Royal Holloway University of London

CERN

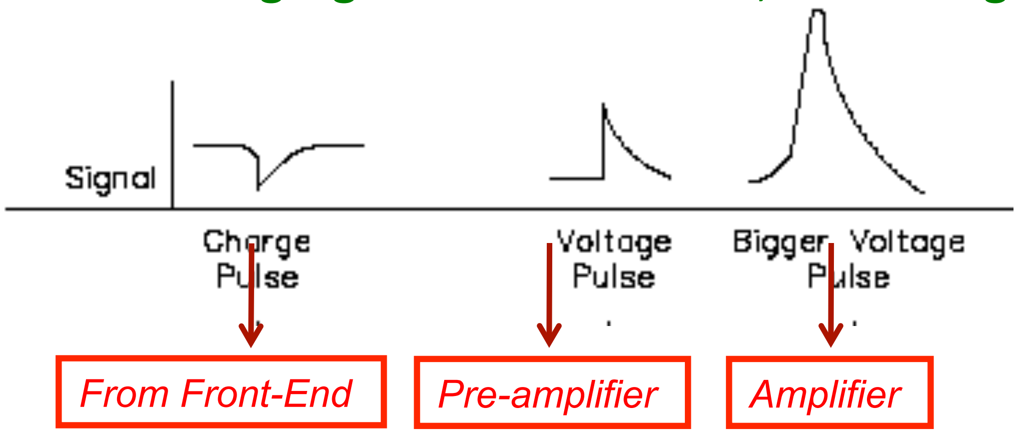
ISO-TDAQ school
Krakow, 01/02/2012

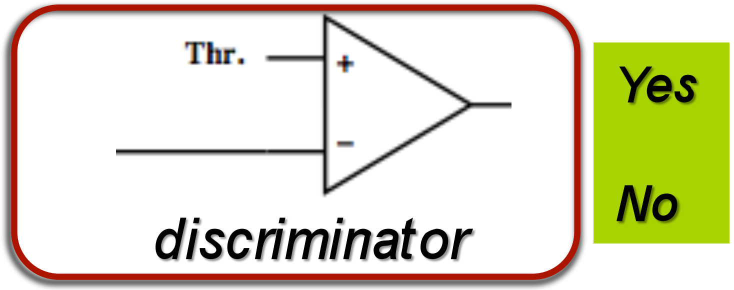
Trigger architectures

F.Pastore (RHUL)

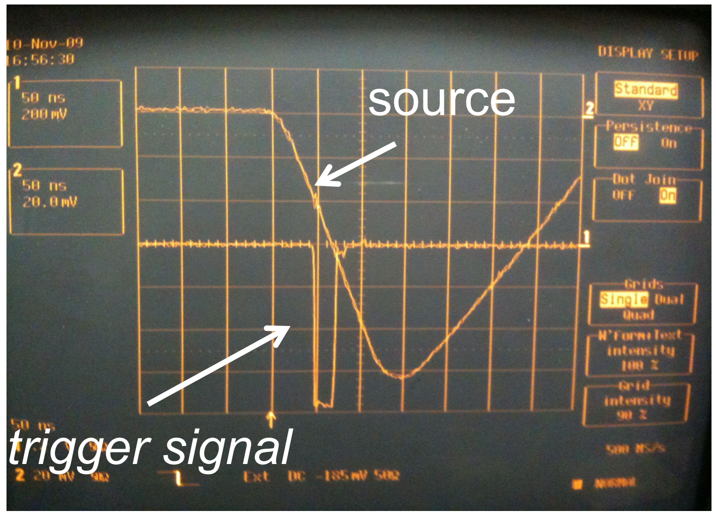
The simplest trigger system

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



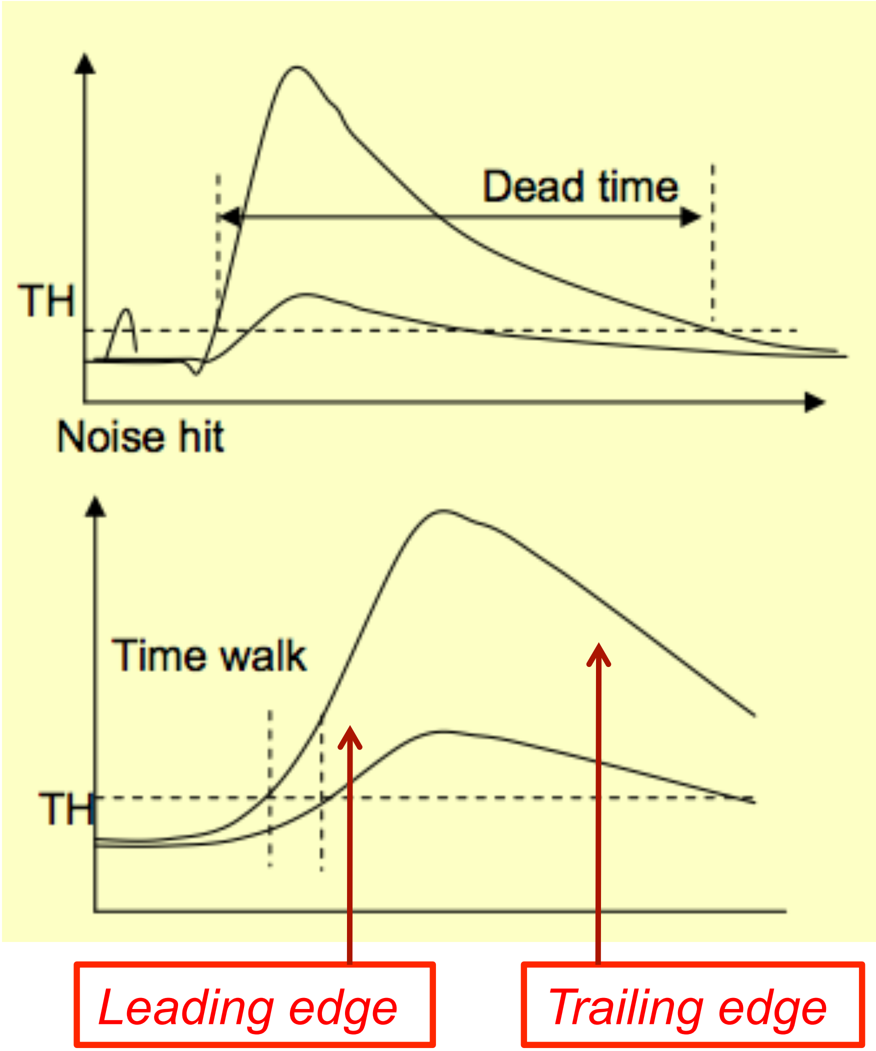


- The simplest trigger: apply a threshold
  - Look at the signal
  - Put a threshold as low as possible, since signals in HEP detectors have large amplitude variation
  - Compromise between hit efficiency and noise rate



Physics signals are different...

- Pulse width
  - **Dead-time** limits effective hit rate
  - We must adapt the width to the desired trigger rate
- Time walk
  - Threshold-crossing time depends on amplitude of signal
  - Must be minimized in a good trigger system



- Elimination of time-walk for signals with the same rise-time, but different amplitude, is possible with more complex discrimination…
- Scintillation detectors and PMT's have a constant trailing time at a particular fraction of the amplitude (usually 10-15%)

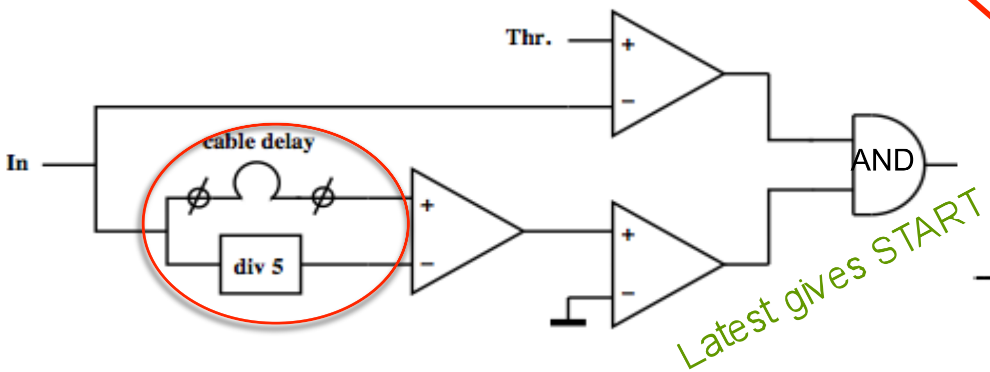
Constant fraction discriminators

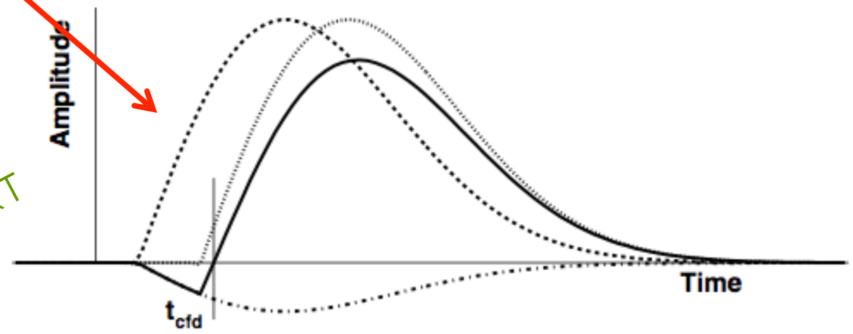
If two signals have the same rising time at a fraction **f**

$t(A_f) - t(A_0) = \text{constant}$

→ ***A(delay, t) = f • A(t)*** ***at t_{CFD}***

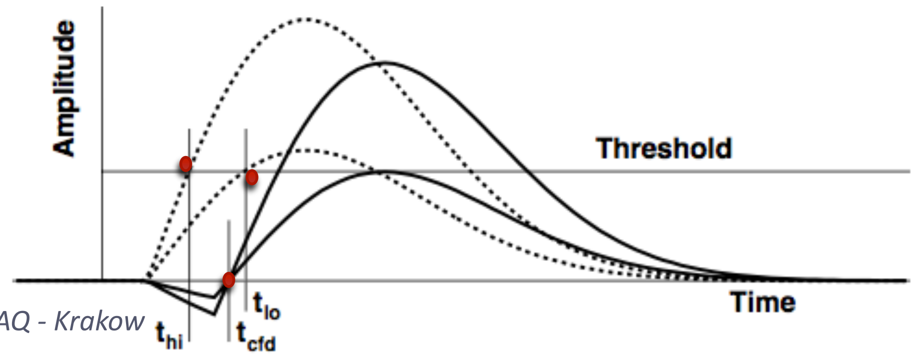
- - - *Input pulse*
· · · · *Delayed input pulse*
— · – *Attenuated inverted input*
—— *Bipolar pulse*





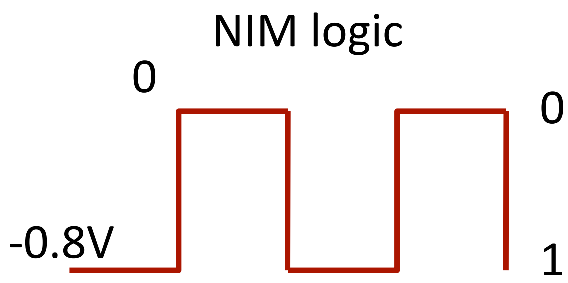
The output of the CFD fires when the bipolar pulse changes polarity

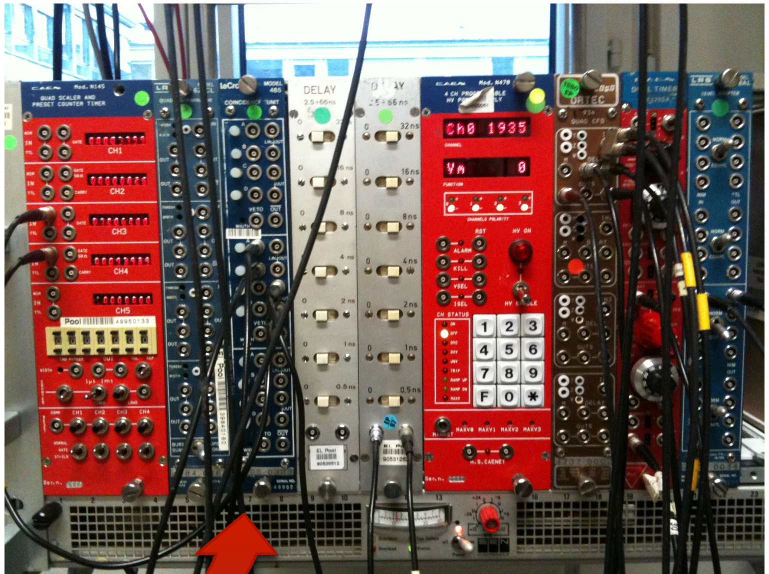
- Attenuation and delay (configurable) applied before the discrimination determine t_{CFD}
- If the delay is too short, the unit works as a normal discriminator for signals with a low amplitude because then the output of the normal discriminator fires later than the CFD part

