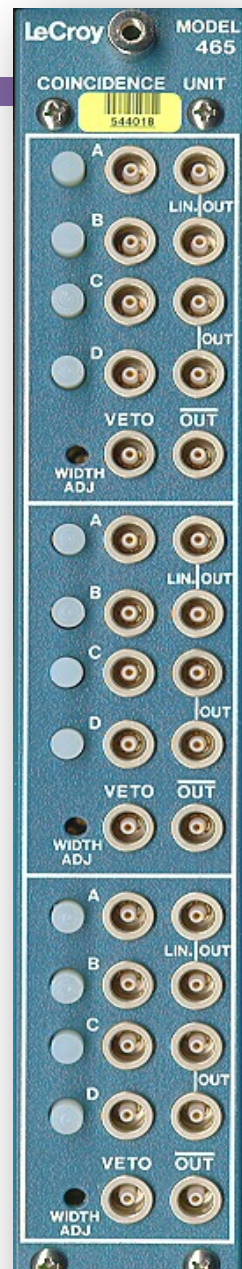
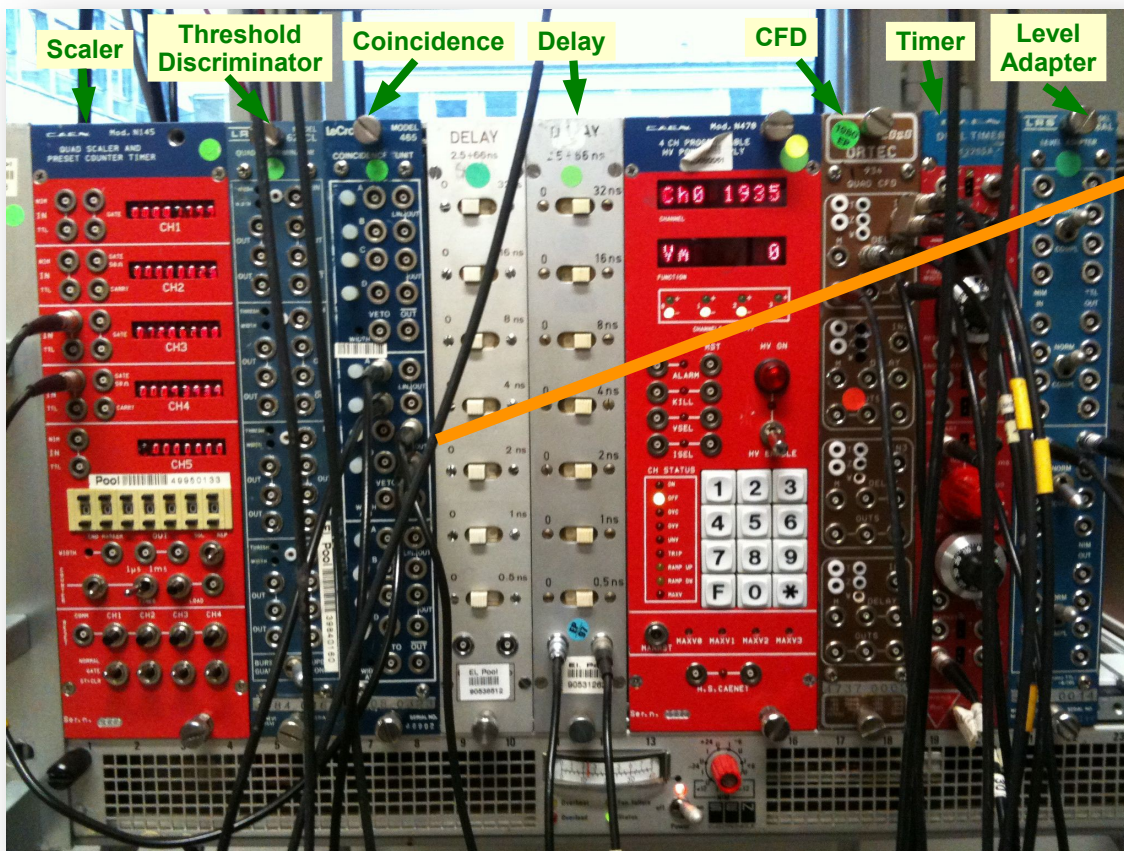


➤ Threshold levels are configurable via screwdriver adjust

Trigger logic implementation

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





LeCroy Coincidence Unit

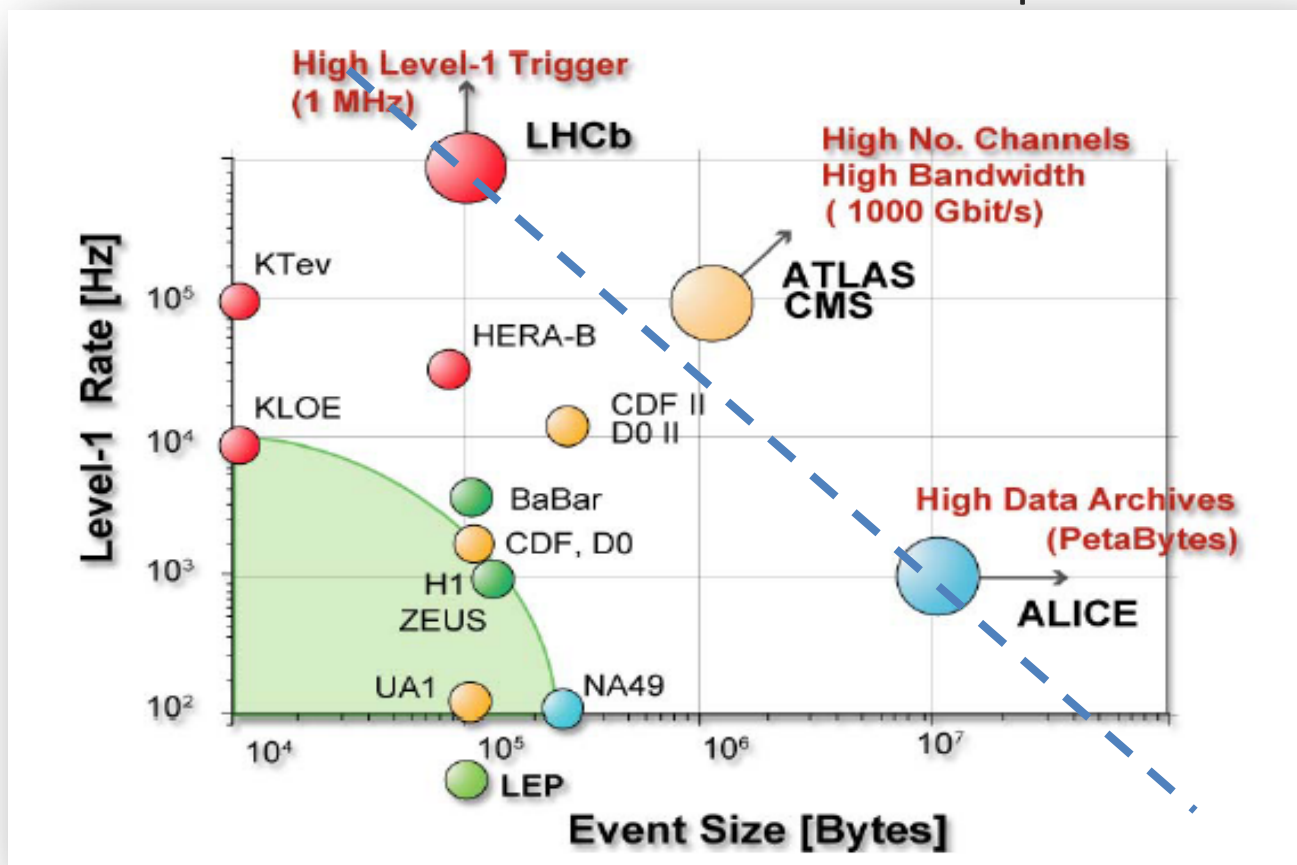
Summary of the trigger requirements

- High Efficiency
 - Low dead-time
 - Fast decision
- Reliability and robustness
- Flexibility

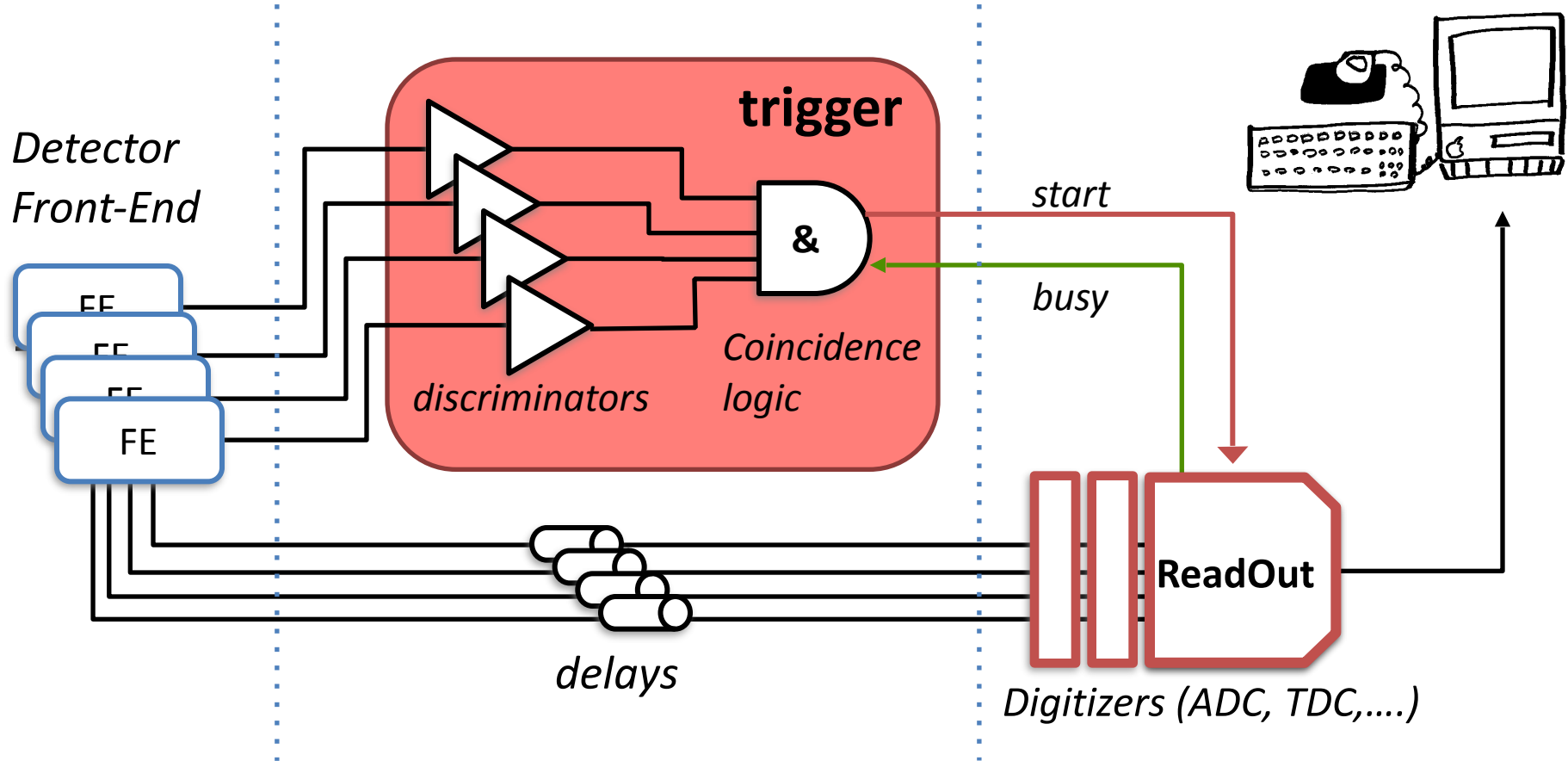
Trigger and data acquisition trends

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

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



A simple trigger system

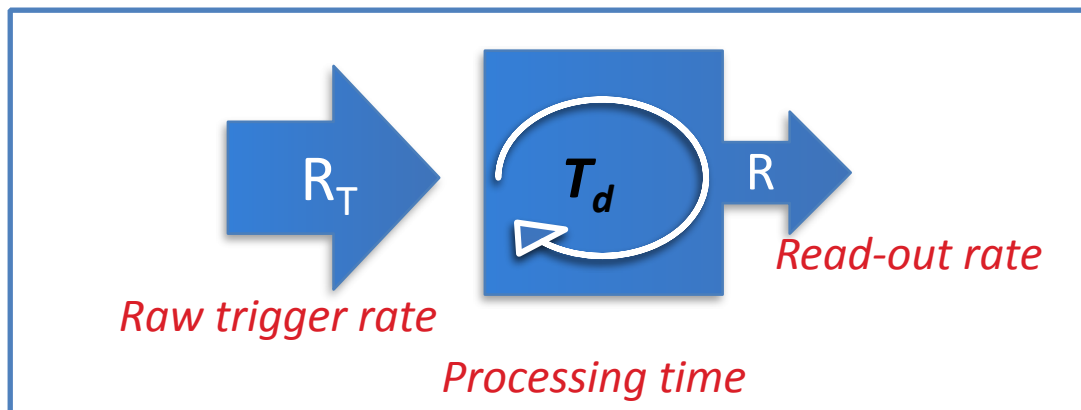


- Due to **fluctuations**, the incoming rate can be higher than processing one
- Valid interactions can be rejected due to system **busy**

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
- **Fluctuations produce dead-time!**

Affects efficiency!