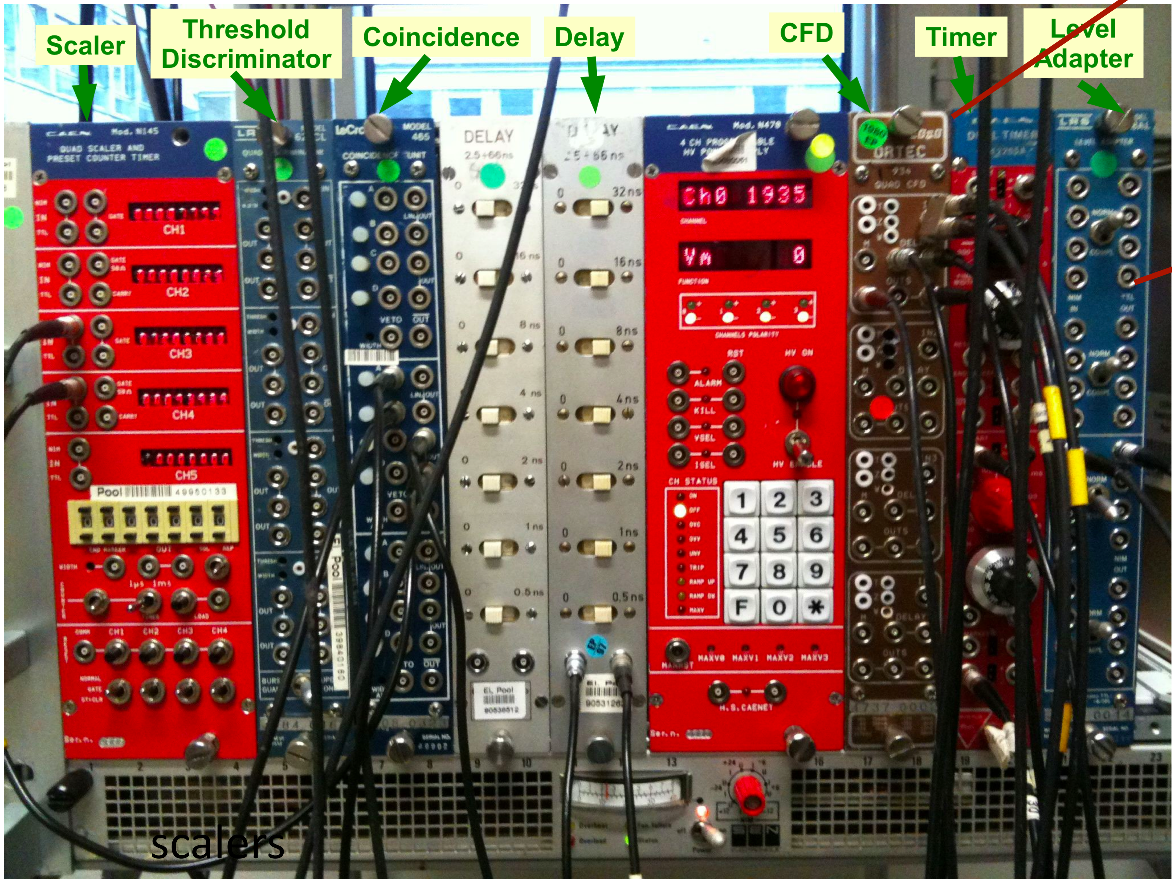


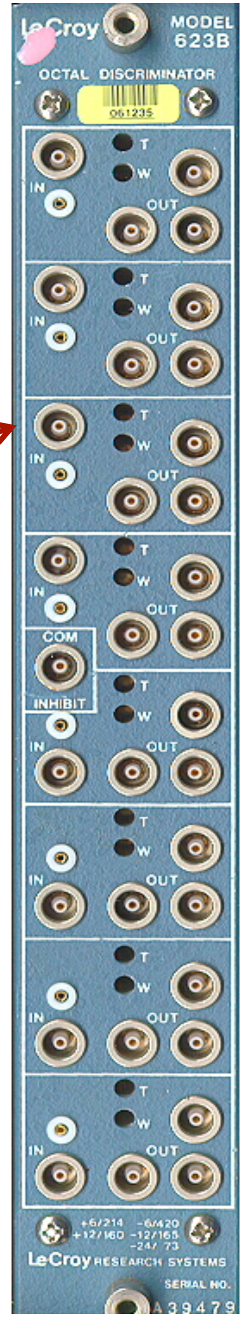
And now build your own trigger system

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









- Threshold levels configurable via screwdriver adjust

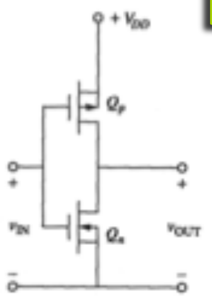
ORTEC CFD

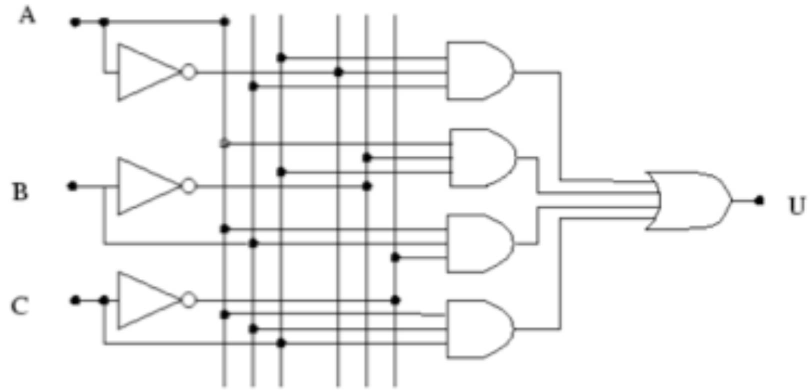
LeCroy discriminator

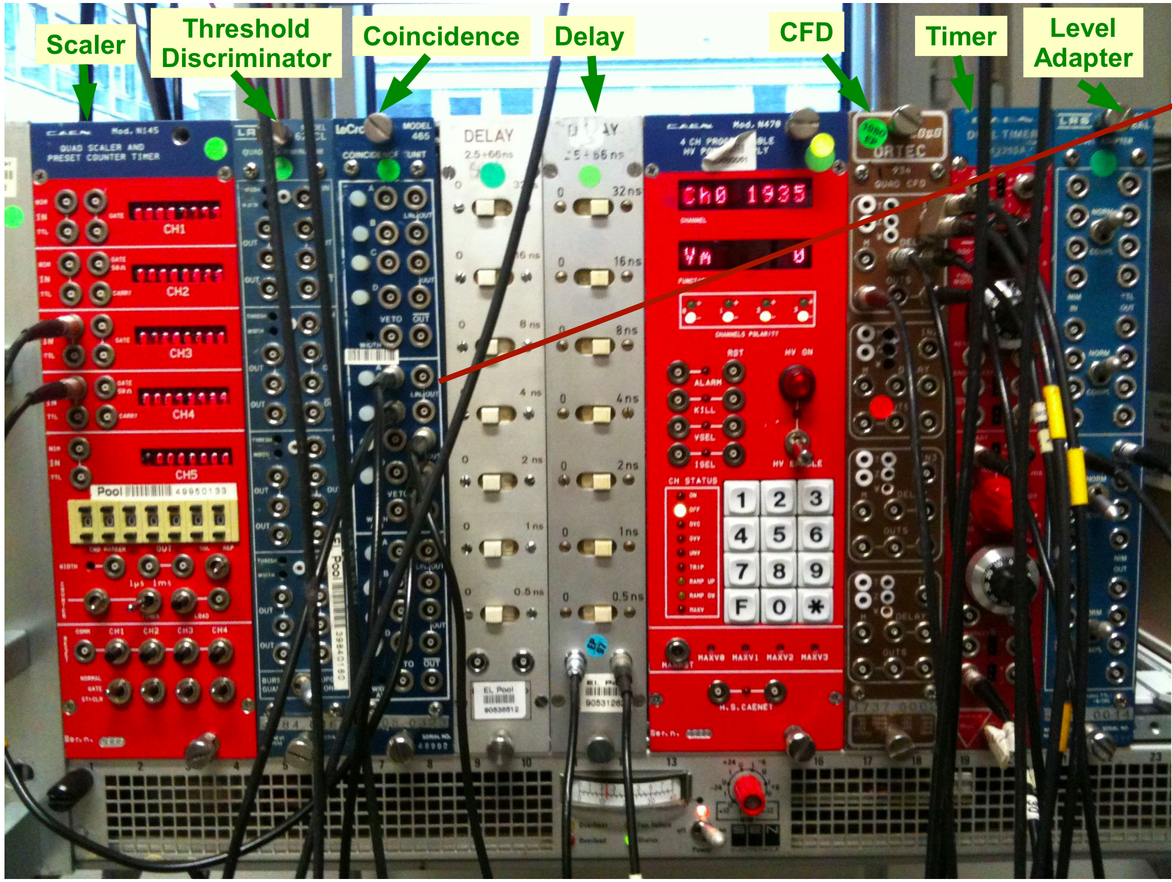
Trigger logic implementation

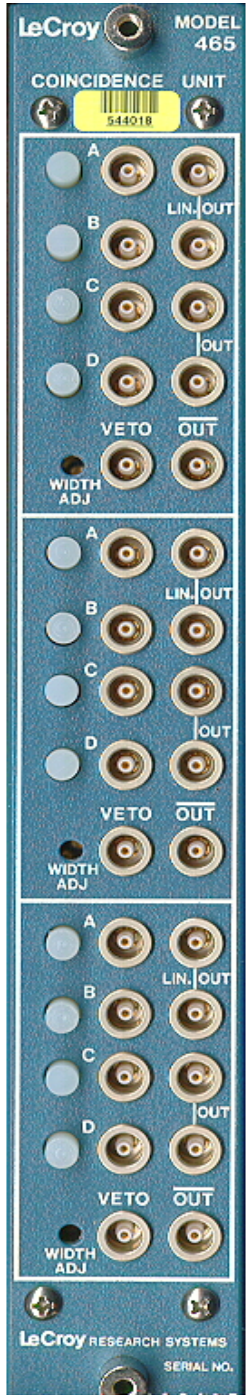
- Analog systems: amplifiers, filters, comparators, ….
- Digital systems:
  - Combinatorial: sum, decoders, multiplexers,….
  - Sequential: flip-flop, registers, counters,….
- Converters: ADC, TDC, …..

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







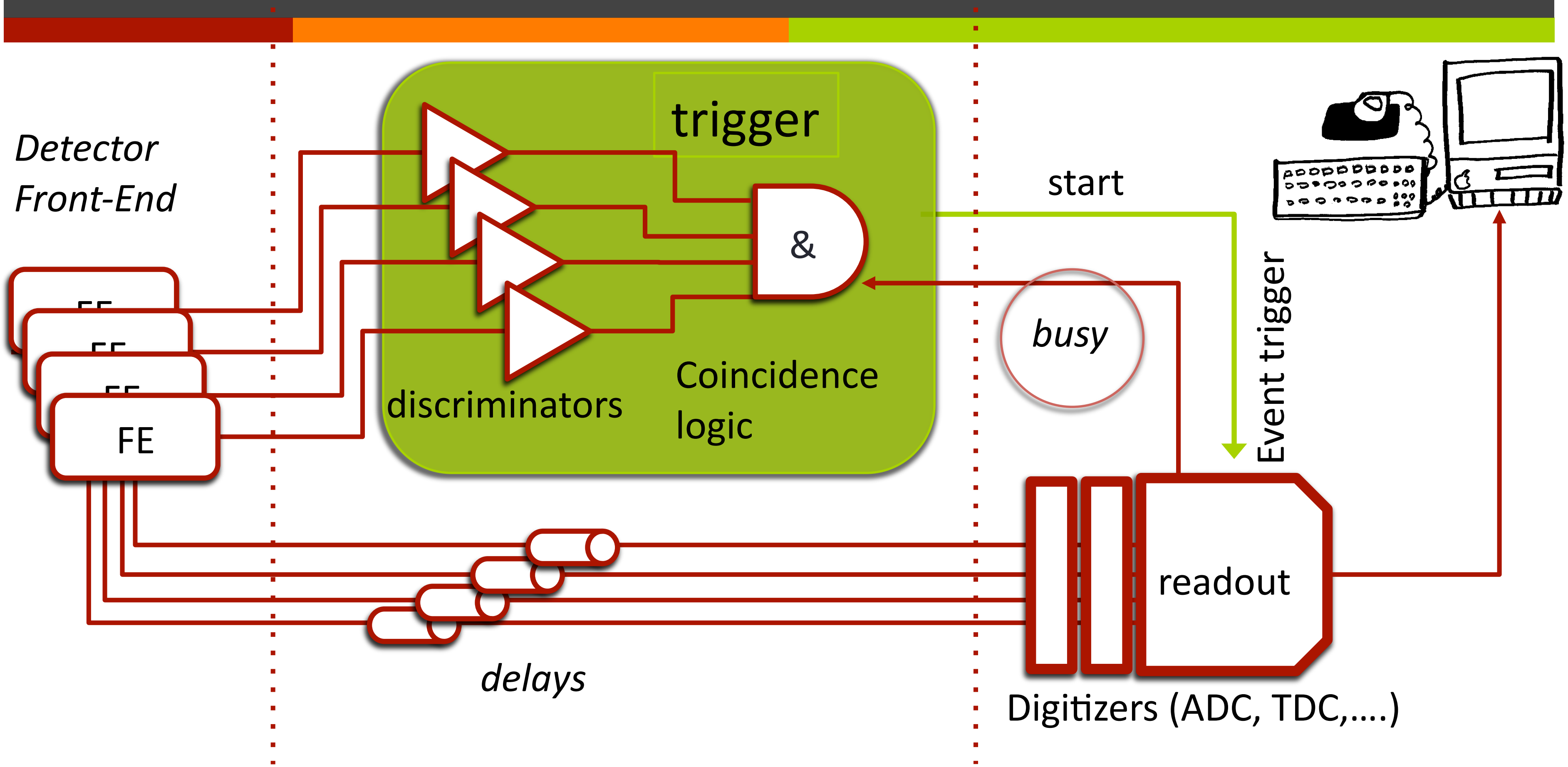


LeCtroy Concidence Unit

Summary of the trigger requirements

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

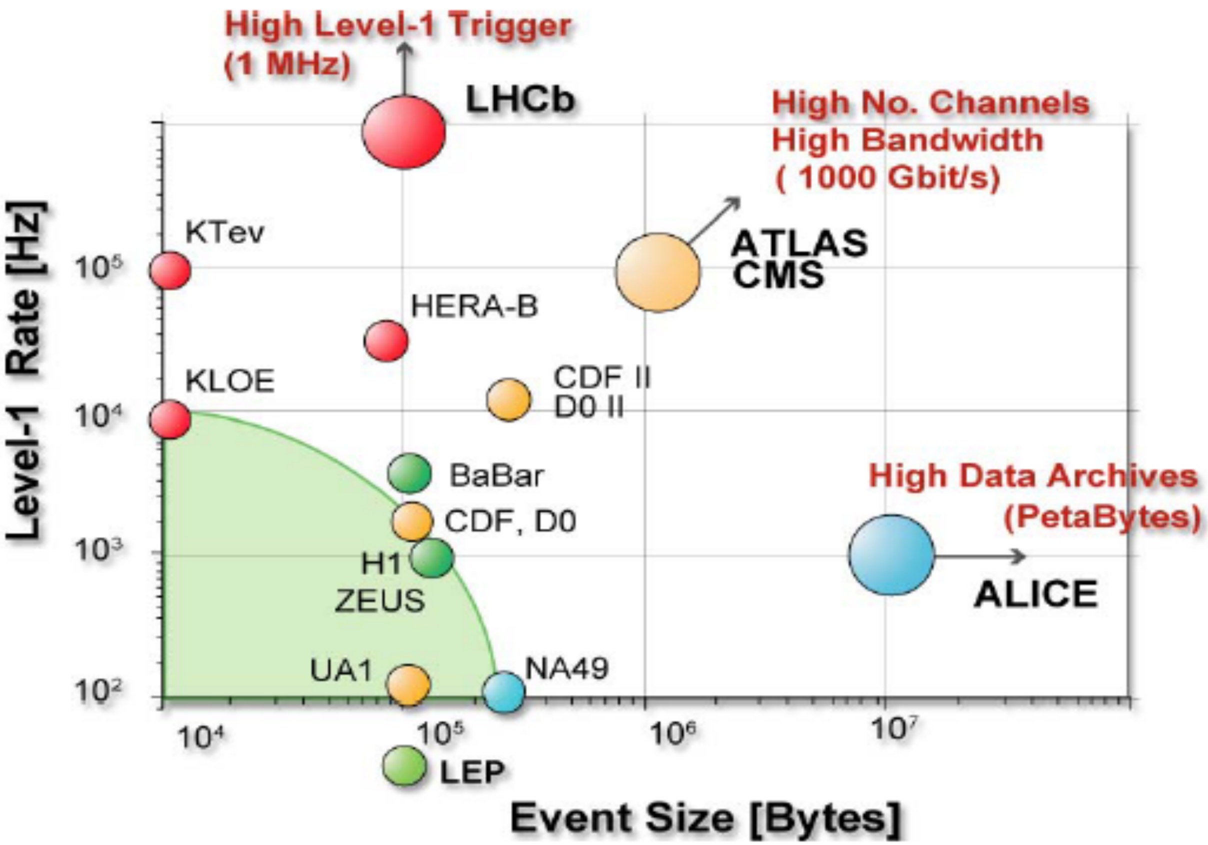
A simple trigger system



- Due to fluctuations, incoming rate is higher than processing one
- Valid interactions are rejected due to system **busy**