Maximise recording rate

R_T = Trigger rate (average)

R = Readout rate

T_d = processing time of one event

fraction of lost events = $R \times T_d$

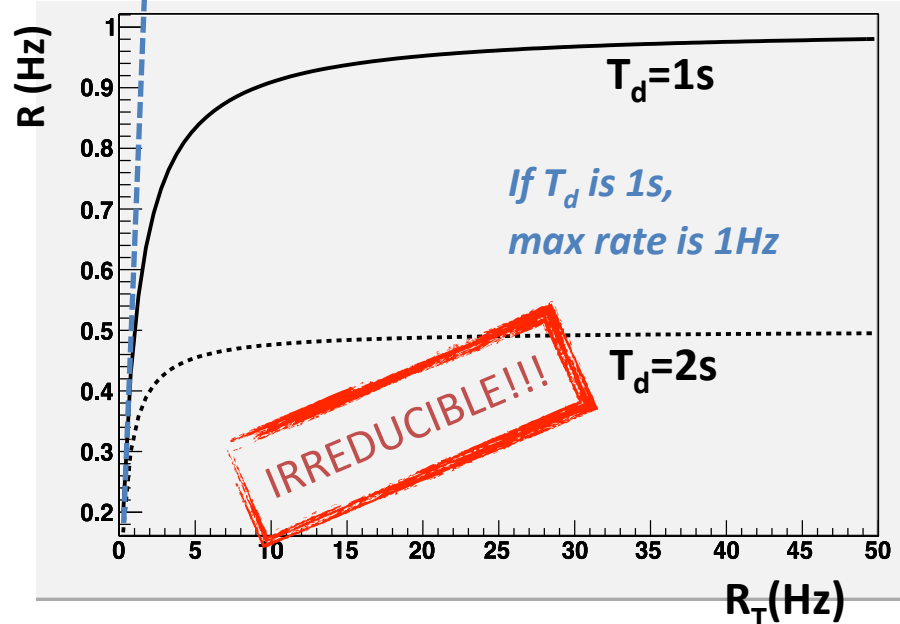
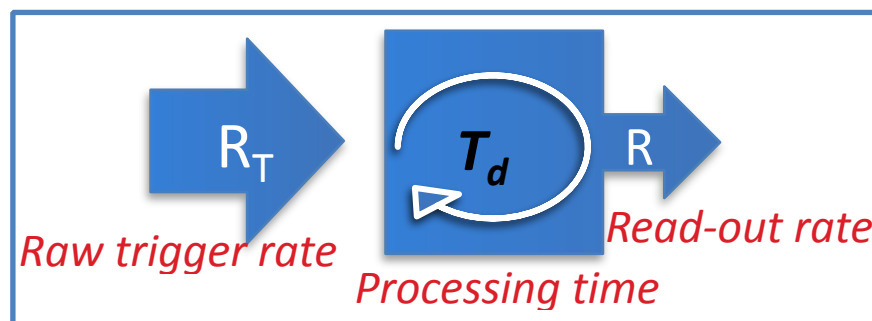
number of events read: $R = (1 - R \times T_d) \times R_T$

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

Fraction of surviving events!

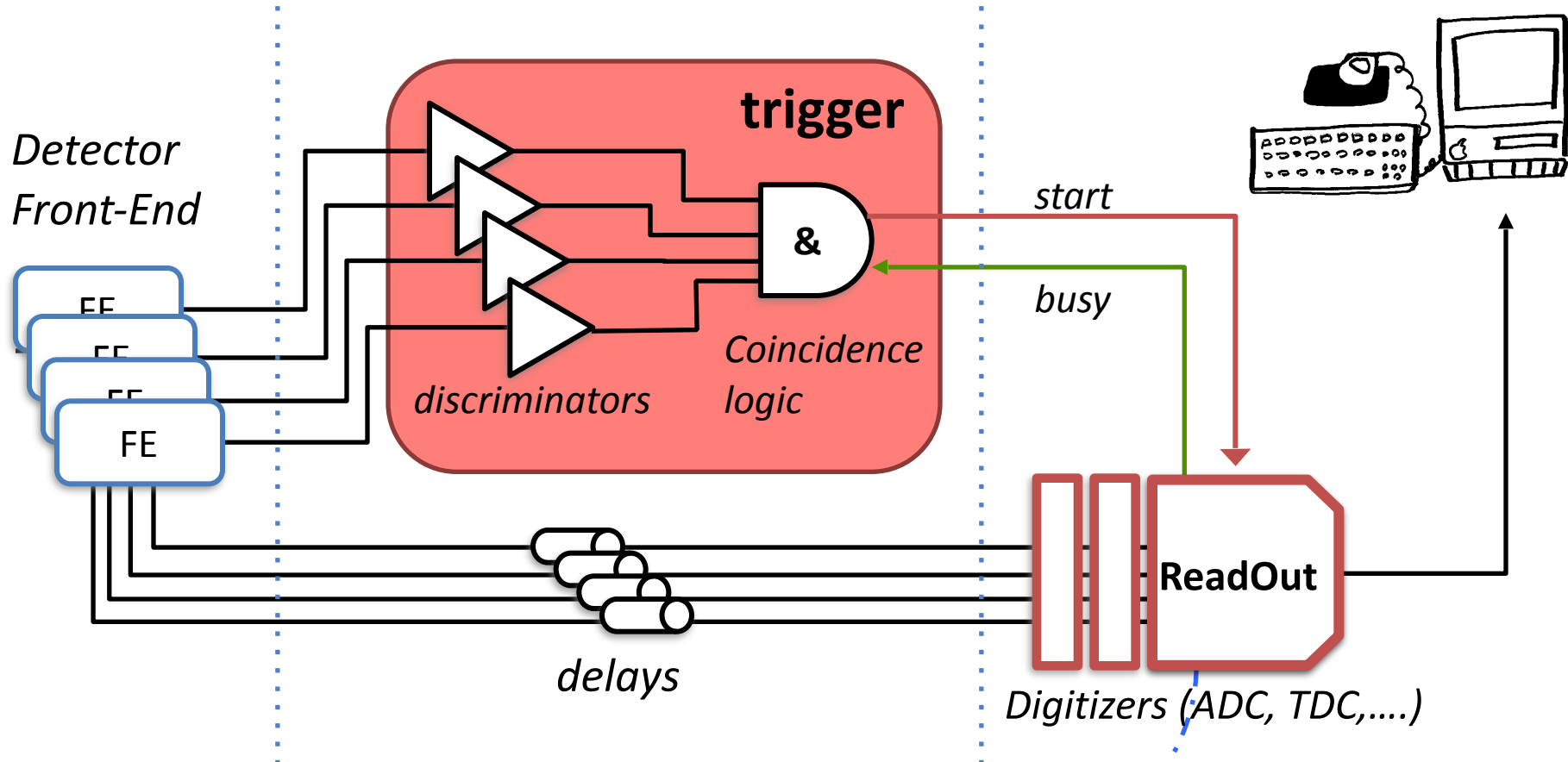
- We always lose events if $R_T > 1/T_d$
- If exactly $R_T = 1/T_d \rightarrow$ dead-time is 50%

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



FAST TRIGGER!
LOW INPUT RATE!

A simple trigger system



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

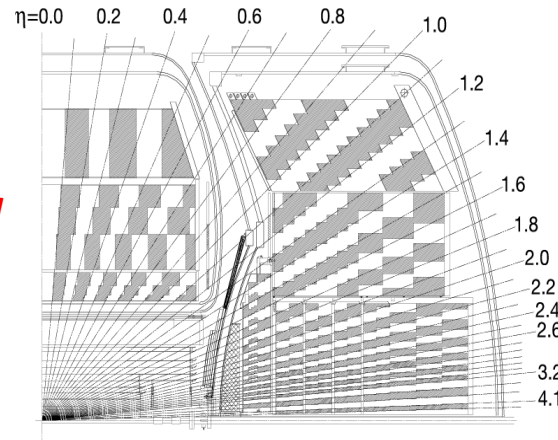
Fraction of lost events due to finite readout

Features to minimize dead-time

➤ 1: Parallelism

- Independent readout and trigger processing paths, one for each sensor element
- Digitization and DAQ processed in parallel (as many as affordable!)

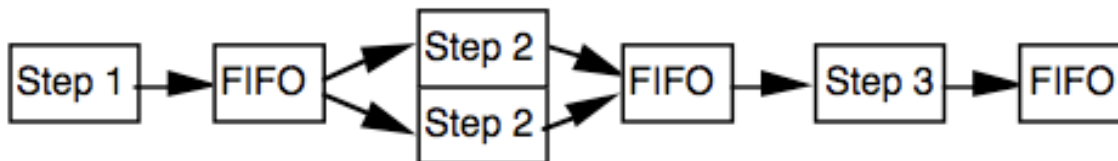
Segment as much as you can!



DZero calorimeters showing the transverse and longitudinal segmentation pattern

➤ 2: Pipeline processing to absorb fluctuations

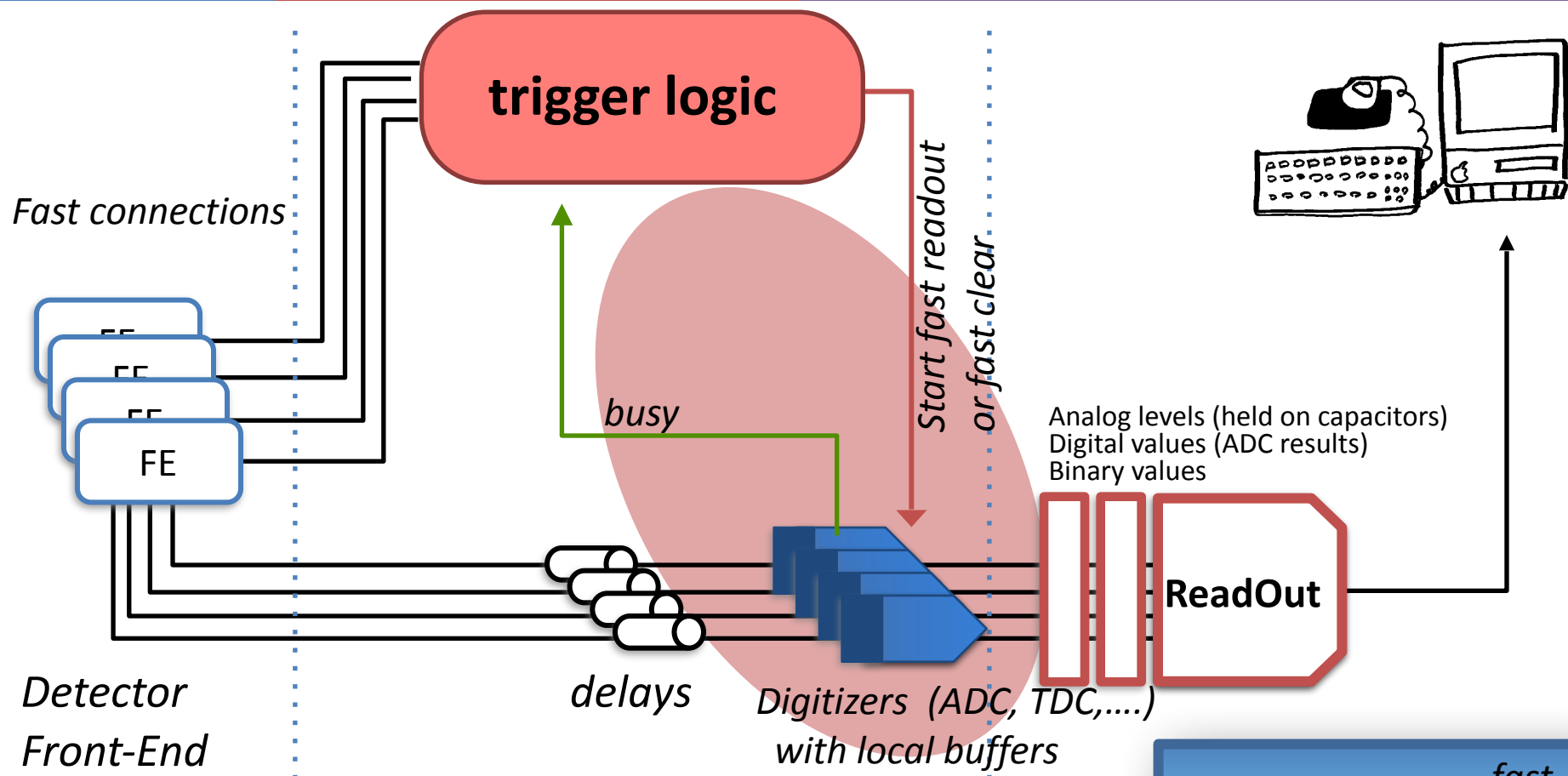
- Organize the process in different steps
- Use local **buffers** between steps with different timing



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

Try to absorb in capable buffers

Minimizing readout dead-time...