Trigger and data acquisition trends

- As the data volumes and rates increase, new architectures need to be developed
  - Allowed data bandwidth = Rate x Event size

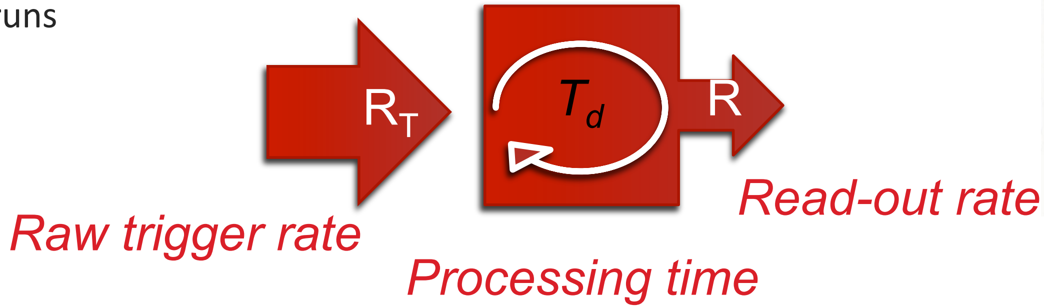


Dead-time

In our example of the photo-camera, if we want to take photos close in time, the limit on the maximum rate is the processing time of the camera

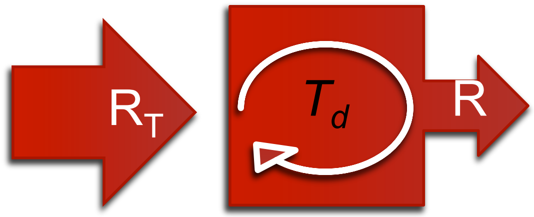
- The most important parameter controlling the design and performance of high speed **DAQ systems**
  - Occurs whenever a given step in the processing takes a **finite amount of time**
  - It's the fraction of the acquisition time in which no events can be recorded, typically of the order of **few %**

- Mainly three sources:
  - Readout dead-time:
    - before the complete event has been readout, no other events can be processed (during this time the DAQ asserts a BUSY)
  - Trigger dead-time:
    - trigger logic processing time, summed over all the components
  - Operational dead-time:
    - data-taking runs



Maximize event recording rate

R_T = raw trigger rate
R = number of events read per second (DAQ rate)
T_d = readout time interval per event



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

Fraction of lost events!

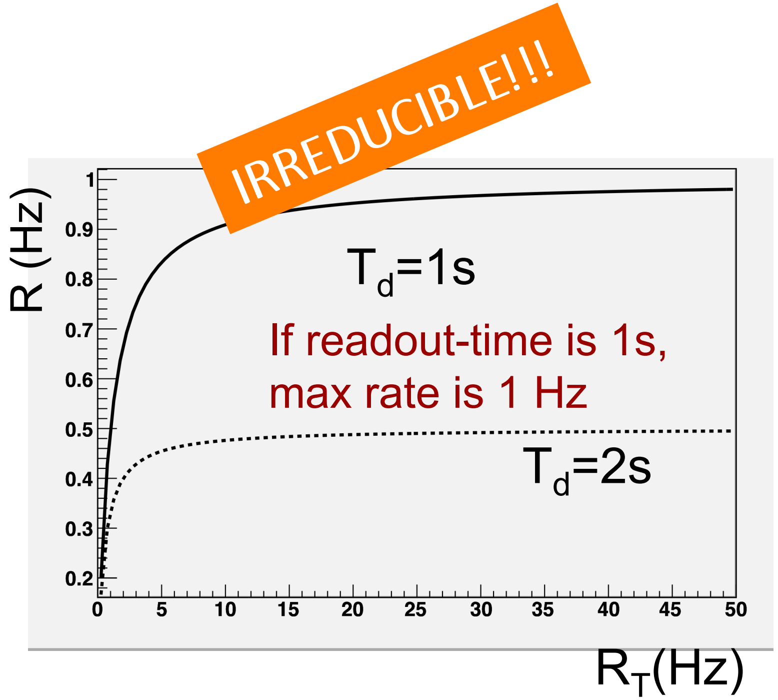
The fraction of surviving events (lifetime ratio) is:

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

T_d limits the maximum DAQ rate ($R=1/T_d$) regardless of the input trigger rate :

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

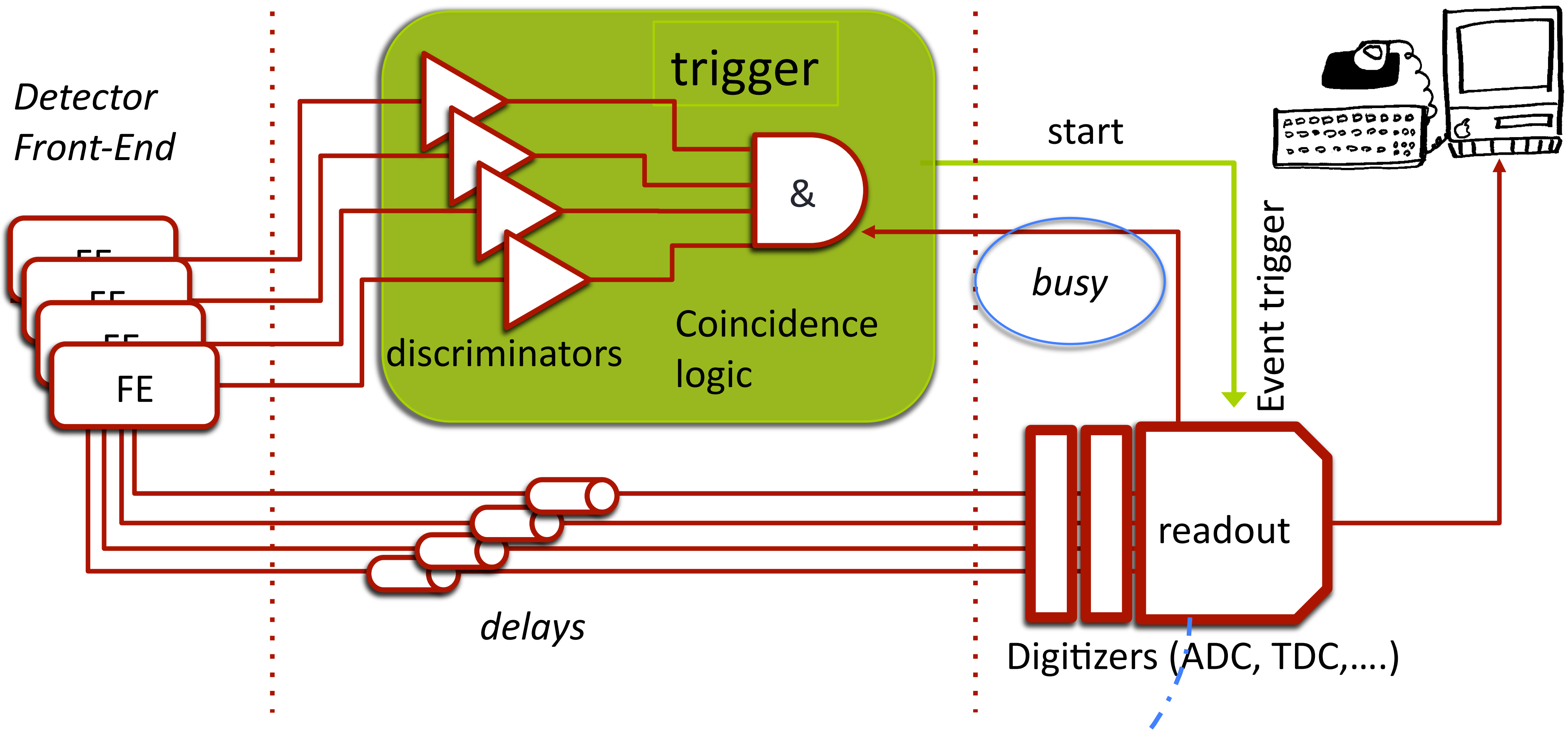
- Due to fluctuations, the incoming rate is higher than the processing one



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

FAST TRIGGER!
LOW DATA RATE!

A simple trigger system



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

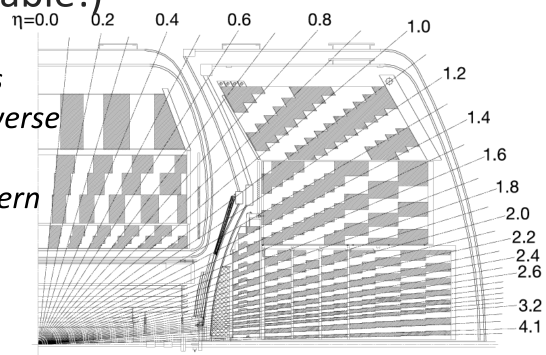
Fraction of lost events due to readout

Features to minimize dead-time

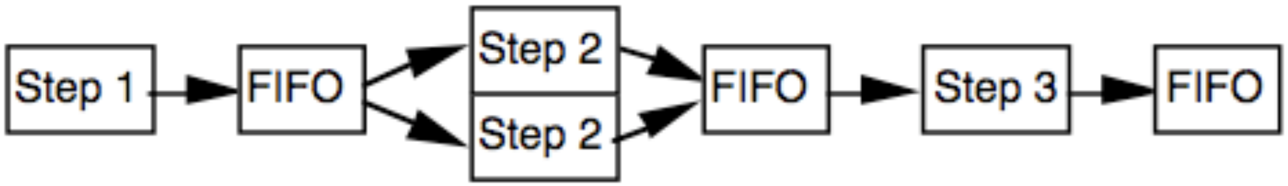
- Two approaches are applied for large dataflow systems
  - **Parallelism**
    - Independent readout and trigger processing paths, one for each detector element
    - Digitization and DAQ processed in parallel (as many as affordable!)

Segment as much as you can!

DZero calorimeters showing the transverse and longitudinal segmentation pattern



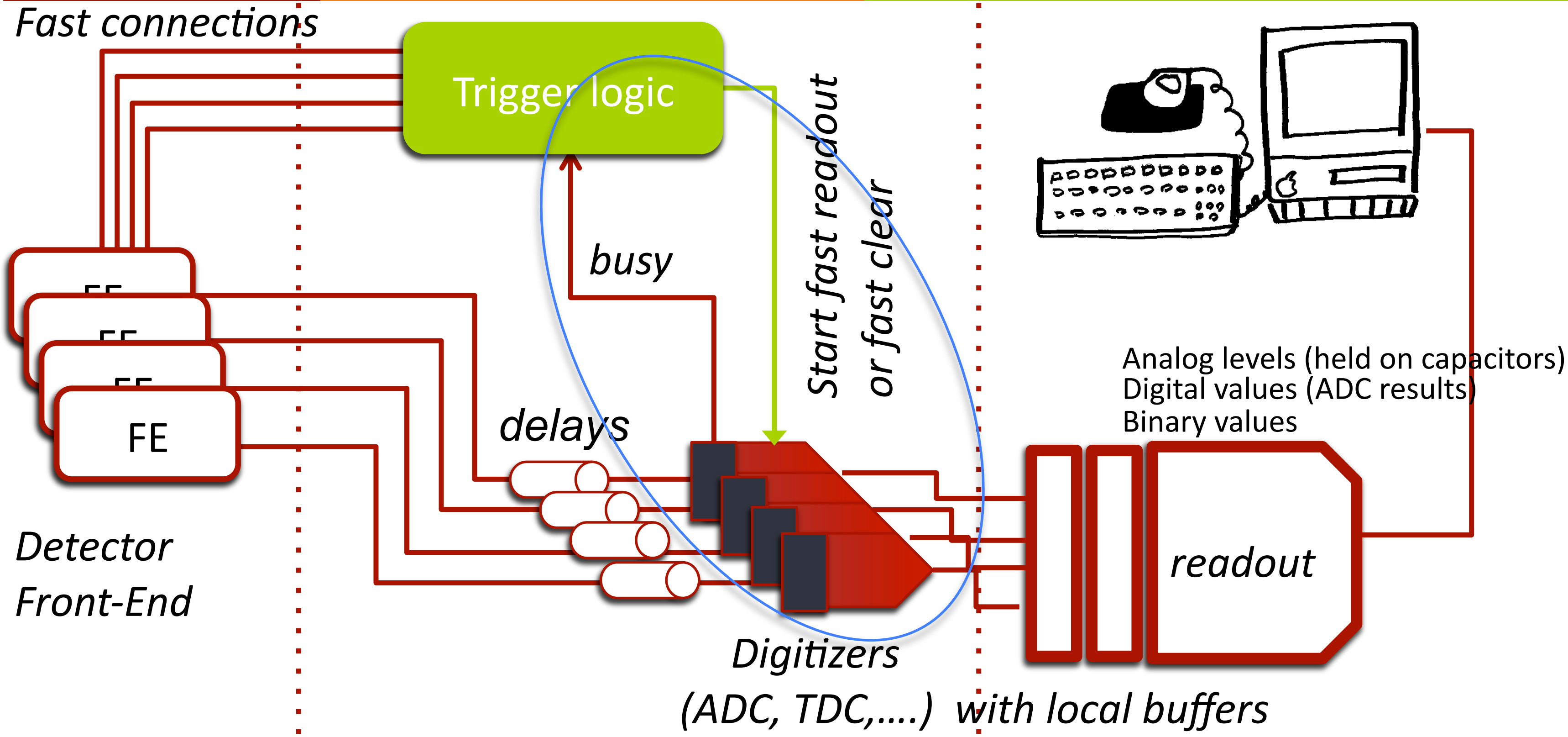
  - **Pipeline processing** to absorb fluctuations
    - Organize the process in different steps
    - Use of local **buffers** (FIFOs) between steps allows steps with different timing (big events processed during short events).
    - The depth of local buffers limits the processing time of the subsequent step



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

Try to absorb in capable buffers

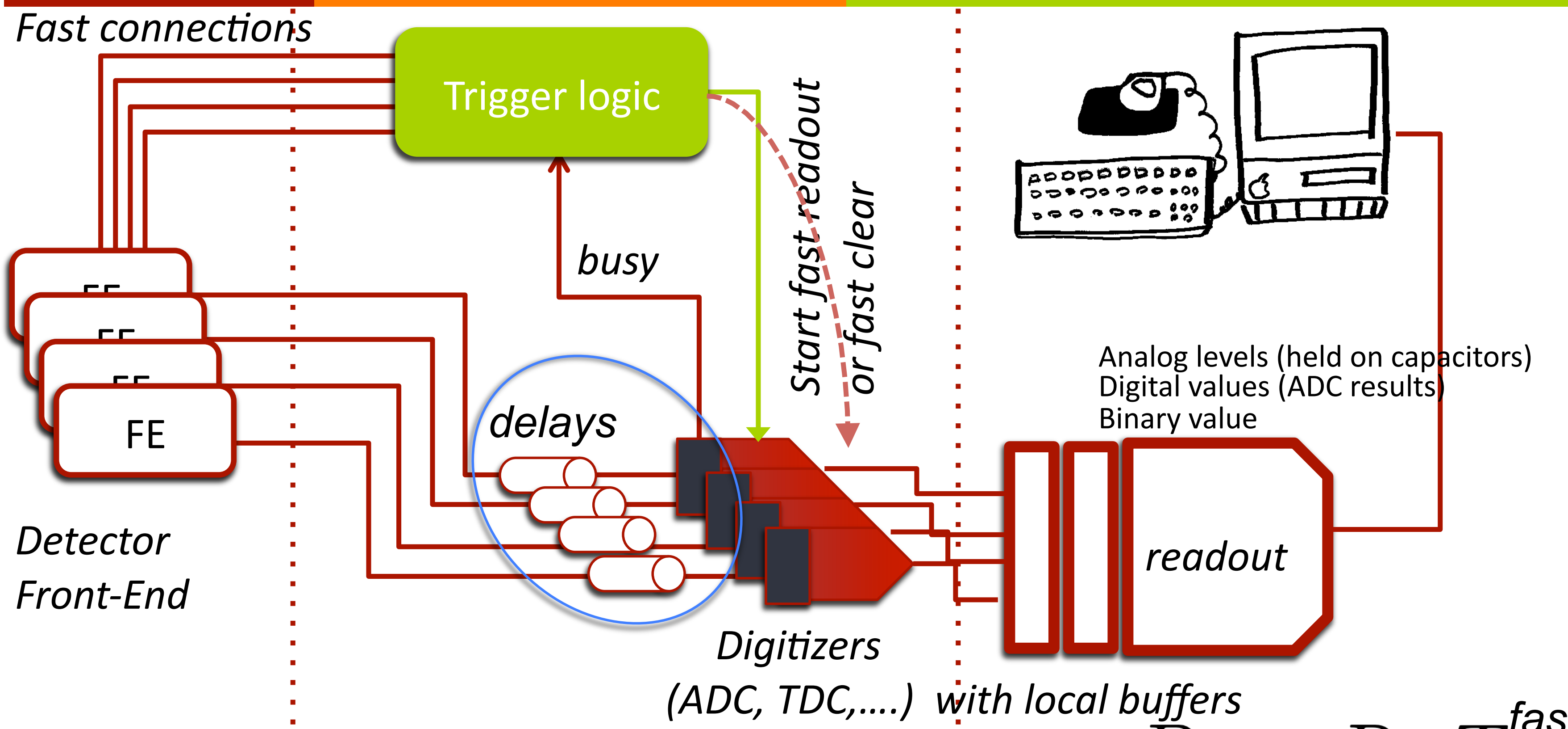
Minimizing readout dead-time...



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

- **Pipeline**: Different stages of readout: fast local readout plus global event readout (slow)
  - Dead-time is the product of the trigger rate and the **fast readout time**
- **Parallelism**: Use multiple digitizers: trigger sends a fast readout or a fast clear command to all local buffers

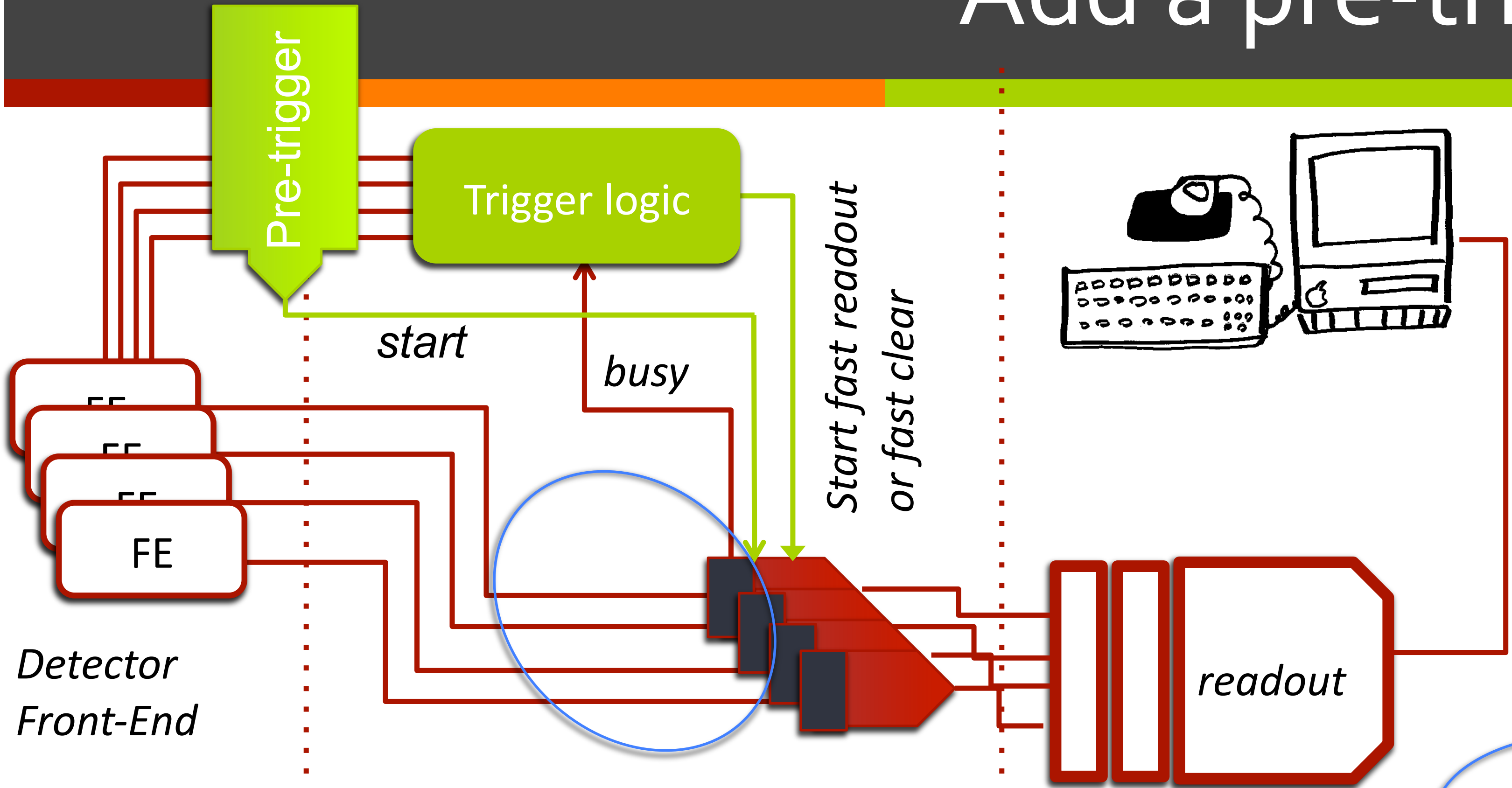
Trigger latency



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

- Trigger latency = time to form the trigger decision and distribute it to the digitizers
- Signals have to be delayed until the trigger decision is available at the digitizers
  - But more complex is the selection, longer the latency

Add a pre-trigger



$$D_t = R_T \cdot T_{RO} + R_{pT} \cdot L_{pT}$$

- Add **a very fast** first stage of the trigger, signaling the presence of minimal activity in the detector
  - Sends **START** to the digitizers (gate for ADCs, start of TDCs..), confirmed later by the main trigger (start fast readout) or not (fast clear)
  - Must be available when the signals from the detectors arrive at the digitizers
- The main trigger can come later (after the digitization) -> 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 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

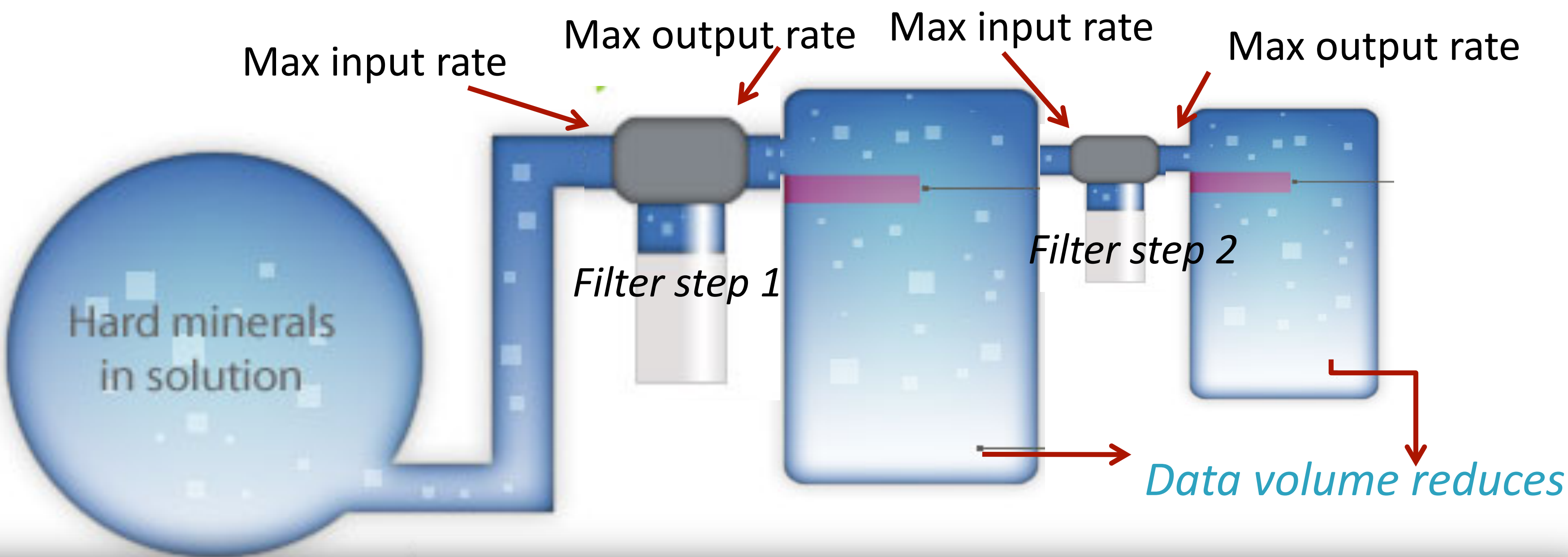
T_{LRO} = Local readout time

- Readout dead-time is minimum if its input rate R_N is low
- Aim is to minimize each product!