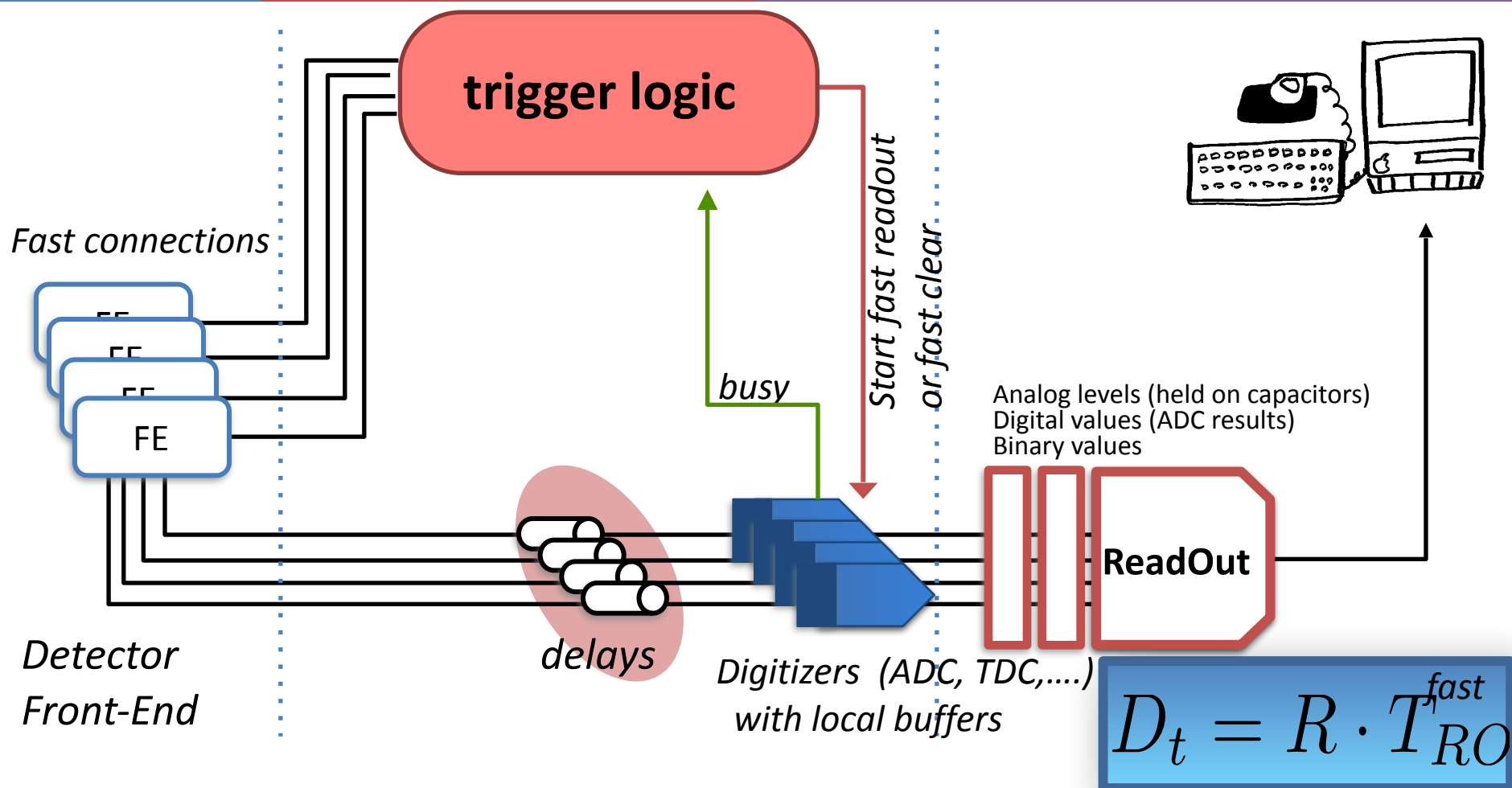


$$D_t = R \cdot T_{RO}^{fast}$$

➤ **Parallelism:** Use multiple digitizers

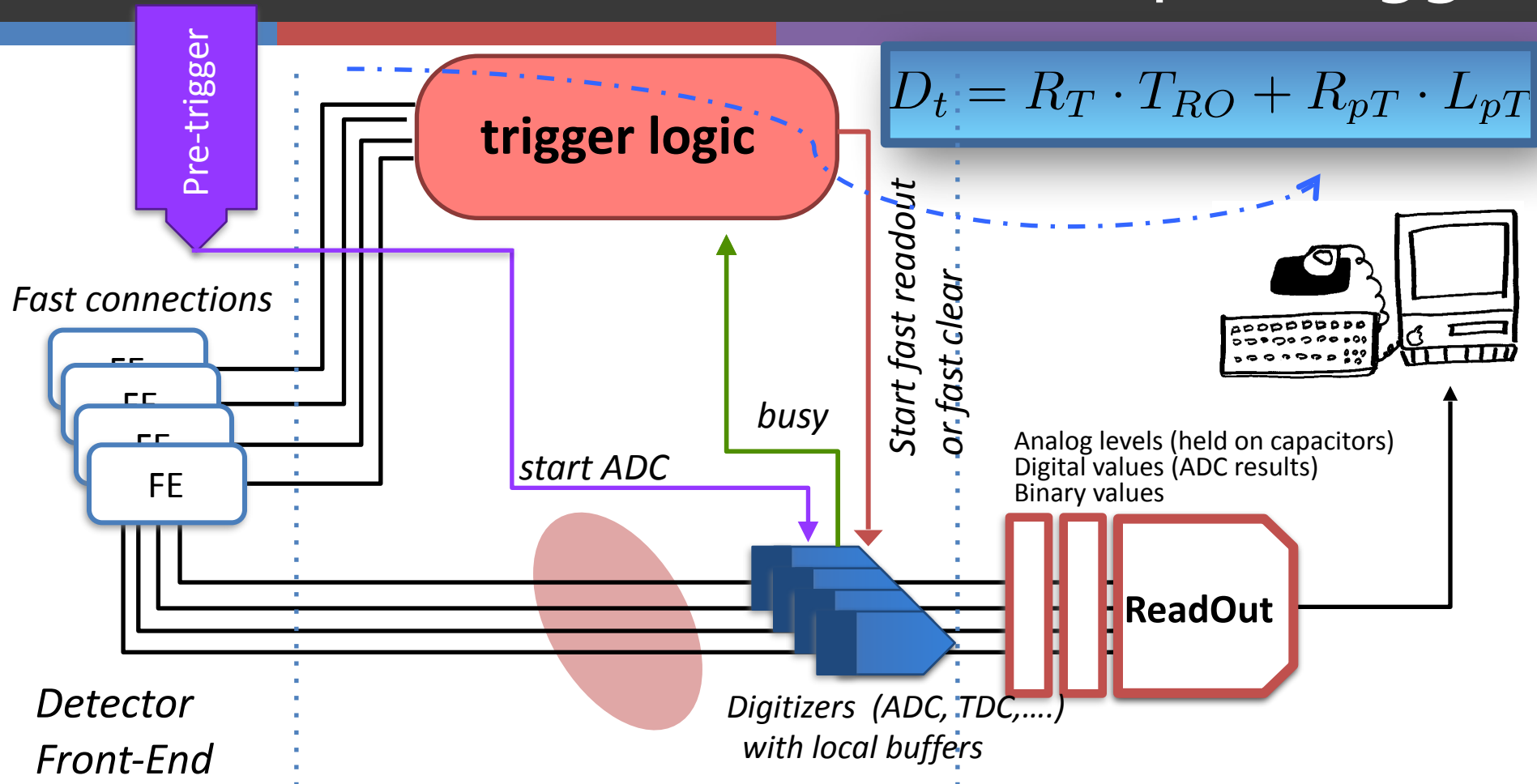
➤ **Pipelining:** Different stages of readout: fast local readout + global event readout (slow)

Trigger latency



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

Add a pre-trigger



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

Coupling trigger rate and readout

- 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

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

$i=1$ is the pre-trigger

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

R_i = Rate after the i -th level

L_i = Latency for the i -th level

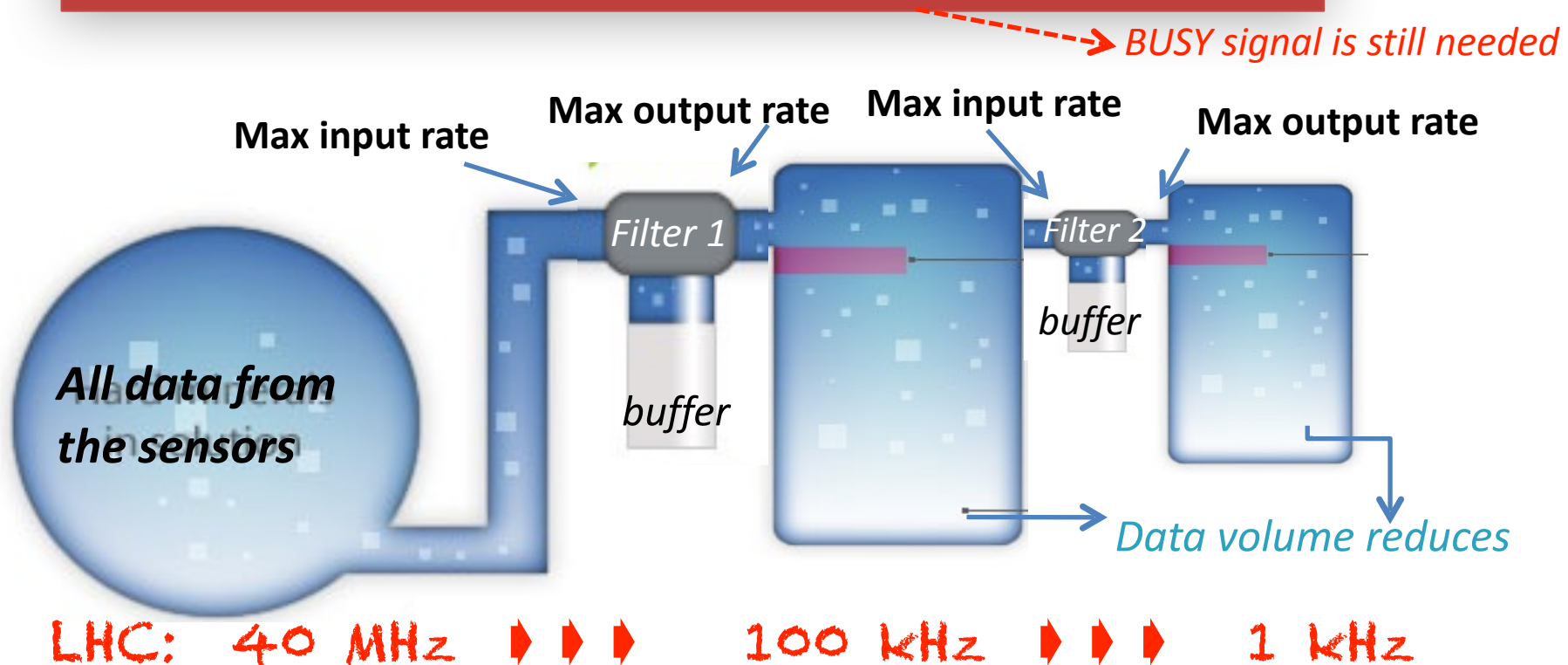
T_{LRO} = Local readout time

Try to minimize each factor!

Buffering and filtering

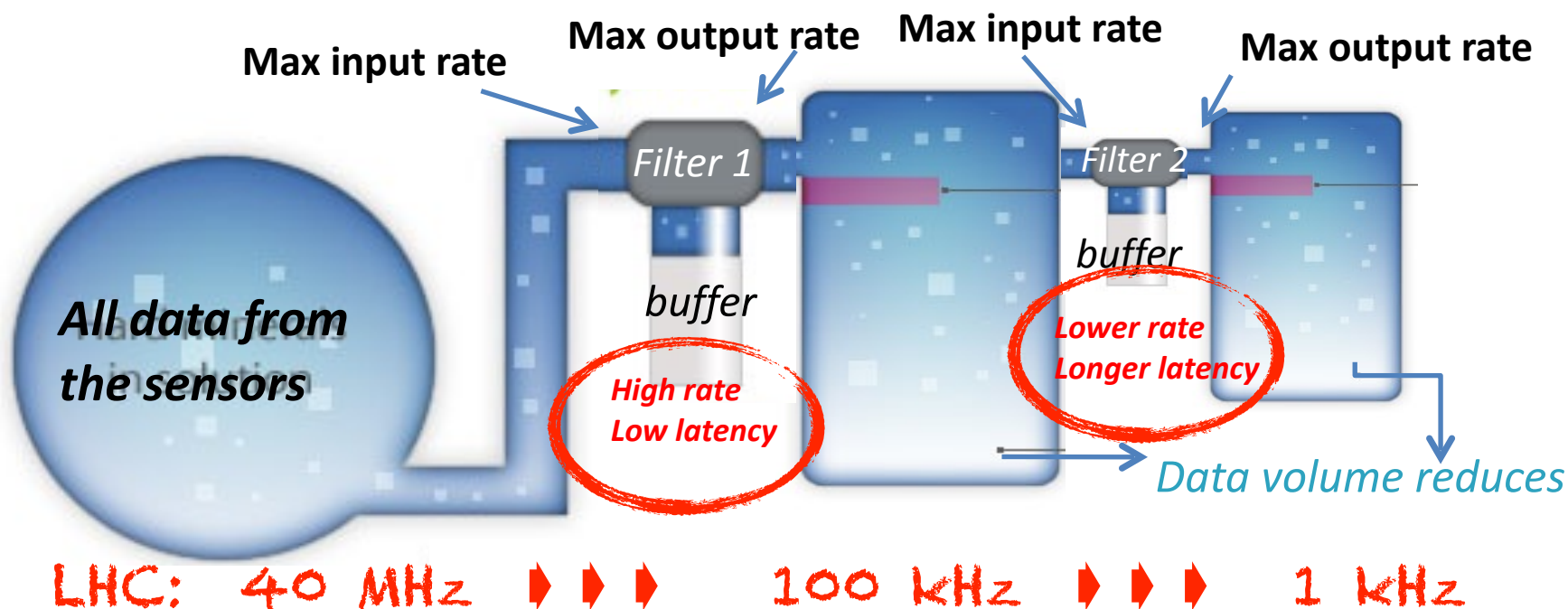
- At each step, data volume is reduced, more refined filtering to the next step
 - The **input** rate defines the filter **processing time** and its **buffer size**
 - The **output** rate limits the maximum latency allowed in the **next step**
 - Filter **power** is limited by the capacity of the next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

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



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)
- Latest filter can have longer latency (more selective)



Multi-level triggers

- Adopted in large experiments
- 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)