Buffering and filtering

- **At each step**, data volume is reduced, more refined filtering to the next step
- At each step, data are held in buffers
  - 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

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



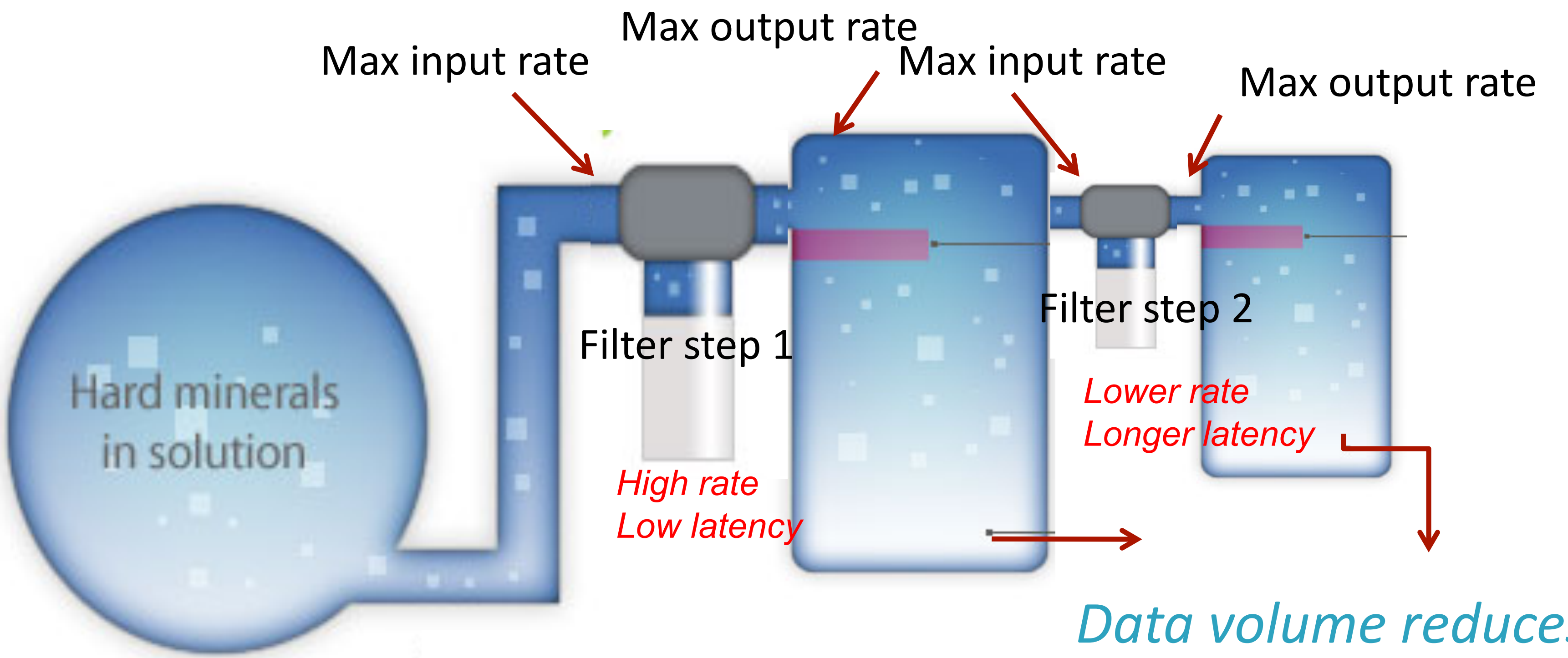
As long as the buffers do not fill up (overflow), no additional dead-time is introduced!

BUSY signal is still needed

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)

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



Data volume reduces

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 power, with longer latency (more complex algorithms)

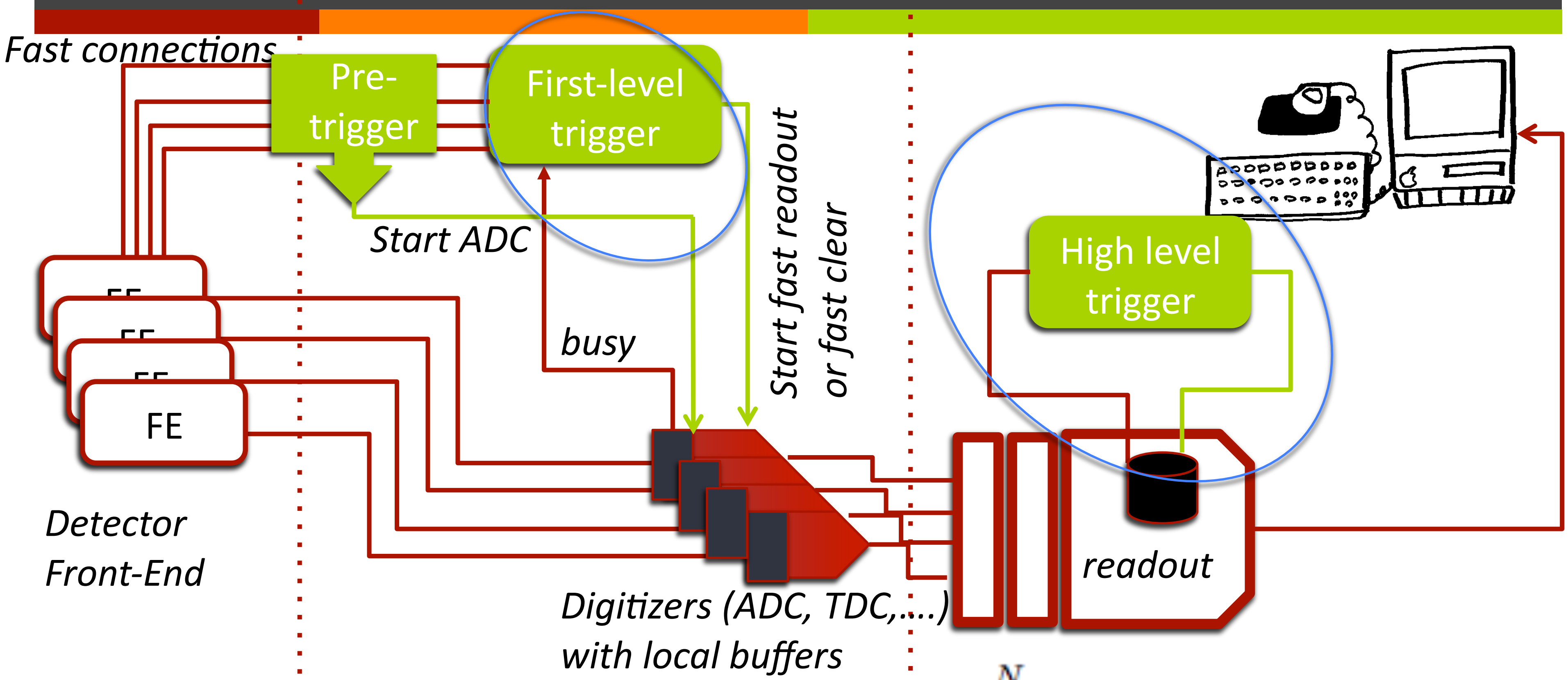
Lower event rate
Bigger event fragment size
More granularity information
More complexity
Longer latency
Bigger buffers

LHC experiments

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

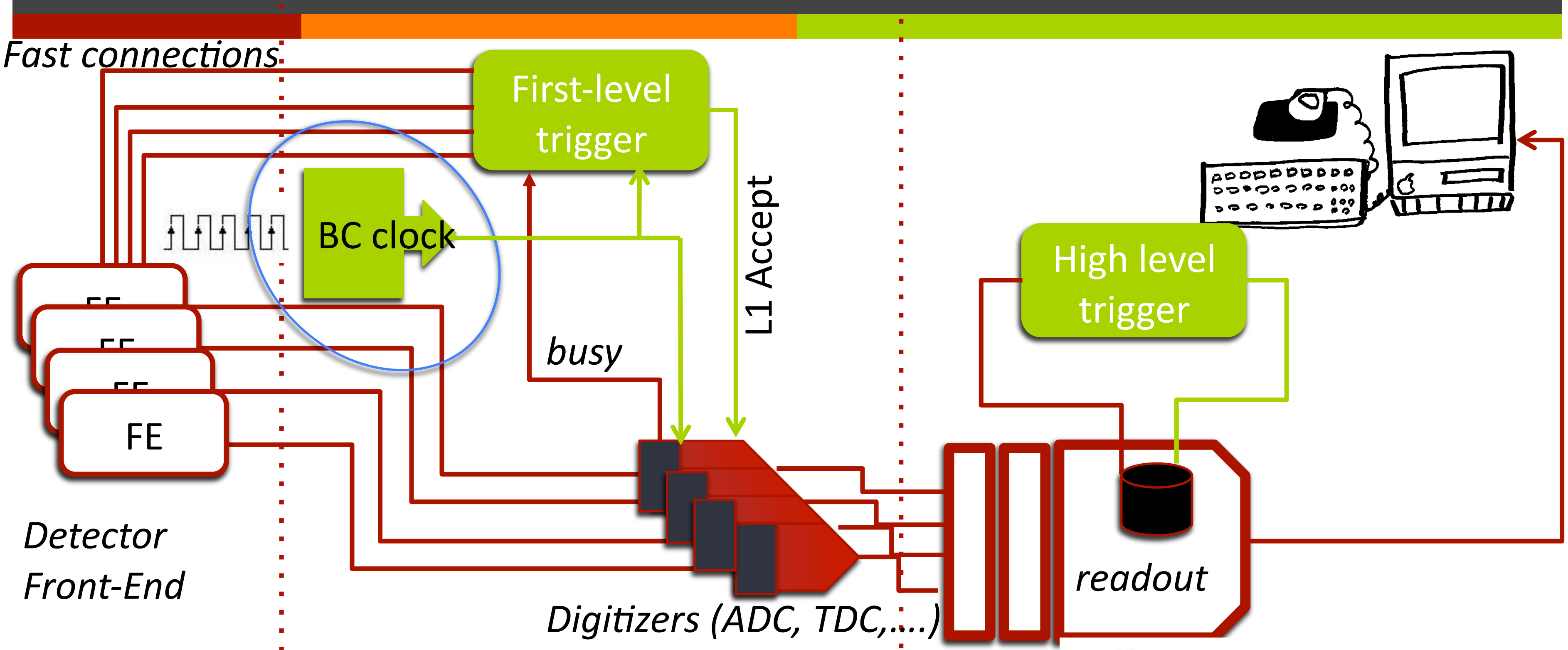
Schema of a multi-level trigger



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

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

Schema of a multi-level trigger @ colliders



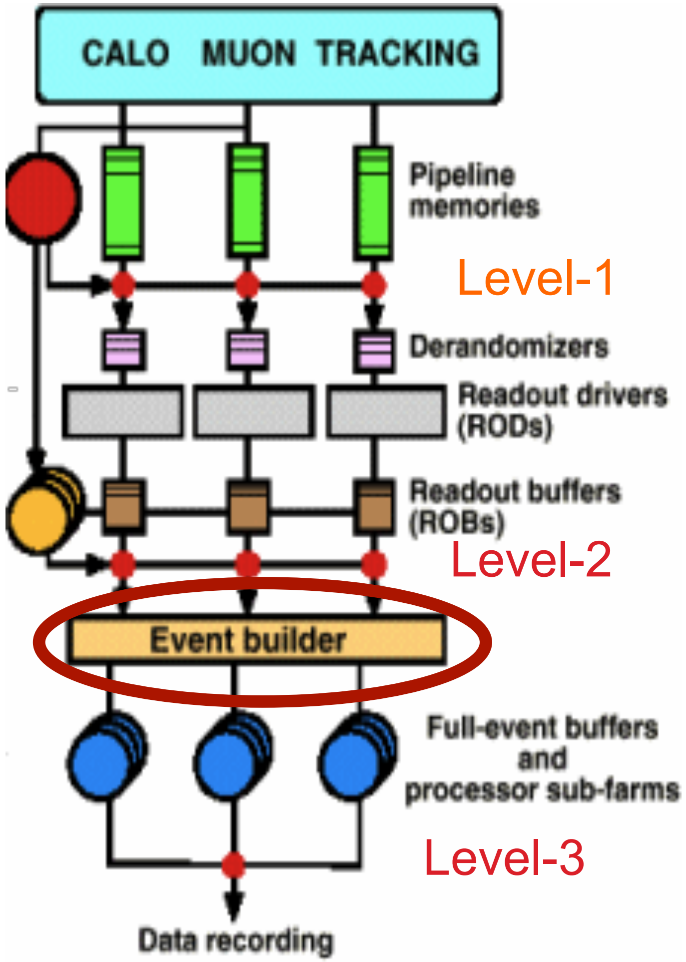
$$(\sum_{i=2}^{N} R_{i-1} \times L_i) + R_N \times T_{\mathrm{LRO}}$$

- In the collider experiments, 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
    - Fast electronics working at the BC frequency

Logical division between levels

- **First-level**: Rapid rejection of high-rate backgrounds
  - **Fast custom electronics** processing fragments of data from FE
  - **Coarse granularity** data from detectors
    - Calorimeters for electrons/γ/jets, muon chambers
    - Usually does not need to access data from the tracking detectors (only if the rate can allow it)
  - **Needs high efficiency, but rejection power can be comparatively modest**

- **High-level**: rejection with more complex algorithms
  - **Software** selection, running on computer farms
  - Progressive reduction in rate after each stage allows use of more and more complex algorithms at affordable cost
  - Can access only part of the event or **the full event** (see next slides)
    - Full-precision and **full-granularity** information
    - **Fast tracking** in the inner detectors (for example to distinguish e/γ)



Level-1: reduce the latency

- Pipelined trigger
- Fast processors
- Fast data movement

Level-1 trigger processing time

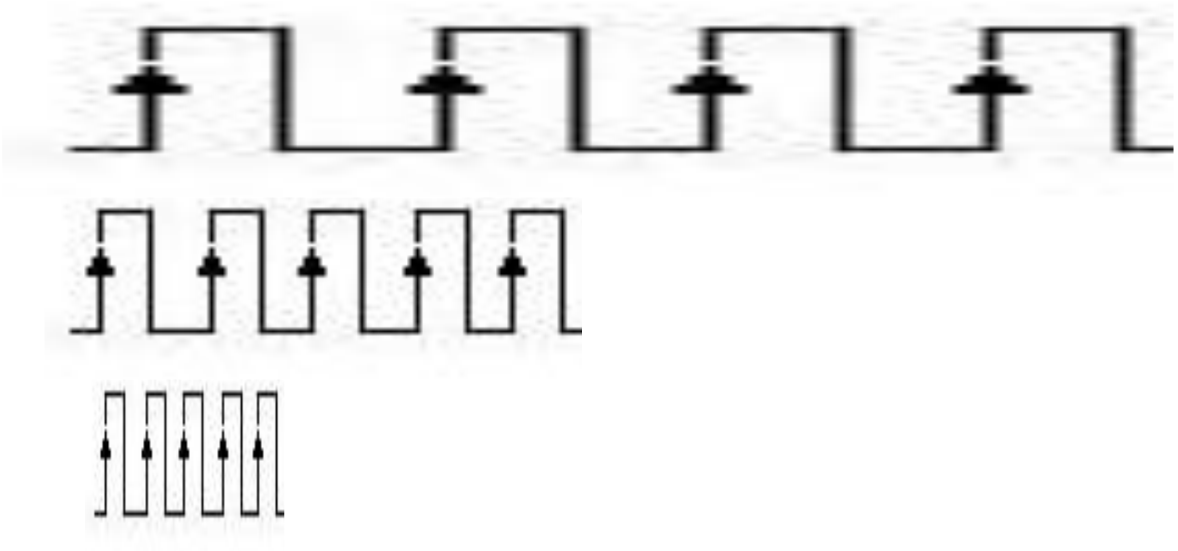
$$R = \mu \cdot f_{BC} = \sigma_{in} \cdot L$$

- @LEP, BC interval = 22 μs: complicated trigger processing within few μs latency was allowed
- In modern colliders: the required high luminosity is driven by high rate of bunch-crossing, then the BC period is short
  - It's not possible to make a trigger decision within this short time!

Tevatron: 396 ns

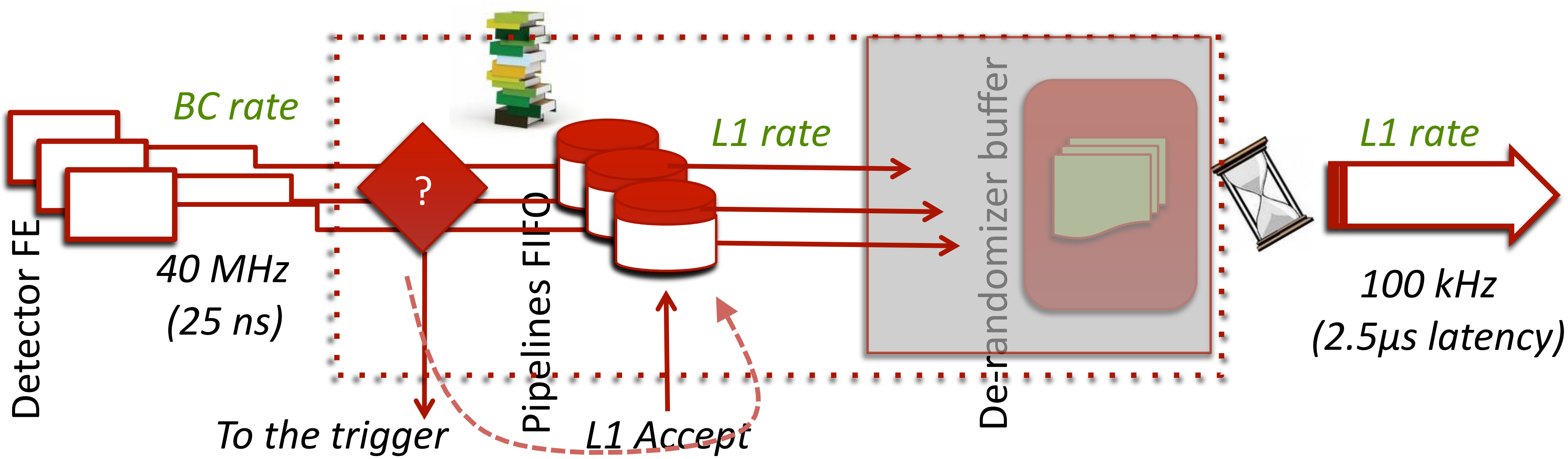
HERA: 96 ns

LHC: 25 ns



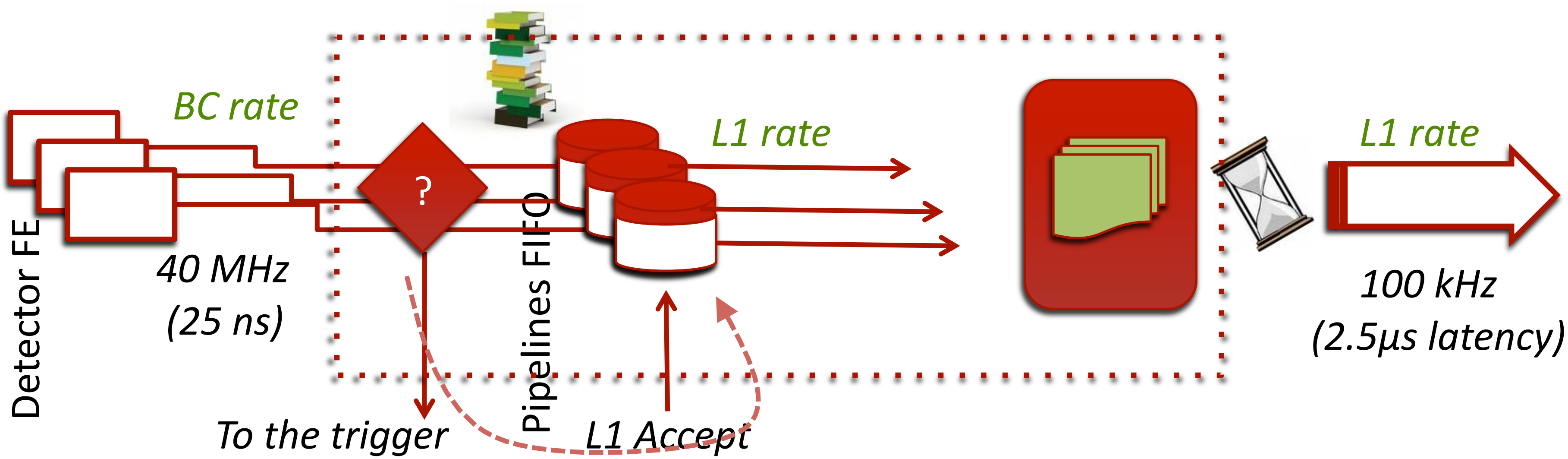
Level-1 trigger readout

- Pipeline readout at L1:
  - Data retained during the L1 latency in logical pipeline
  - Level-1 trigger result (L1Accept) starts the readout of the local FIFOs
- The level-1 buffers must be at least as deep as the expected latency, or the data associated with a particular L1 decision would be lost before the decision is made
  - BC 25 ns, L1 latency 2.5 μs -> minimum 100 events buffer (100 BCs)
- Fixed latency allows to find the data of the correct BC



Level-1 trigger readout

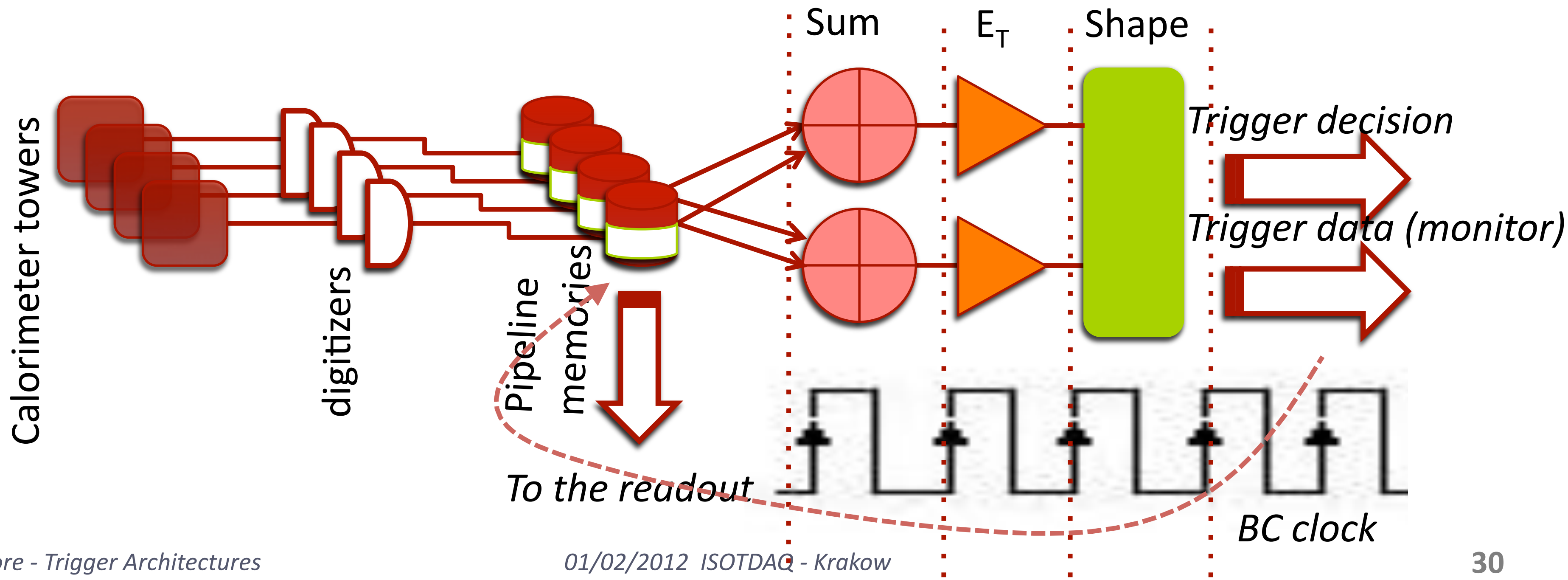
- From FIFOs, data are collected any L1-Accept into a de-randomizer who processes data into a preliminary (partial) event-building
  - Small **dead-time** is added in input (few BCs) to avoid overlap of data
  - **Dead-time** is added in output to avoid de-randomizer overflow (if two triggers are too close in time)
    - LHC: 5 BC dead-time x 100 kHz L1 rate x 25 ns = 1.25 %



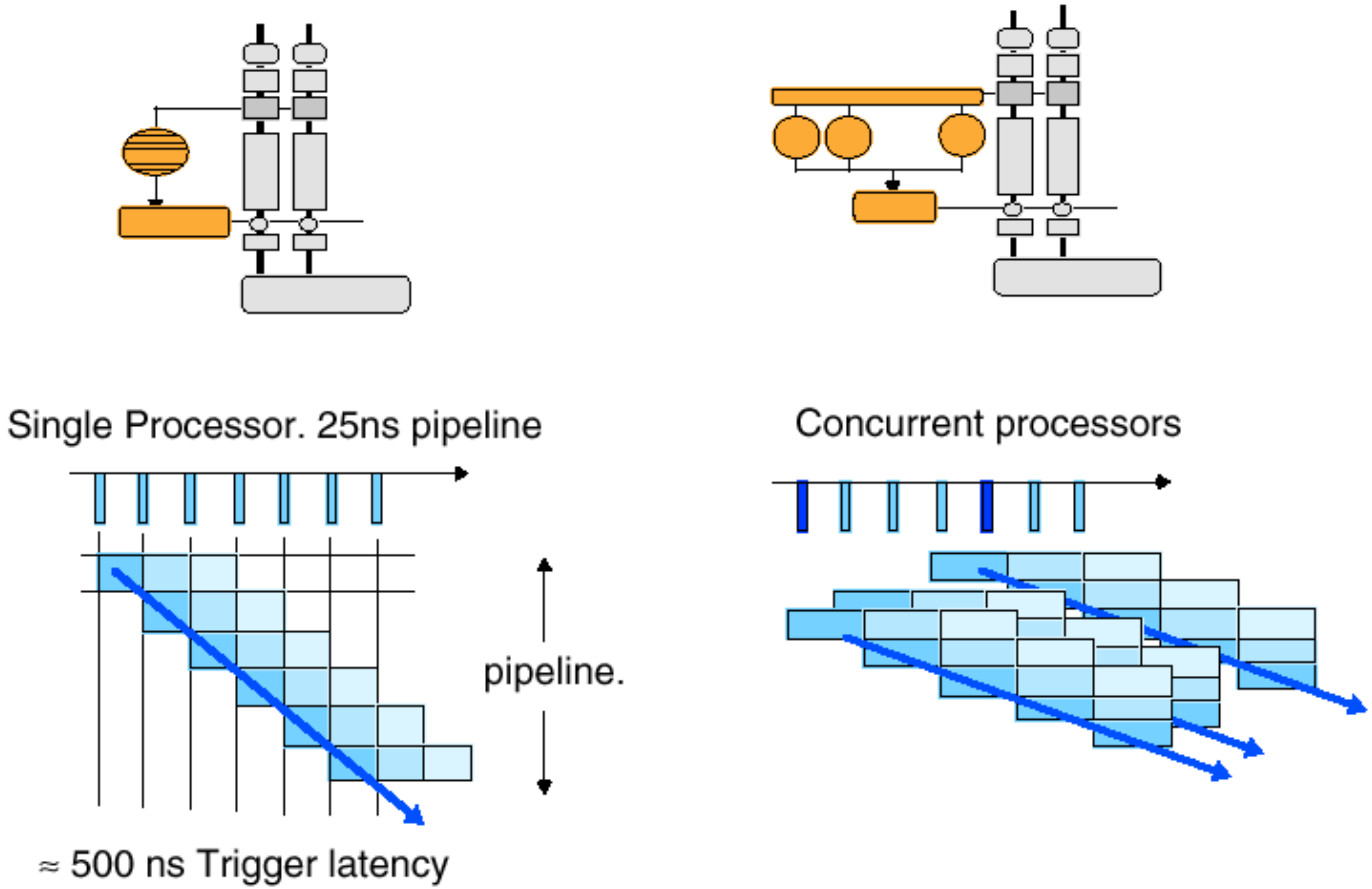
Level-1 pipeline trigger

- Every BC, a L1 trigger decision must be issued: since data are buffered in pipelines, the decision can be taken later, within **a fixed trigger latency**
  - Latency is given by the sum of the processing time of each step and the data transmission time

- It's necessary that the trigger **concurrently** processes many events
  - Perform operations in parallel within different processors
  - Divide the processing in steps, each performed within one BC