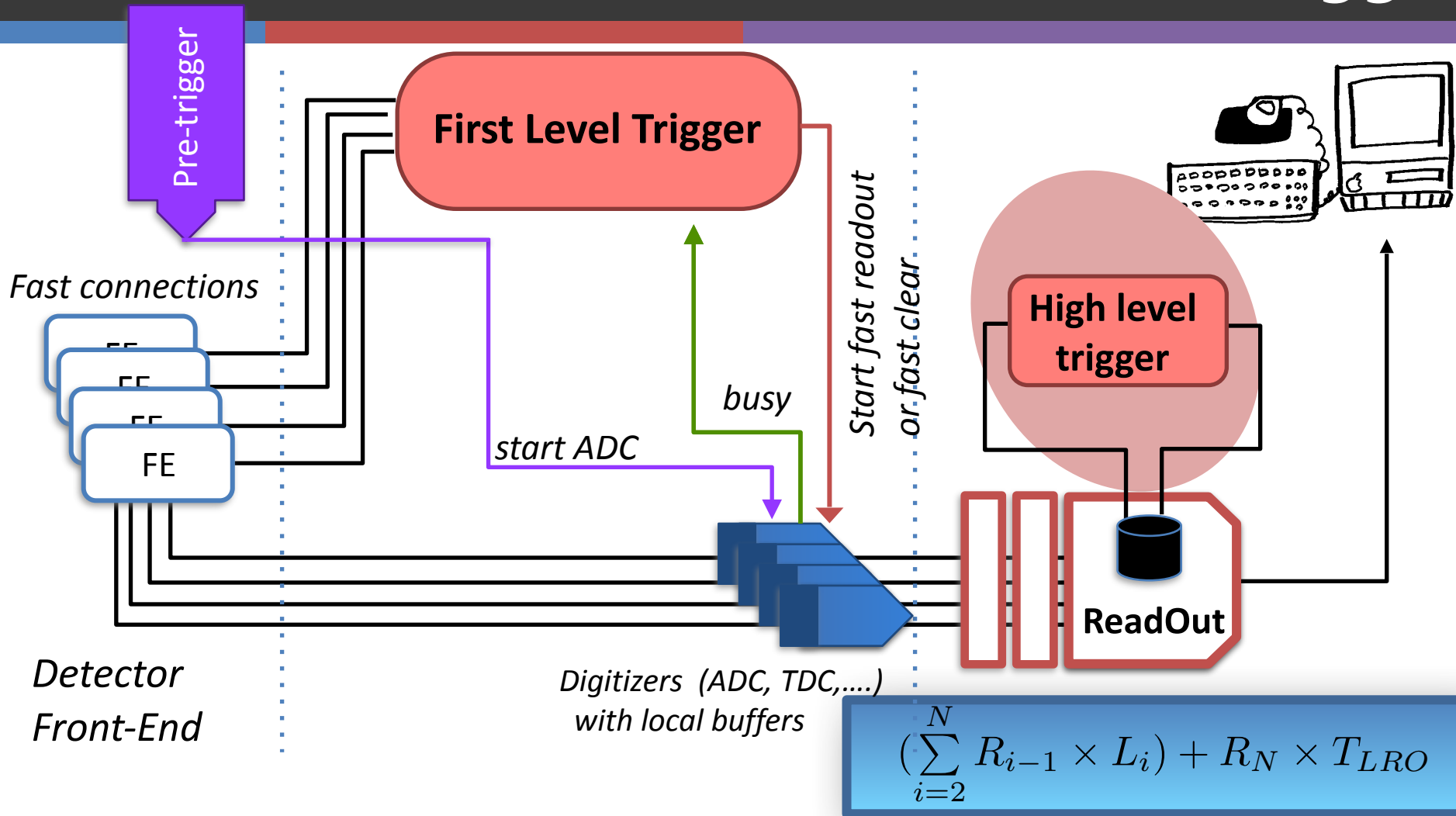
LHC experiments @ Run1

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

*Lower event rate
Bigger event fragment size
More granularity information
More complexity
Longer latency
Bigger buffers*

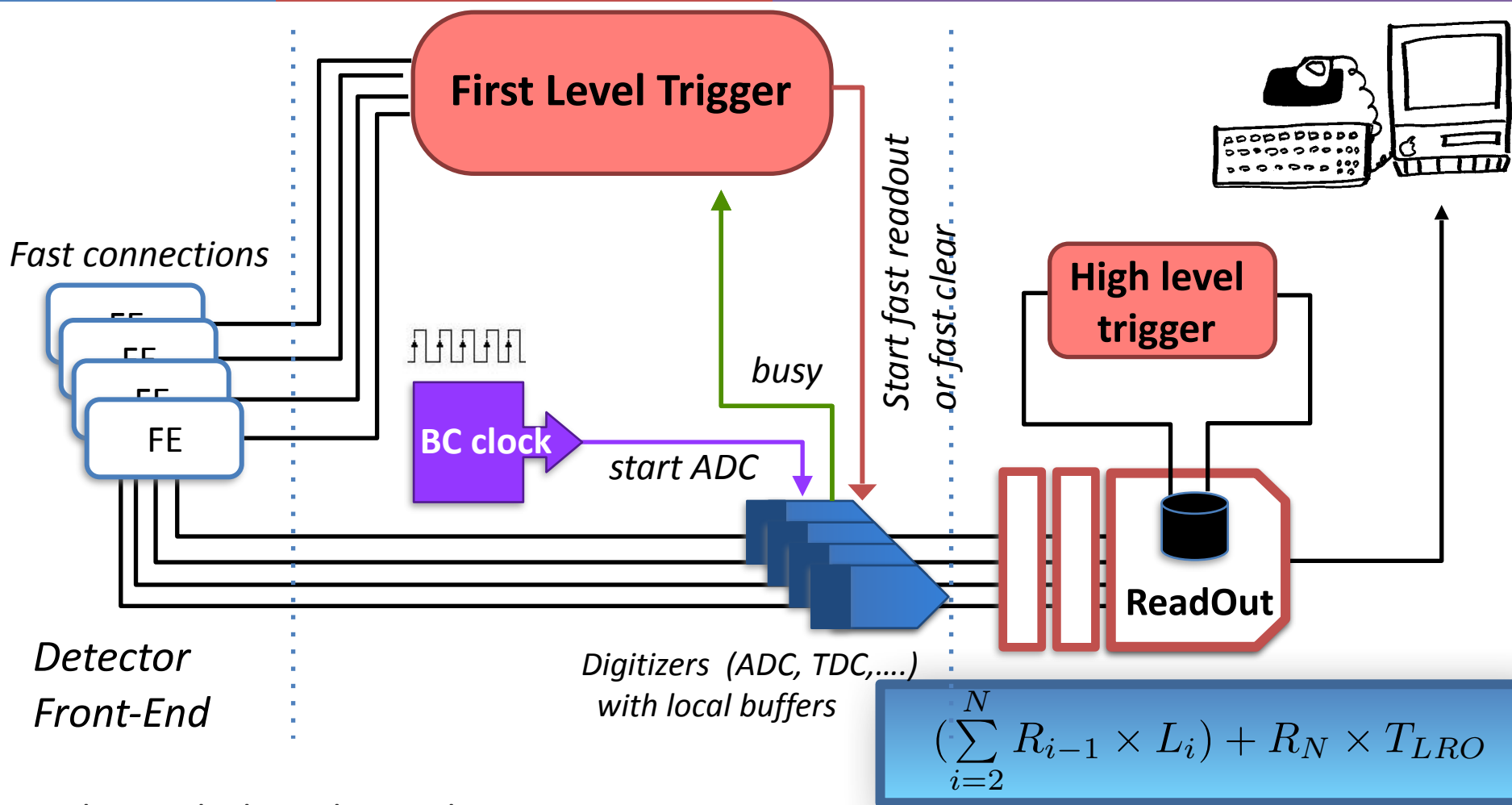
Efficiency for the desired physics must be kept high at all levels, since rejected events are lost for ever

Schema of a multi-level trigger



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

Schema of a multi-level trigger @ colliders



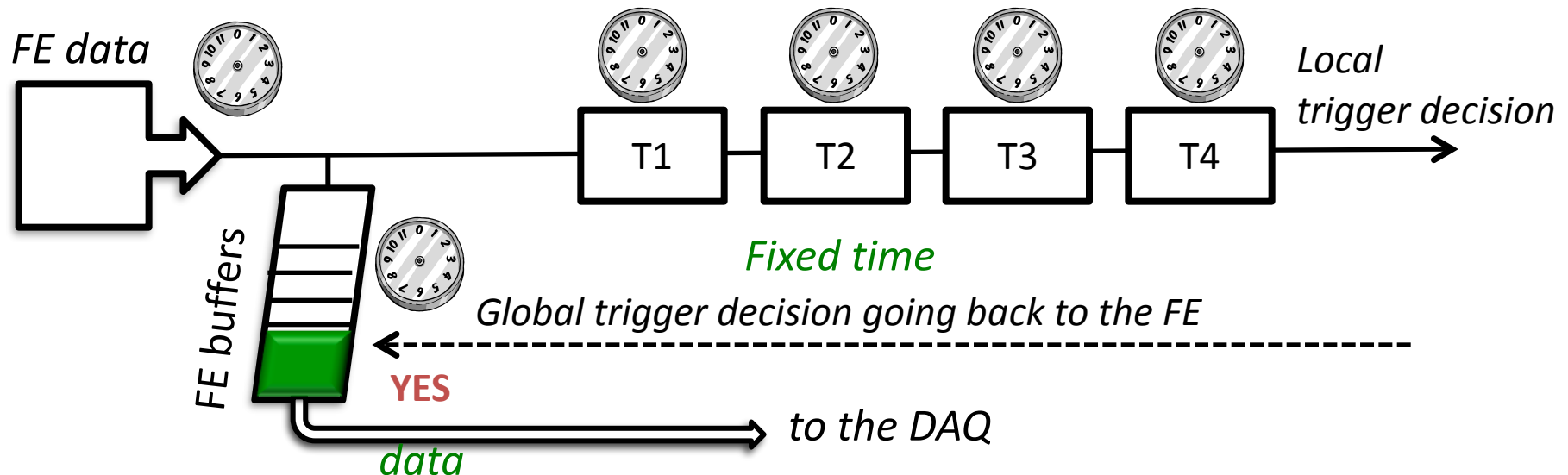
The BC clock can be used as a pre-trigger



First-level trigger is **synchronous** to the collision clock: can use the time between two BCs to make its decision, without dead-time, if it's long enough

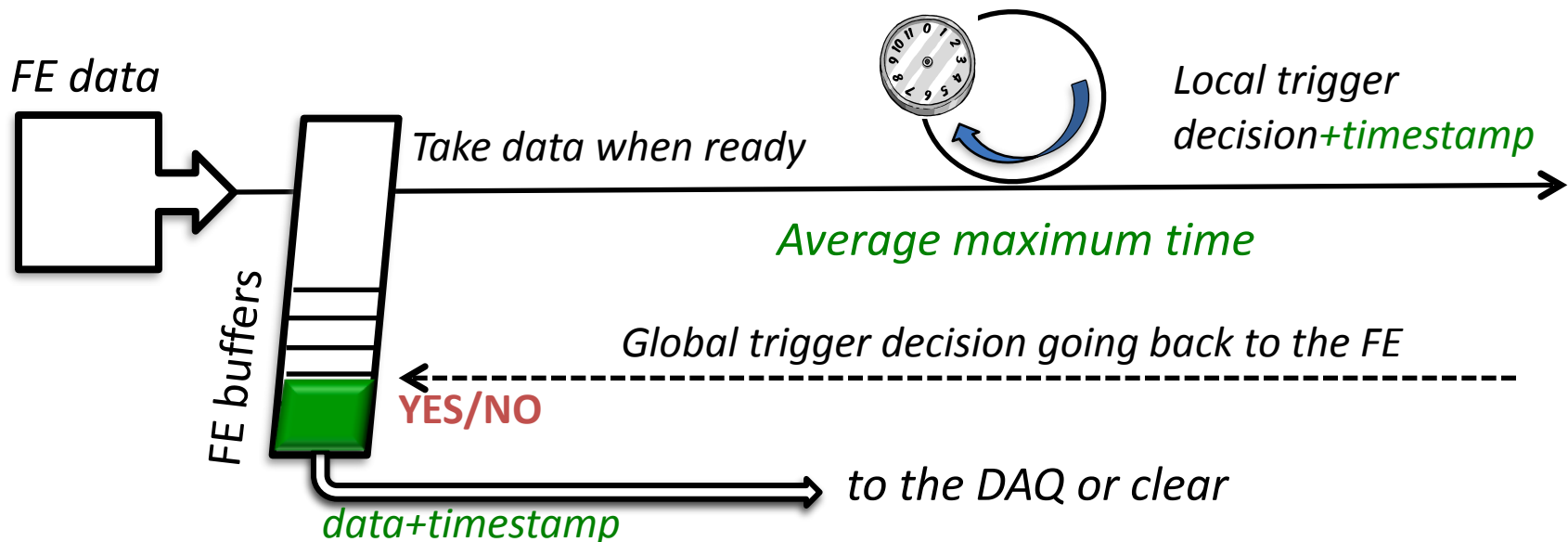
Synchronous or asynchronous?

- **Synchronous:** operations in phase with a clock
 - All trigger data move in lockstep with the clock through the trigger chain
 - Fixed latency
 - The data, held in storage pipelines, are either sent forward or discarded
 - If buffer size \neq latency \rightarrow truncated events
 - Used for L1 triggers in collider experiments, making use of the bunch crossing clock
 - **Pro's:** dead-time free (just few clock cycles to protect buffers)
 - **Con's:** cost (high frequency stable electronics, sometimes needs to be custom made); maintain synchronicity throughout the entire system, complicated alignment procedures if the system is large (software, hardware, human...)



Synchronous or asynchronous?