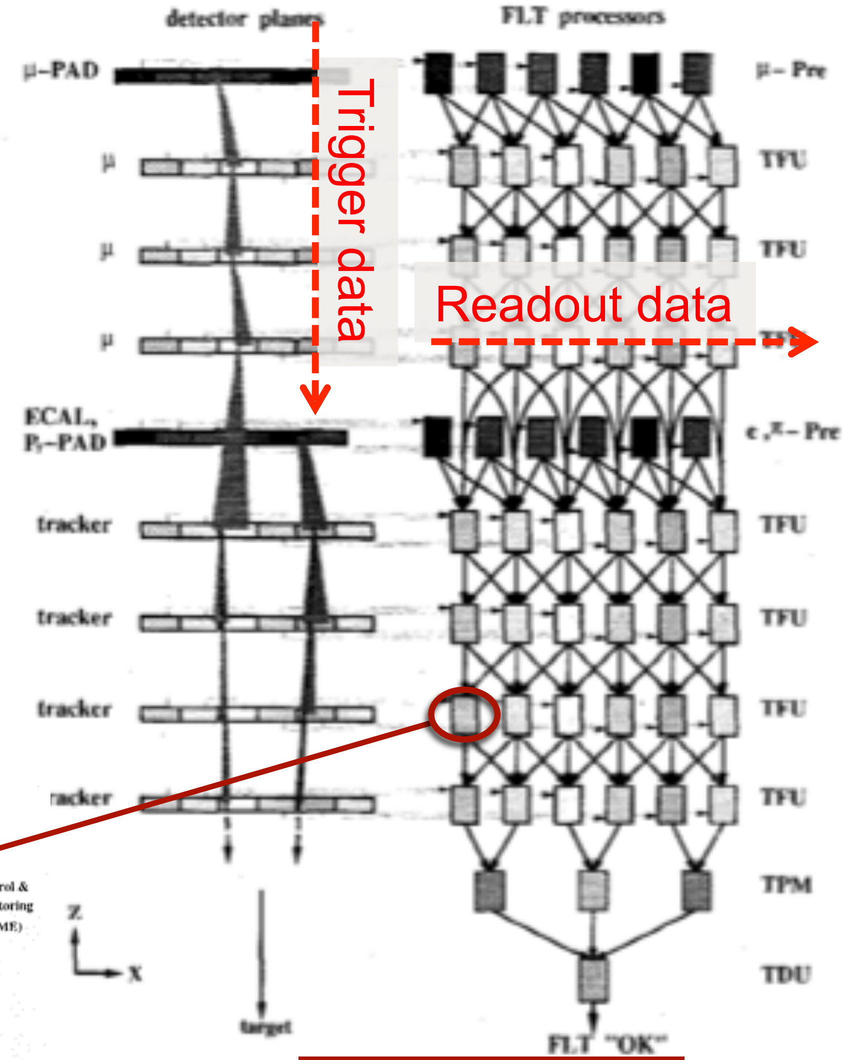
Level-1 processor architecture

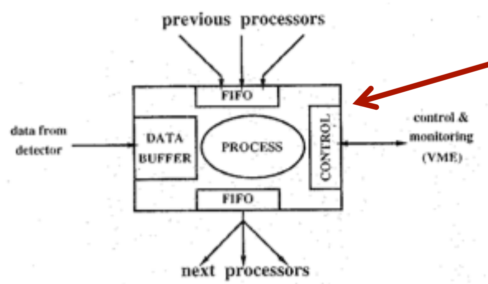


- Massive parallel and pipelined processing

Example: HERA-B

- Search for a **primary track** in the full acceptance
- **Iterative algorithm**: each step processes only a small *Region of Interest* (RoI) defined by the previous step
  - Each unit handles only the hit information for its corresponding 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 data memory **synchronously with BC clock** (left to right)
  - RoI data transferred asynchronously from unit to unit (top to bottom)

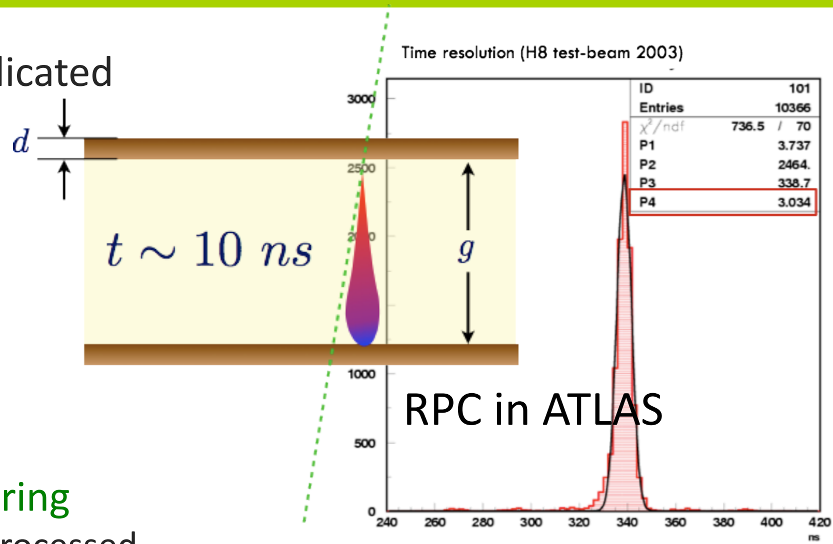




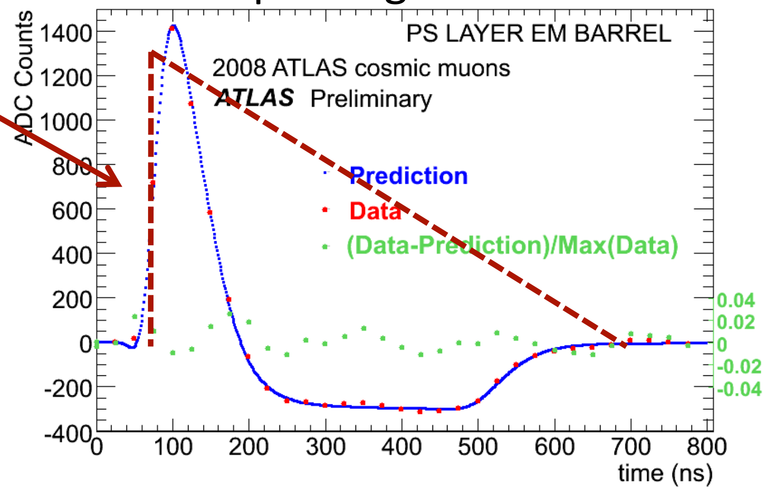
Track Finder Unit

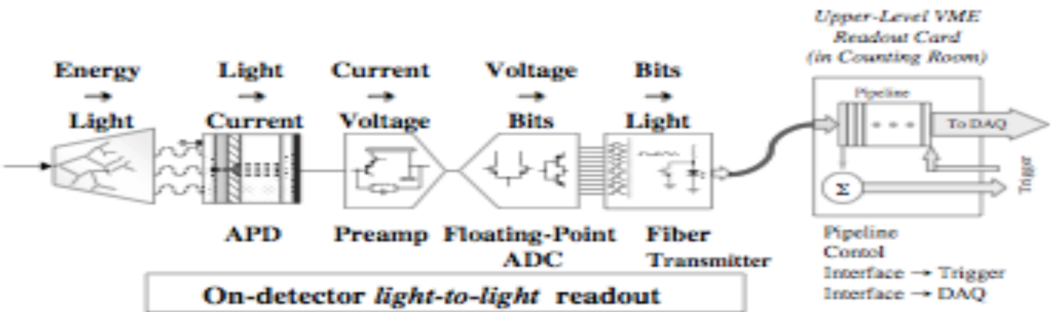
Chose your detector

- 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



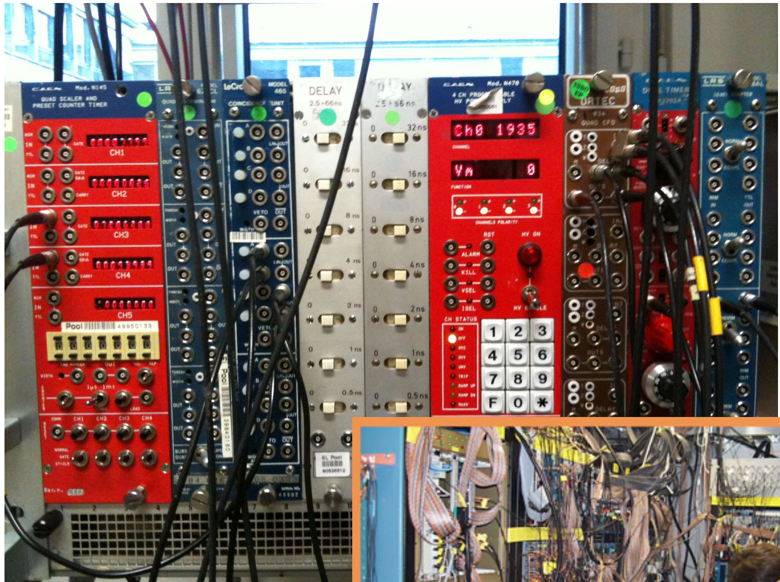
ATLAS Liquid Argon calorimeter



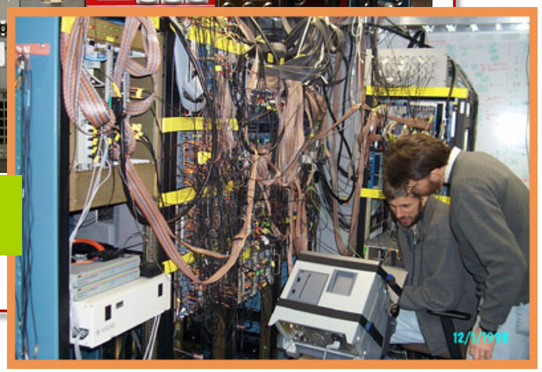


Choose L1 trigger your system

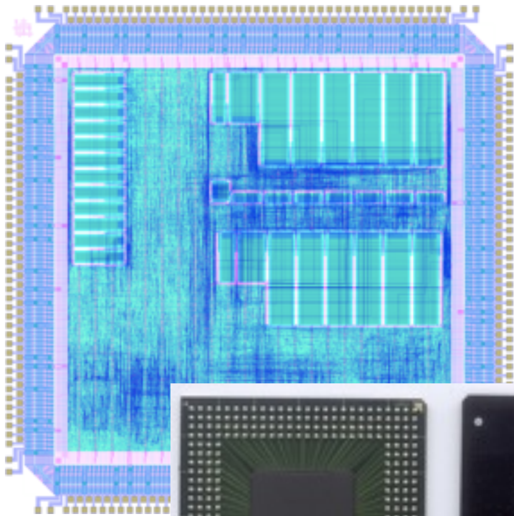
- Modular electronics
  - Simple algorithms
  - Low-cost
  - Intuitive and fast use



Your crate tomorrow

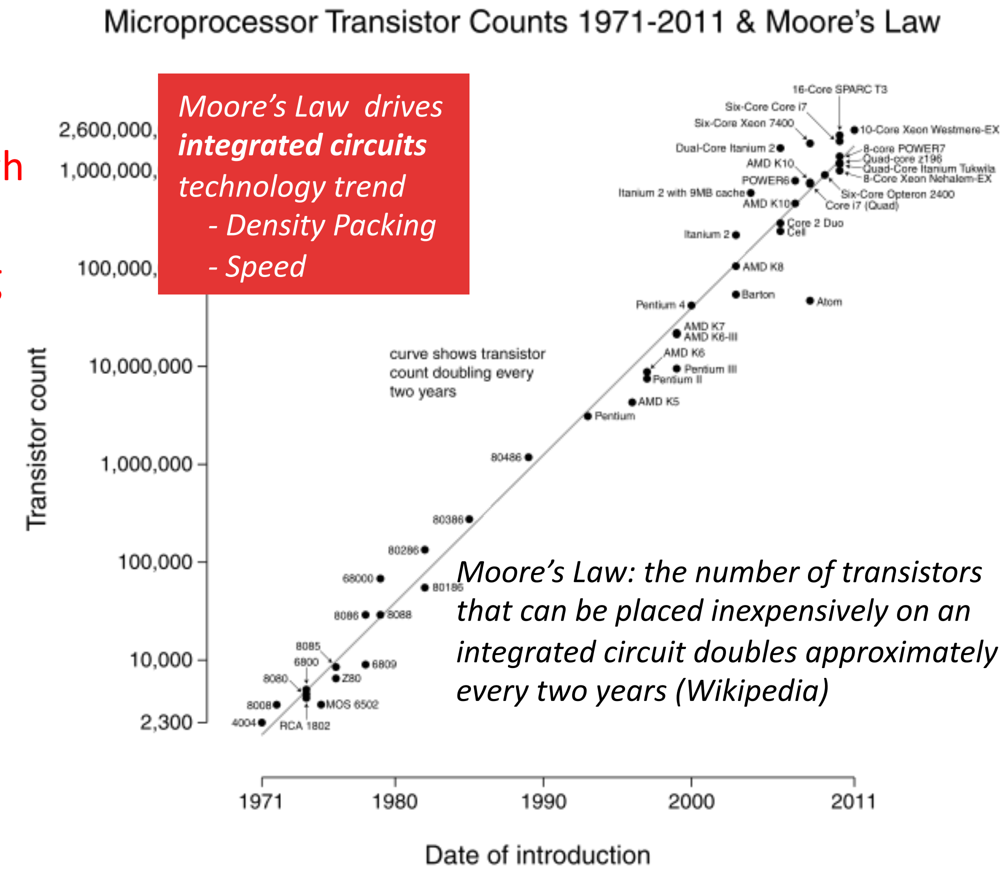


- Digital integrated systems
  - Highly complex algorithms
  - Fast signals processing
  - Specific knowledge of digital systems



Level-1 trigger technologies

- Requirements for high rate systems
  - Complex and flexible algorithms
    - **Programmable** solutions with high level languages
  - Data compression and formatting
  - Monitor and automatic fault detection