- **Asynchronous:** operations start at given conditions (when data are ready or last processing is finished)
 - Used for larger time windows
 - Average latency (with large buffers to absorb fluctuations)
 - If buffer size \neq dead-time \rightarrow lost events
 - Used also for software filters
 - **Pro's:** more robust against bursts of data; running on conventional CPUs
 - **Con's:** needs a timing signal synchronised to the FE to latch the data, needs time-marker stored in the data, data transfer protocol is more complex



Level-1: reduce the latency

- Pipelined trigger
- Fast processors
- Fast data movement



Chose your detector

➤ Use analogue signals from existing detectors or dedicated “trigger detectors”

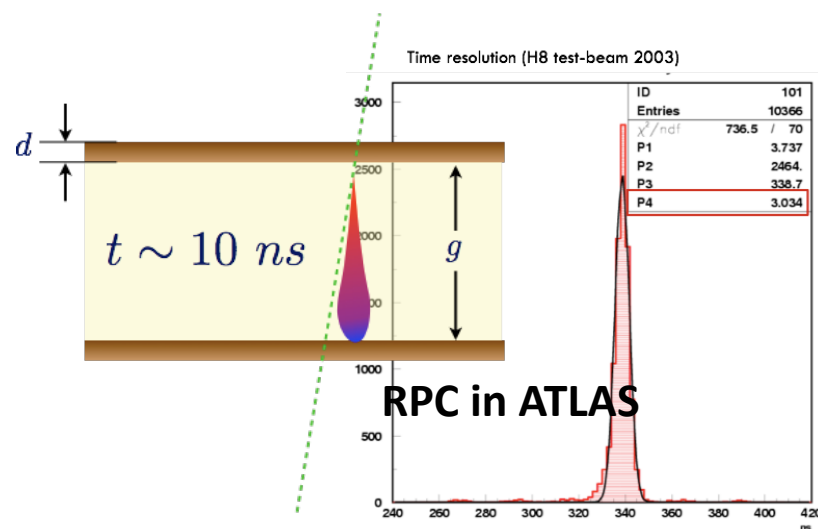
- Organic scintillators
- Electromagnetic calorimeters
- Proportional chambers (short drift)
- Cathode readout detectors (RPC,TGC,CSC)

➤ With these requirements

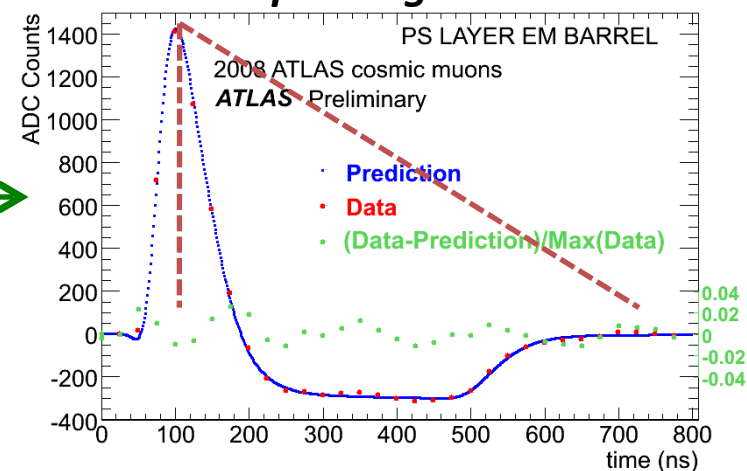
- **Fast signal:** good time resolution and low jittering
 - Signals from slower detectors are shaped and processed to find the unique peak (**peak-finder algorithms**)

- **High efficiency**
- **(often) High rate capability**

➤ **Need optimal FE/trigger electronics to process the signal**



ATLAS Liquid Argon calorimeter



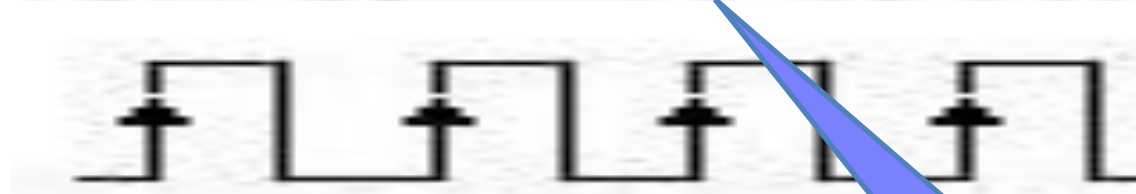
Synch level-1 trigger @ colliders

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

LEP: 22 μ s



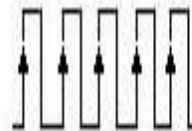
Tevatron: 396 ns



HERA: 96 ns



LHC: 25 ns

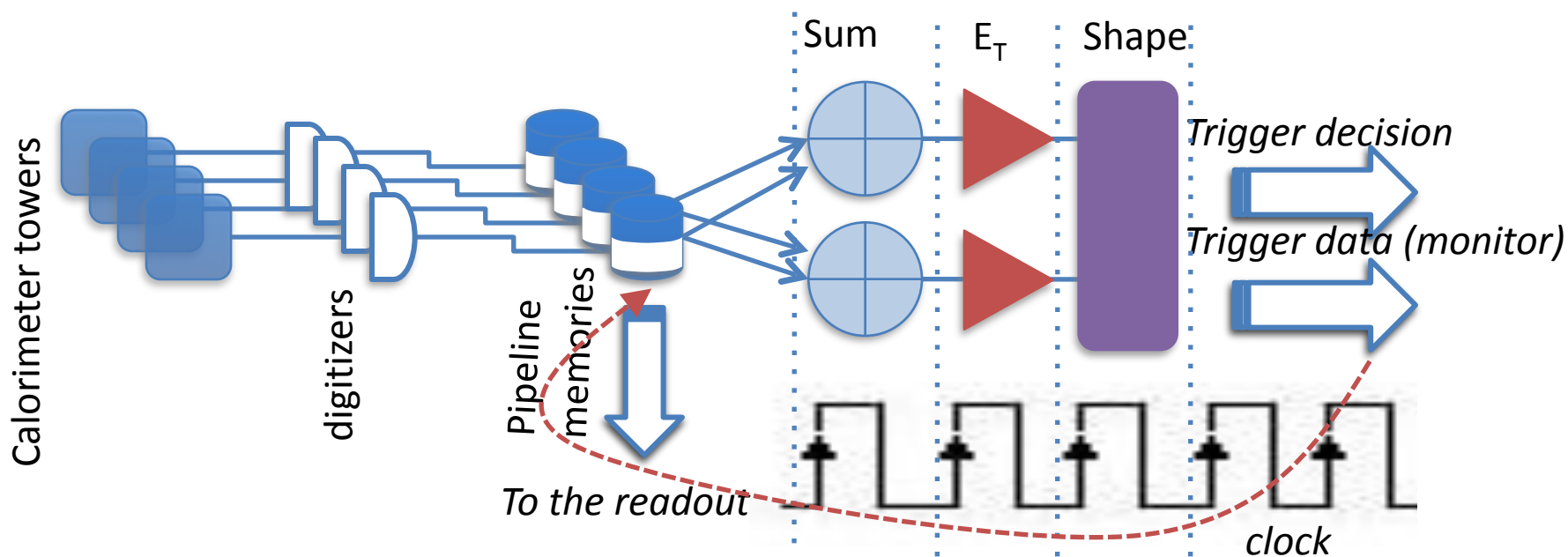


LEP
output rate = 7 Hz
trigger latency = 38 μ s
readout time = 2.5 ms
dead time 2%

- @LEP, BC interval 22 μ s: complicated trigger processing was allowed
- In modern colliders: required high luminosity is driven by high rate of BC
 - It's not possible to make a trigger decision within this short time!

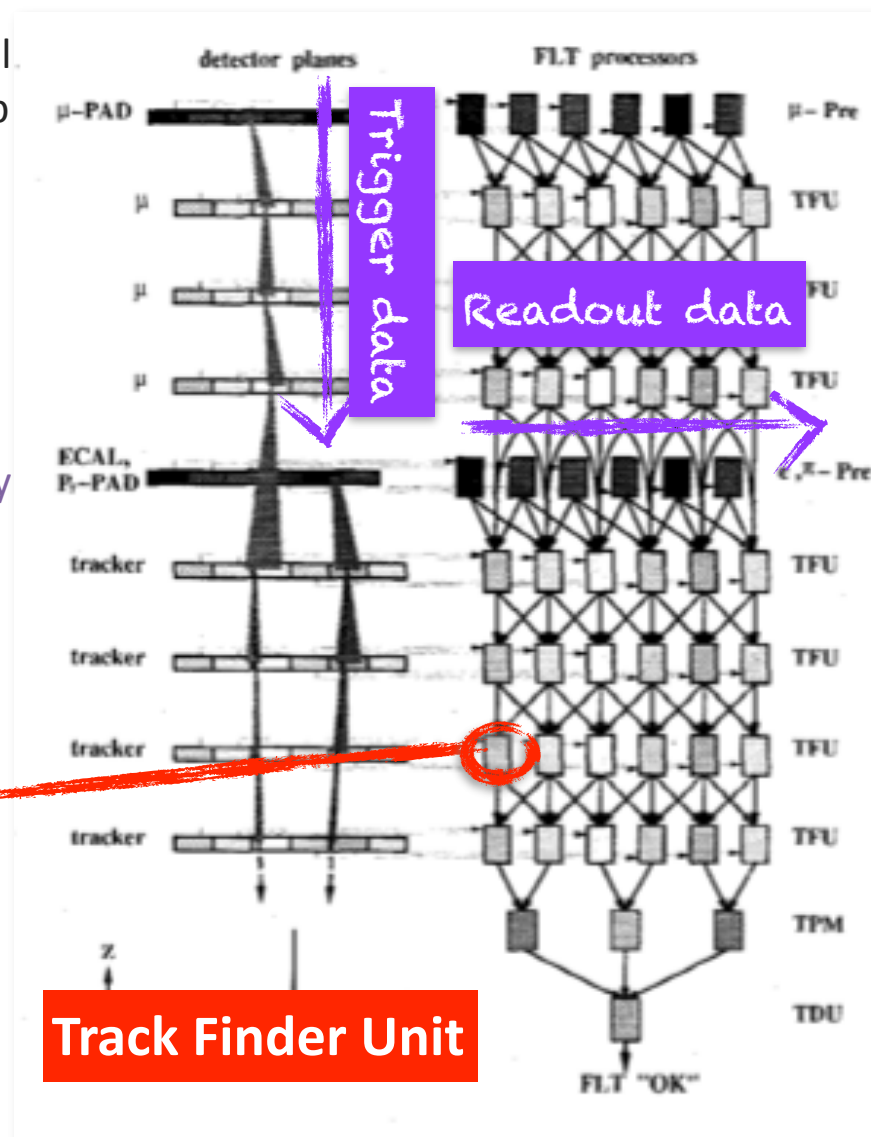
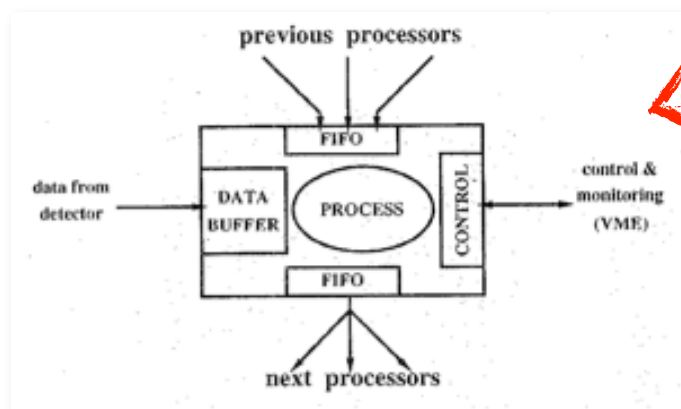
Level-1 pipeline trigger

- With a synchronous system and large buffer pipelines we can allow long **fixed trigger latency (order of μs)**
 - Latency is the sum of each step processing and data transmission time
- Each trigger processor **concurrently** processes many events
 - Divide the processing in steps, each performed within one BC



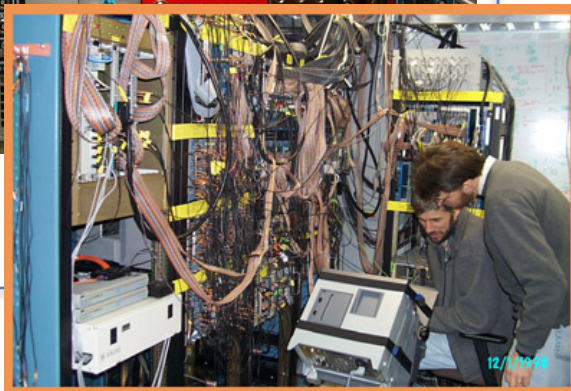
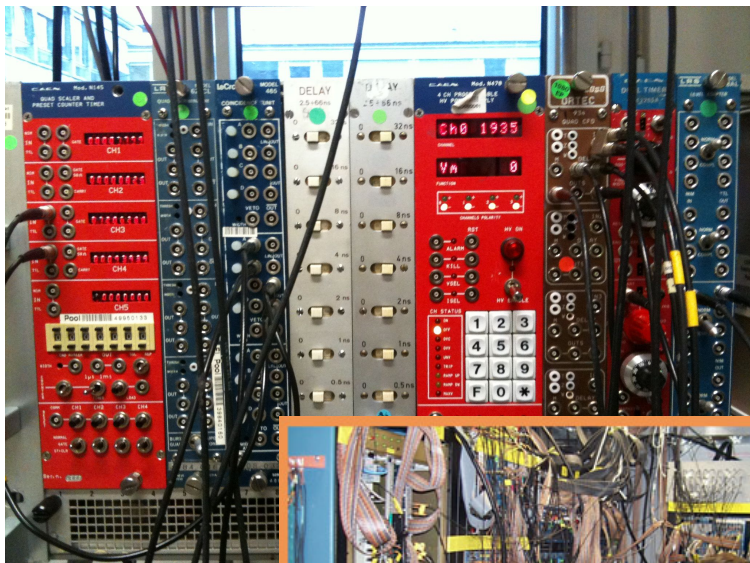
Example: HERA-B track finder

- **Iterative algorithm:** each step processes only a small **Region of Interest (RoI)** defined by the previous step
 - Each unit handles only the hit information corresponding to a small part of the detector
 - Only units whose region is touched by the RoI will process it
- **Two data streams:**
 - **Detector data** transferred to on-board memory **synchronously** with BC clock (left to right)
 - **RoI data** transferred **asynchronously** from unit to unit (top to bottom)

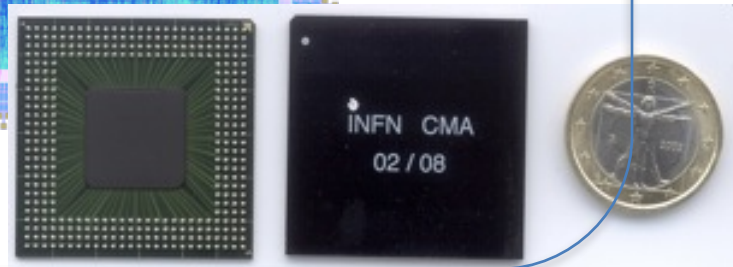
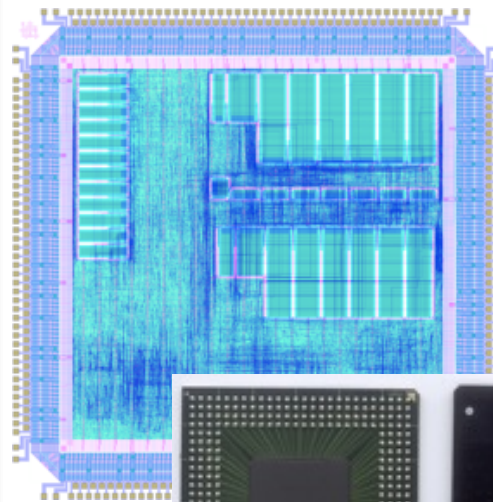


Choose your L1 trigger system

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



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



Level-1 trigger processors

- Requirements at high trigger rates
 - Fast processing
 - Flexible/programmable algorithms
 - Data compression and formatting
 - Monitor and automatic fault detection

- Digital integrated circuits (IC)

- Reliability, reduced power usage, reduced board size and better performance

- Different families on the market:

- **Microprocessors** (CPUs, GPGPUs, ARMs, DSP=digital signal processors..)

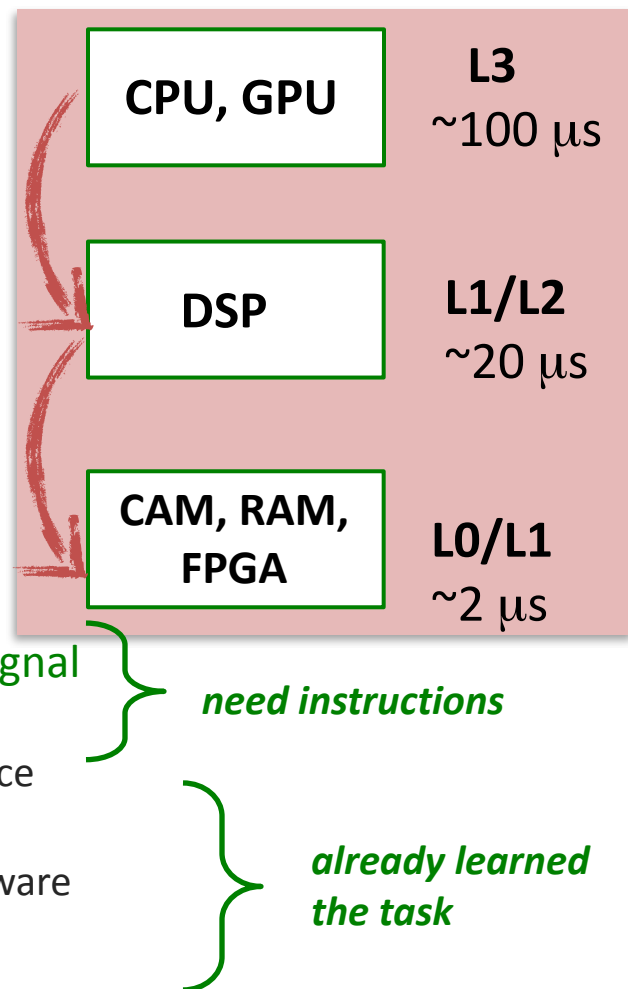
- Available on the market or specific, programmed only once

- **Programmable logic devices** (FPGAs, CAMs,...)

- More operations/clock cycle, but costly and difficult software developing

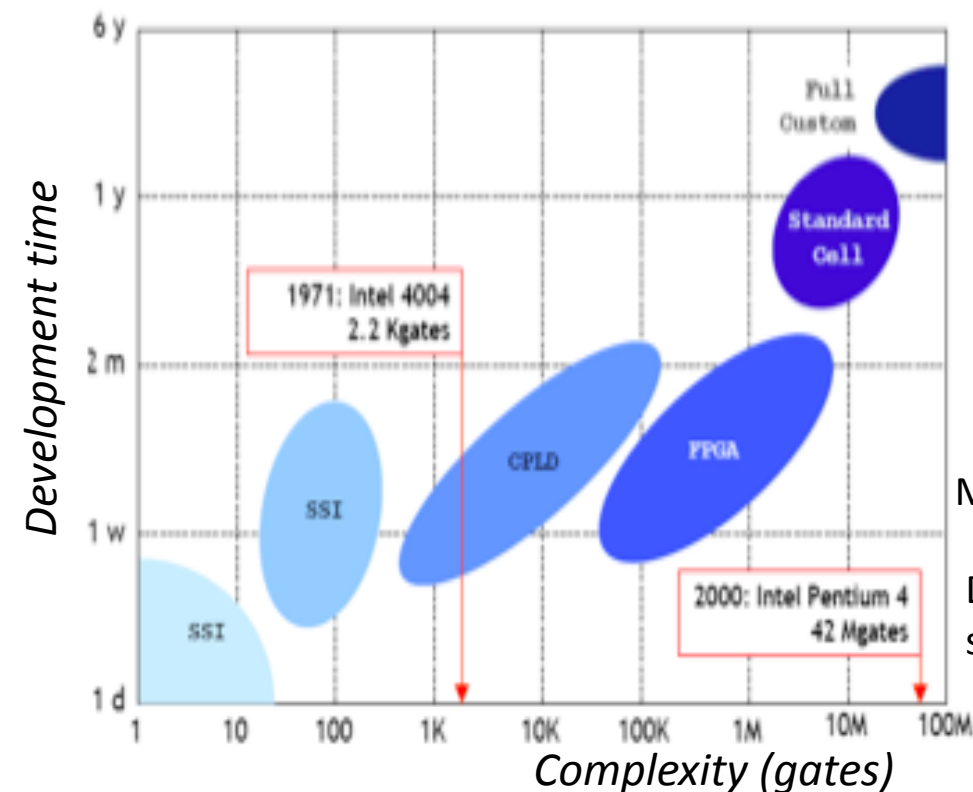
- **New trend is the integration of both:**

- Using standard interface (ethernet), can profit of standard software tools (like for Linux or real-time) and development time is reduced



Custom trigger processors?

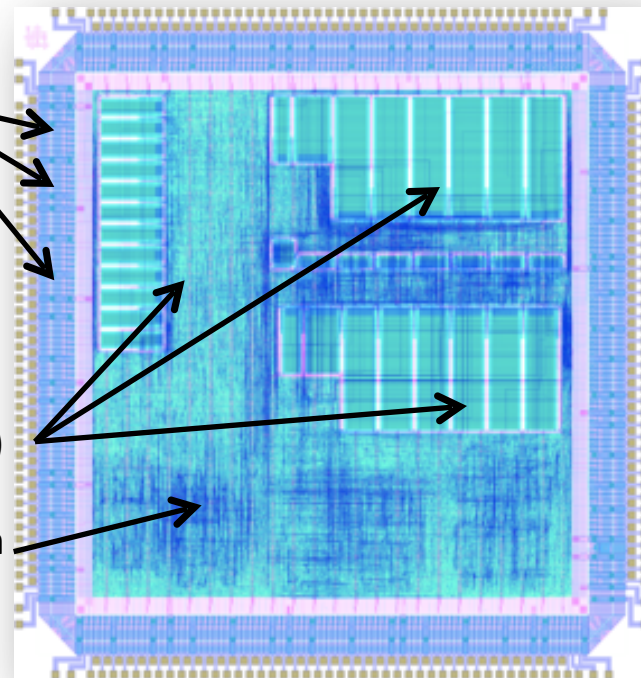
- Application-specific integrated circuits (**ASICs**): optimized for fast processing (Standard Cells, full custom)
 - Intel processors, ~ GHz
- Programmable ASICs (like Field-programmable gate arrays, **FPGAs**)
 - Easily find processors @ 100 MHz on the market (1/10 speed of full custom ASICs)



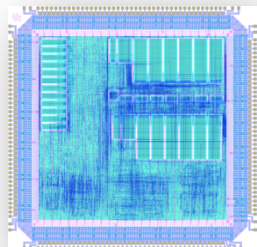
I/O PADs

Microcells (RAM)

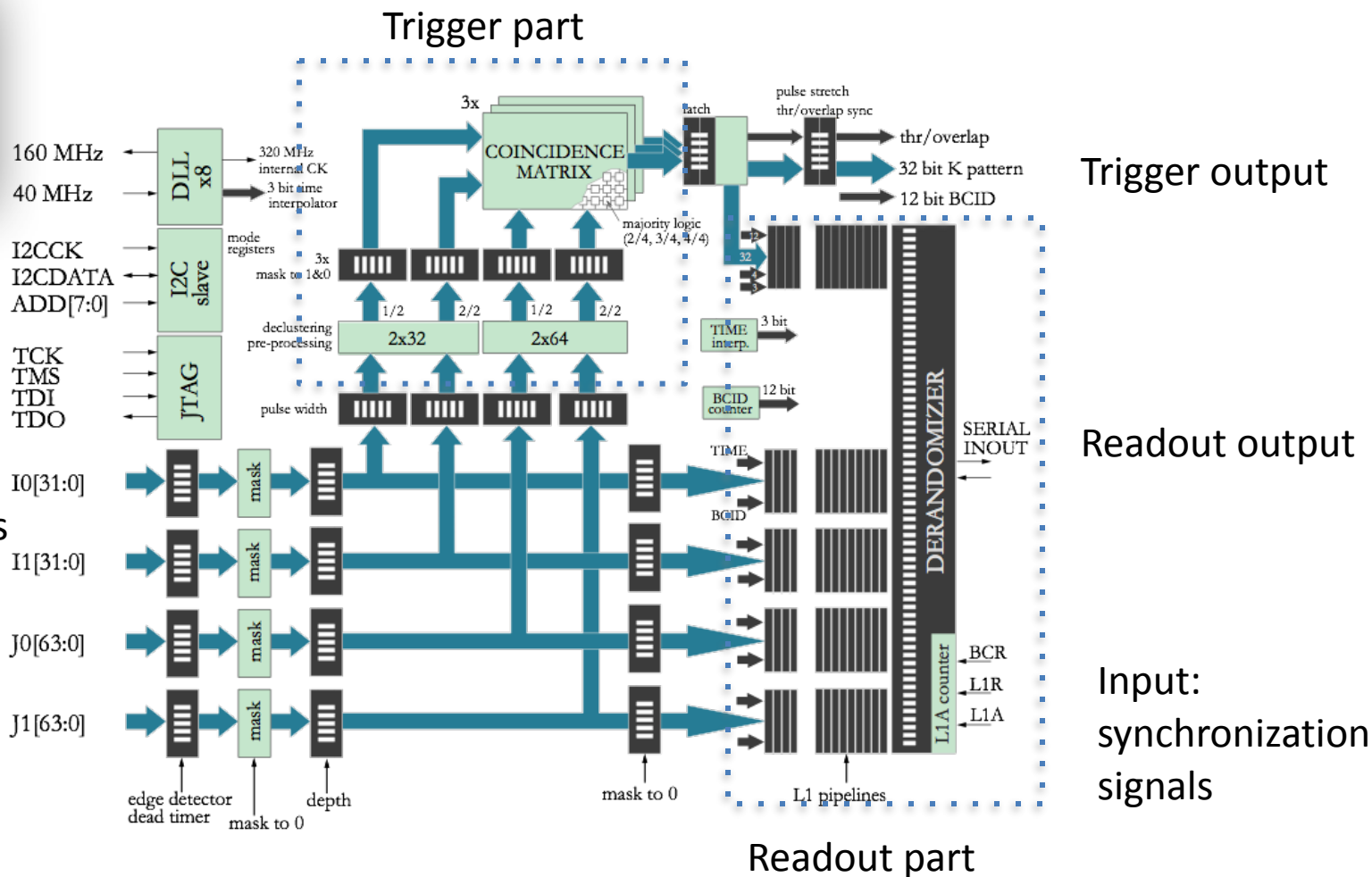
Digital logic with standard cells



Example: logic of a trigger ASIC



Digital inputs
from
detector

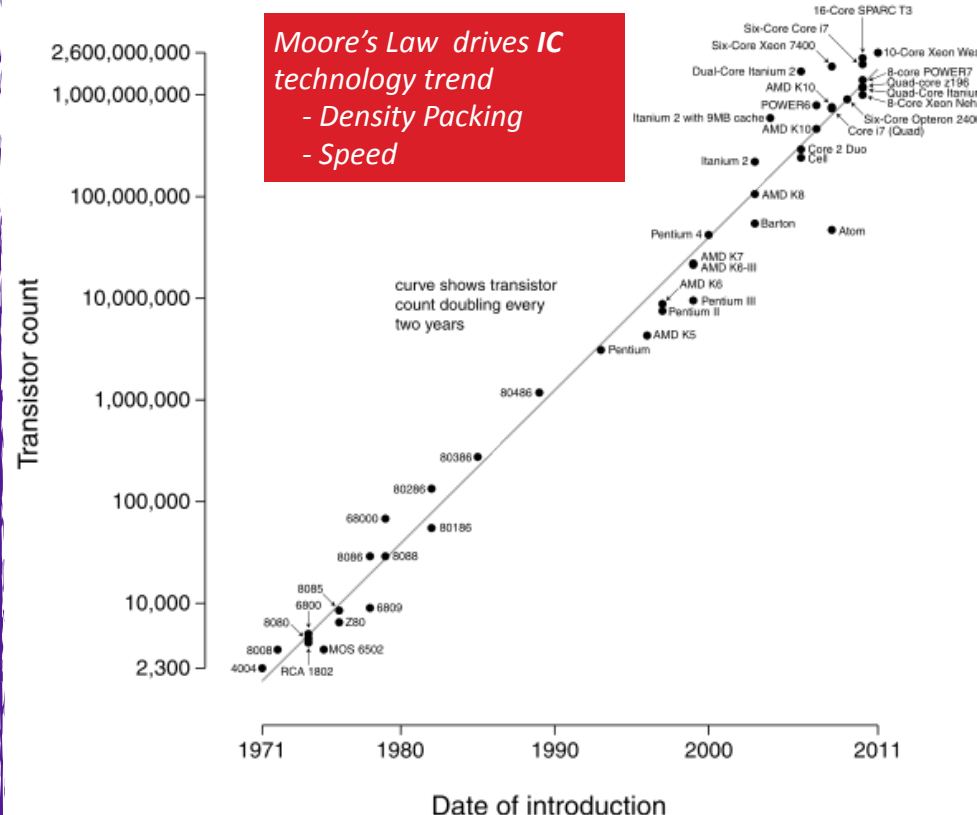


Coincidence Matrix ASIC for Muon Trigger in the Barrel of ATLAS

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>

Microprocessor Transistor Counts 1971-2011 & Moore's Law



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

Data movement technologies

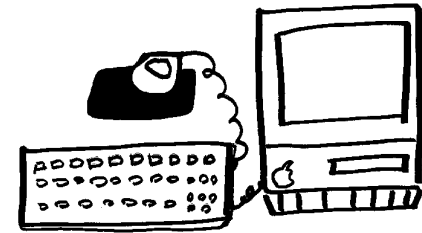
- Faster data processing are placed on-detector (close or joined to the FE)
- Intermediate crates are **good separation** between FE (long duration) and PCs

On-detector



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

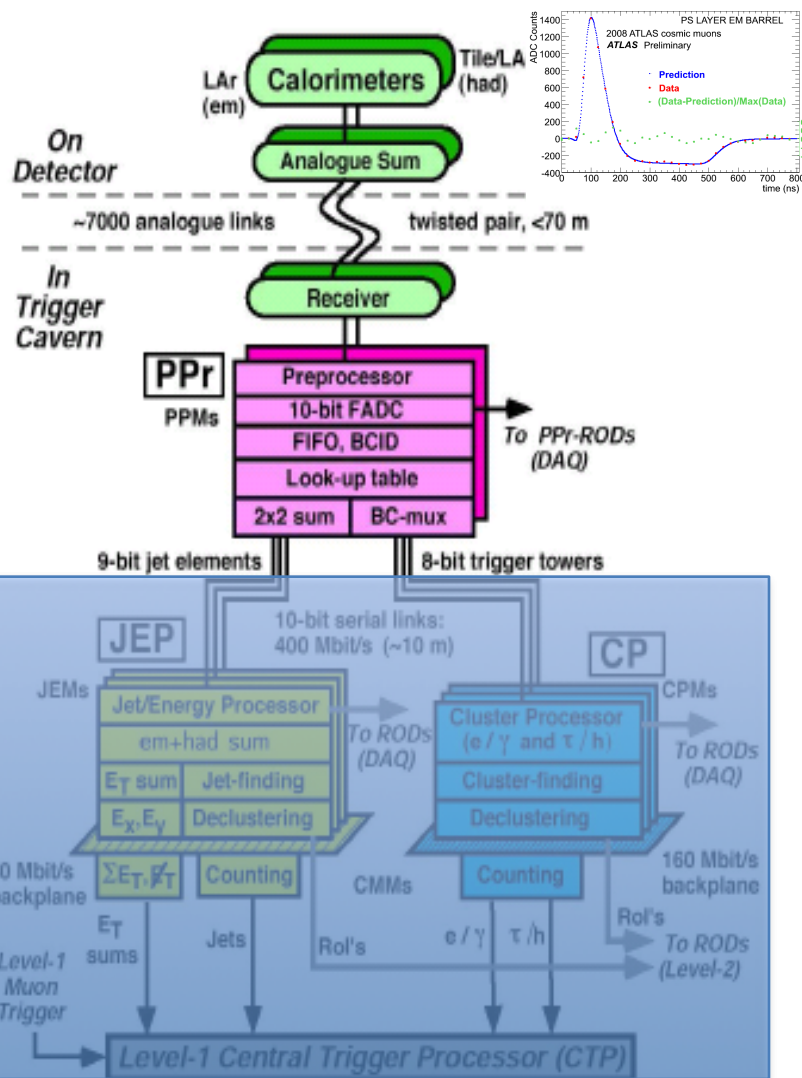
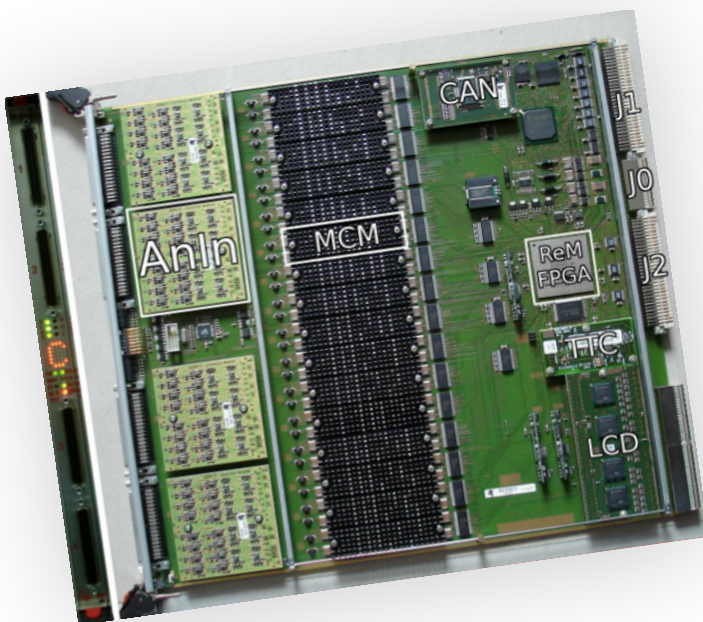
Off-detector



- **High-speed** serial links, electrical and optical, over a variety of distances
 - Low cost and low-power LVDS links, @400 Mbit/s (up to 10 m)
 - Optical GHz-links for longer distances (up to 100 m)
- **High density** backplanes for data exchanges within crates
 - High pin count, with point-to-point connections up to 160 Mbit/s
 - Large boards preferred

Example : ATLAS calorimeter trigger

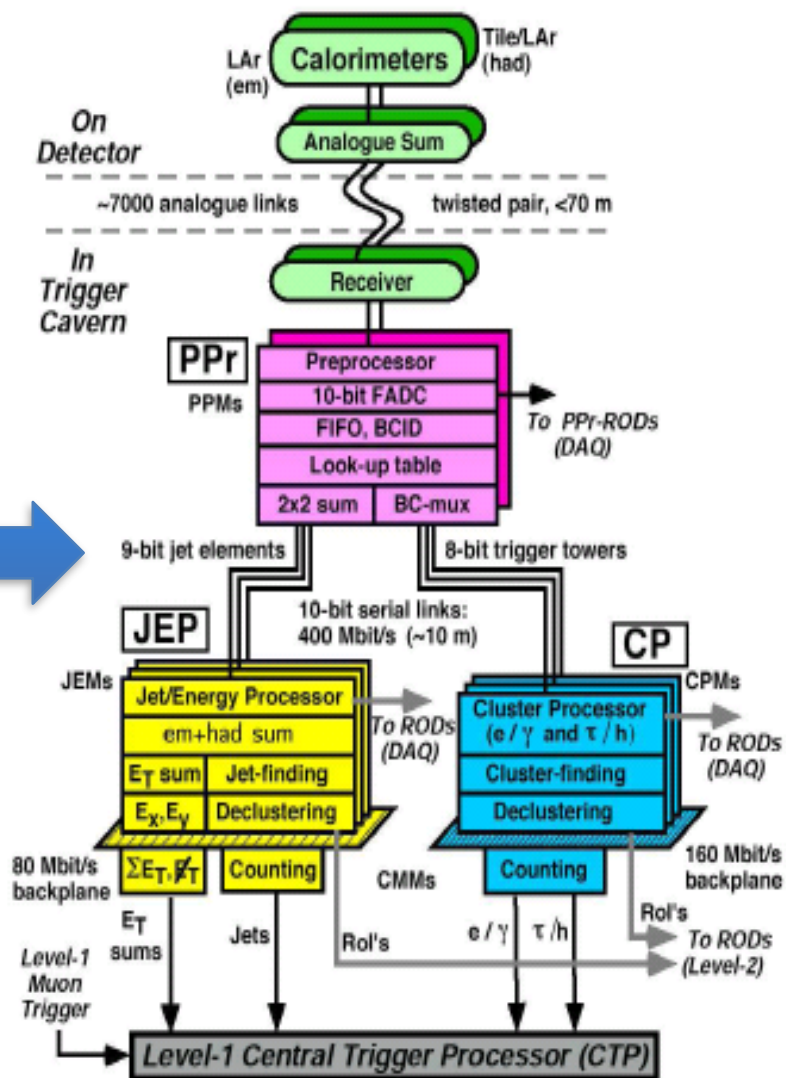
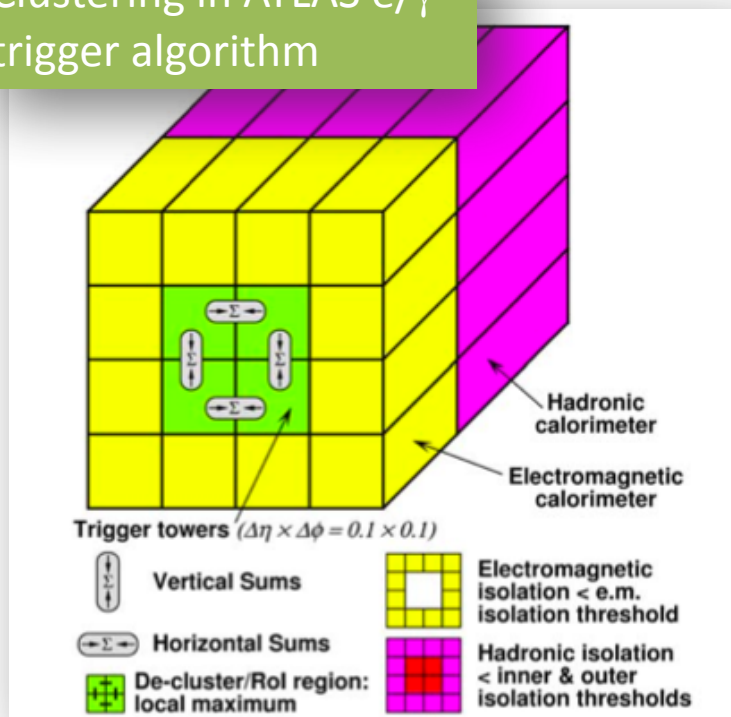
- On-detector:
 - Sum of analog signals from cells to **form towers**
- L1 trigger system is off-detector
- Pre-processor board
 - ADCs with 10-bit resolution
 - ASICs to perform the trigger algorithm
 - Assign energy (ET) via Look-Up tables
 - Apply threshold on ET
 - Peak-finder algorithm to assign the BC



Example: ATLAS calorimeter trigger

- Cluster Processor (CP)
- Jet/Energy Processor (JEP)
- Implemented in FPGAs, the parameters of the algorithms can be easily changed
- Total of 5000 digital links connect PPr to JEP and CP, 400 Mb/s

Clustering in ATLAS e/ γ trigger algorithm



High level triggers



High Level Trigger Architecture

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

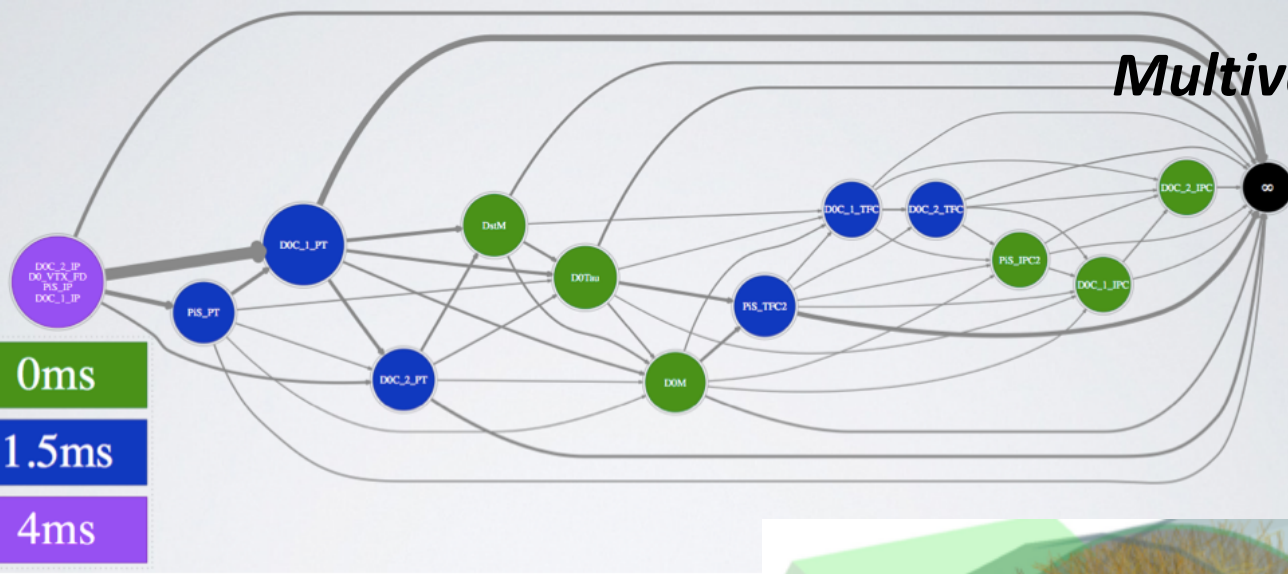
	Levels	L1 rate (Hz)	Event size	Readout bandw.	Data filter out
LEP	2/3	1 kHz	100 kB	few 100 kB/s	~5 Hz
ATLAS	2/3	100 kHz (L2: 10 kHz)	1.5 MB	30 GB/s (incremental Event Building)	~1 kHz
CMS	2	100 kHz	1.5 MB	100 GB/s	~1 kHz

- **LEP**: 40 Mbyte/s VME bus was able to support the bandwidth
- **LHC**: **latest technologies** in processing power, high-speed network interfaces, optical data transmission
- High data rates are held with different approaches
 - Network-based event building (LHC example: CMS)
 - Seeded reconstruction of data (LHC example: ATLAS)

Can we use the offline algorithms online?

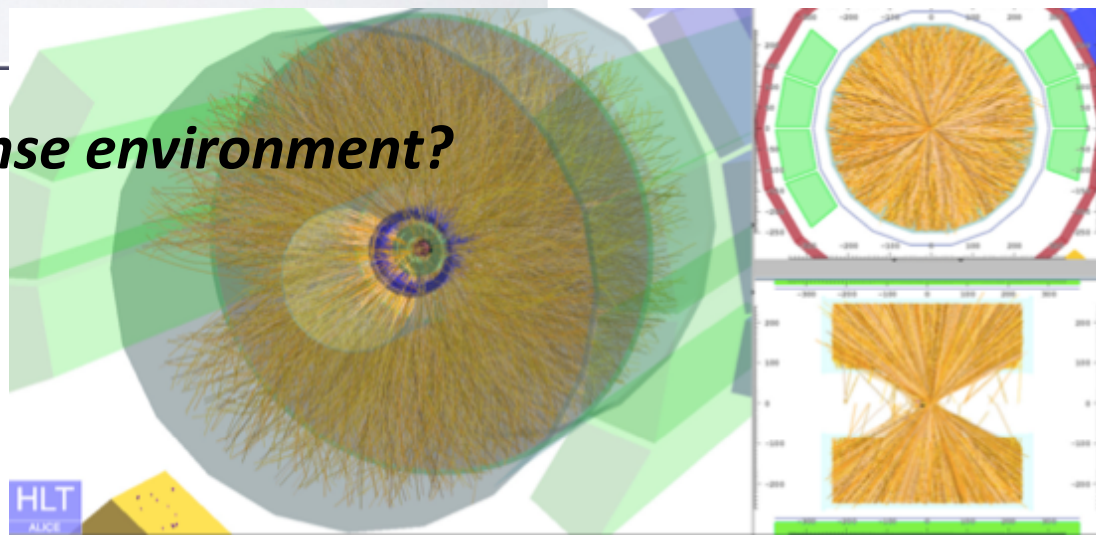
MDDAG, Benbouzid, Kegl et al.

Multivariate analysis?



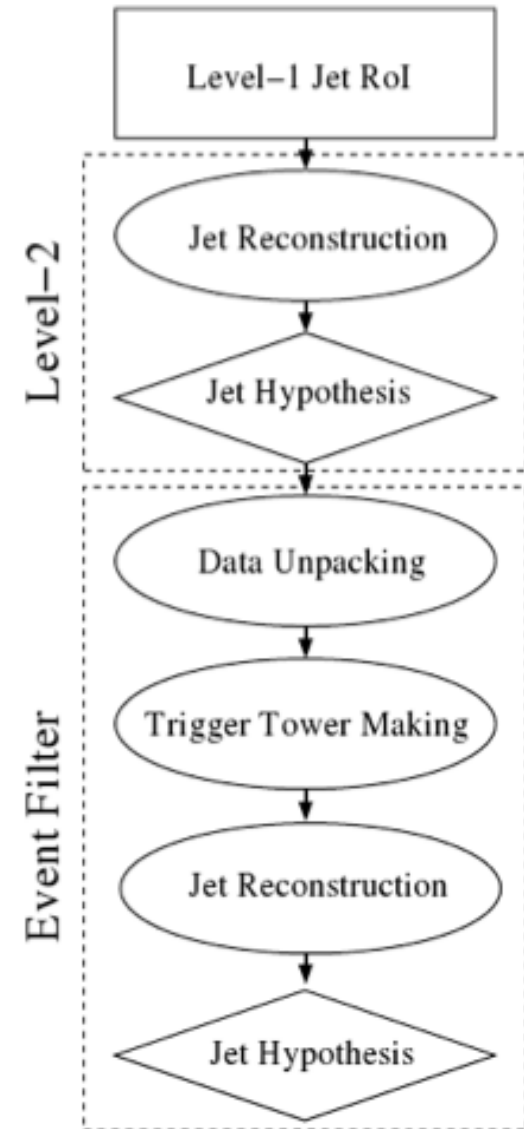
Pattern recognition in dense environment?

Latency is the constraint!



HLT design principles: early rejection

- **Early rejection** is crucial to
 - Reduce the data flux to the Readout buffers
 - Reduce resources (CPU usage, memory consumption....)
- Alternate steps of **feature extraction with hypothesis testing** allows to apply different hypothesis on the same feature
 - Can be optimized in different ways
- A complex **scheduling** can optimise the processing
 - First call algorithms which are fast and with higher rejection
 - Avoid running same algorithm on same data twice
 - Cache algorithm results (memo-ization)
 - Cache input data request (deep memo-ization)
- Decision taken on partial or full Readout/reconstruction
 - Analysing data in few interesting regions (Region-of-interest)
 - The full **event building** is integrated in the decision process



HLT design principles

➤ Early rejection: alternate steps of feature extraction with hypothesis testing

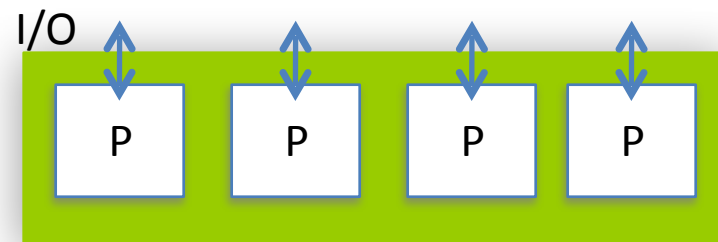
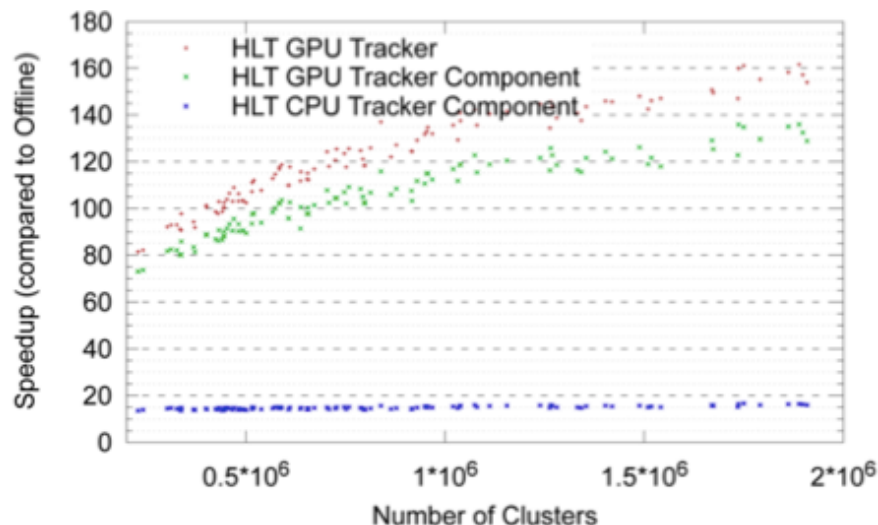
- Reduce data and resources (CPU, memory....)

➤ Event-level parallelism

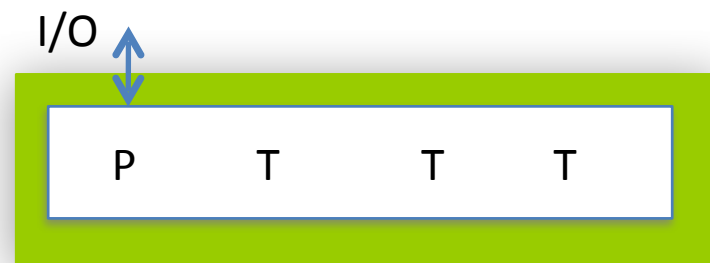
- Process more events in parallel, with multiple processors
- Multi-processing or/and multi-threading

➤ Algorithm-level parallelism

- Need to change paradigms for software developments
- **GPUs** can help in cases where large amount of data can be processed concurrently



Multi-processing



Multi-threading

Algorithms are developed and optimized offline

Try to have common software with offline reconstruction, for easy maintenance and higher efficiency

Now you can build your own trigger system!

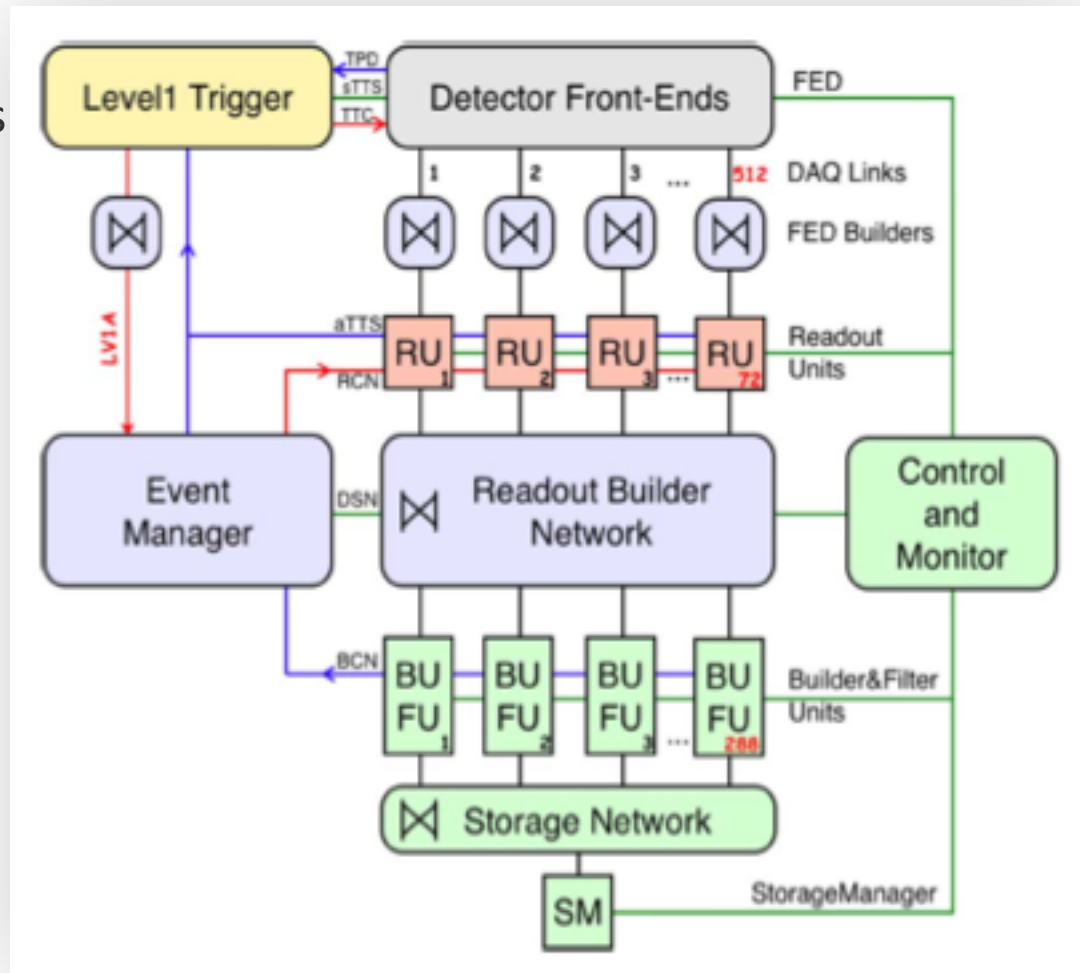
- Trigger and DAQ systems exploit all new technologies, being well in contact with industry
- Microelectronics, networking, computing expertise are required to build an efficient trigger system
- But being always in close contact with the physics measurements we want to study
- Here I just mentioned general problems, that will be deeply described during other lessons
- Profit of this school to understand these bonds!!

Back-up slides



Network-based HLT: CMS

- Data from the readout system (RU) are transferred to the filters (FU) through a builder network
- Each filter unit processes only a fraction of the events
- Event-building is factorized into a **number of slices**, each processing only $1/n^{\text{th}}$ of the events
 - Large total bandwidth still required
 - No big central network switch
 - Scalable



FU = several CPU cores = several filtering processes executed in parallel

Seeded reconstruction HLT: ATLAS

- Level-2 uses the information seeded by level-1 trigger
 - Only the data coming from the region indicated by the level-1 is processed, called **Region-of-Interest (RoI)**
 - The resulting total amount of RoI data is minimal: a few % of the Level-1 throughput
 - Level-2 can use the full granularity information of only a part of the detector
- **No need of large bandwidth**
- Complicate mechanism to serve the data selectively to the L2 processing

Typically, there are less than 2 Rols per event accepted by LVL1