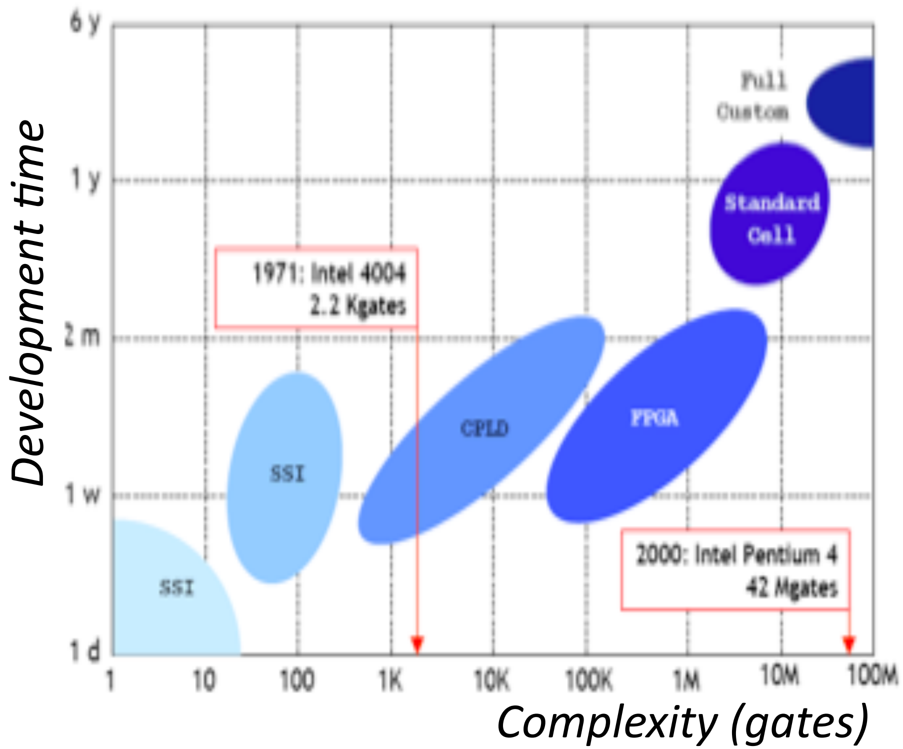


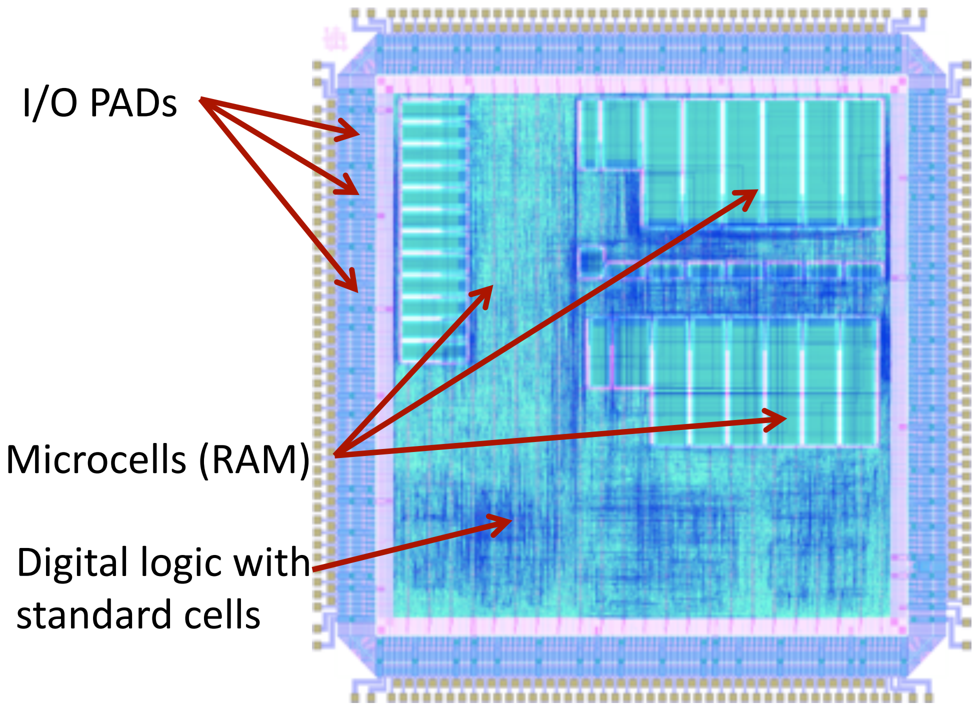
- Integrated circuits
  - Offer advantage in terms of reliability, reduced power usage, reduced boards and better performance

- Microprocessors
  - A single chip with all essential functions of a complete computer: CPU, memory, I/O ports, interrupt logic, **connected on a single bus**
  - Could be **embedded in the readout system**: read, buffer and process data close to the front-end electronics

Fast trigger processors

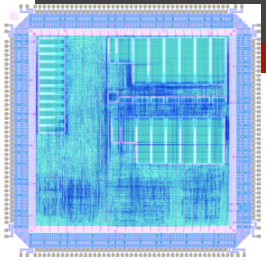
- Application-specific integrated circuits (**ASICs**): optimized for fast processing (Standard Cells, full custom)
  - Intel processors, ~ GHz
- Programmable ASICS (like Field-programmable gate arrays, **FPGAs**)
  - Easily find processors @ 100 MHz on the market (1/10 speed of full custom ASICs)



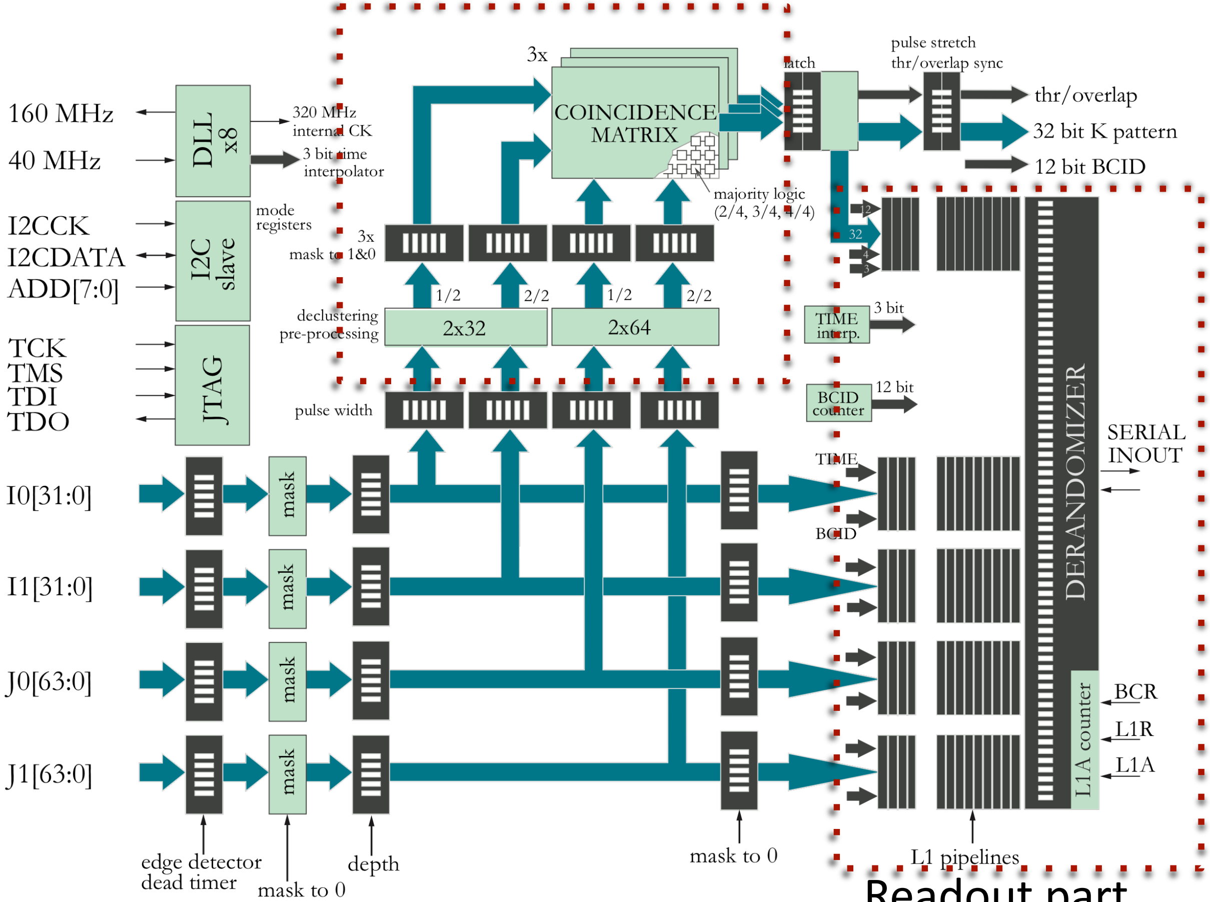


Layout of the CM ASIC

Example: logic of a trigger ASIC



Trigger part



Readout part

Trigger output

Readout output

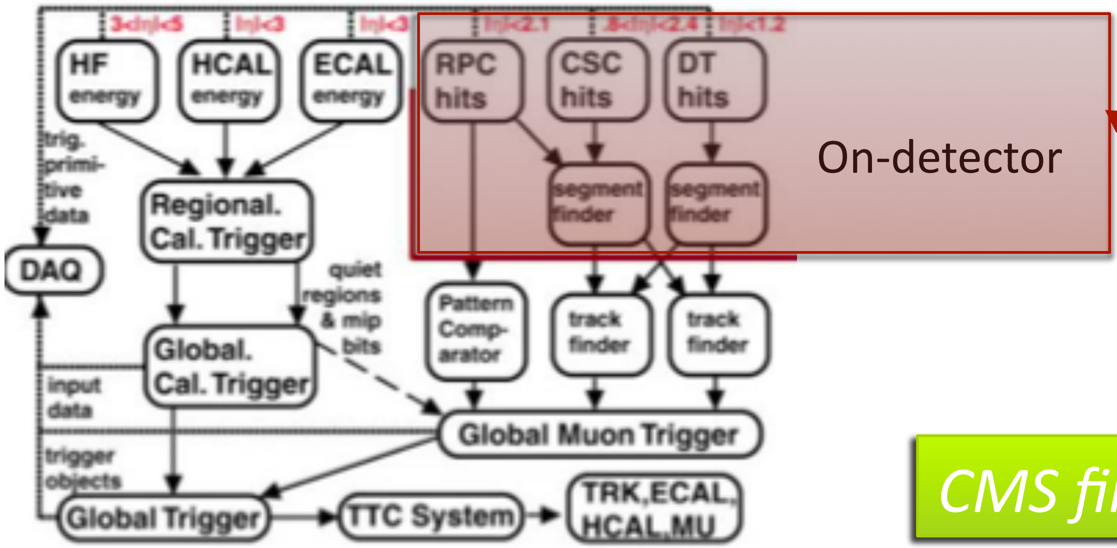
Digital inputs from detector

Input: synchronization signals

Coincidence Matrix ASIC for Muon Trigger in the Barrel of ATLAS

Data movement technologies

- A trigger system is made of different components
  - Some elements have to be mounted on the detector (**on-detector**), some others can be placed into crates with bus connections (**off-detector**)



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

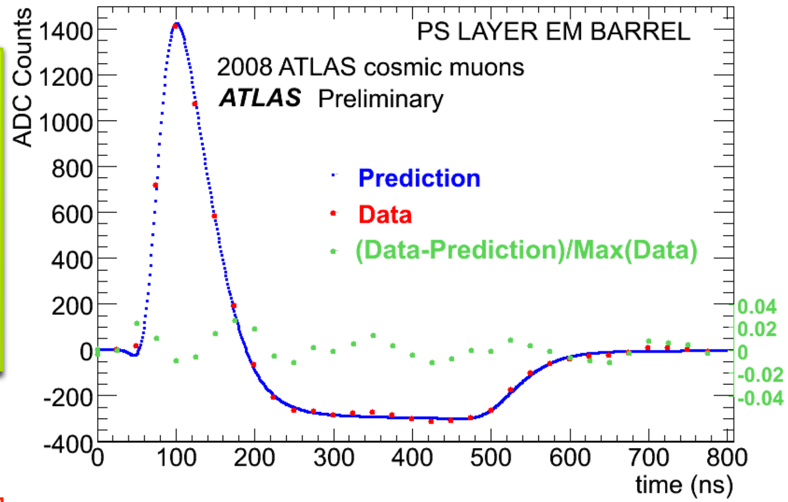
CMS first-level trigger

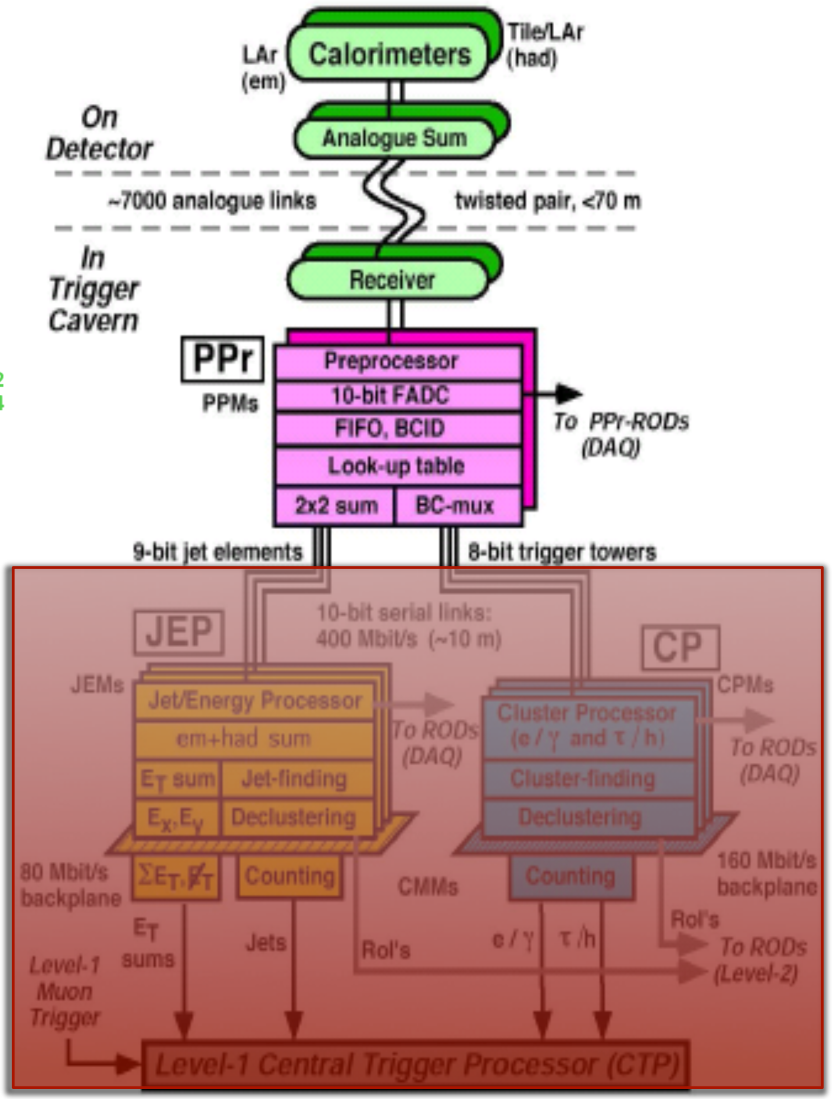
- **High-speed** serial links, electrical and optical, over a variety of distances
  - Low cost and low-power LVDS links, @400 Mbit/s (up to 10 m)
  - Optical GHz-links for longer distances (up to 100 m)

- **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

Pulse of the Liquid-Argon Calorimeter in ATLAS





On-detector

- Sum of the analog signals from cells to form **trigger towers**

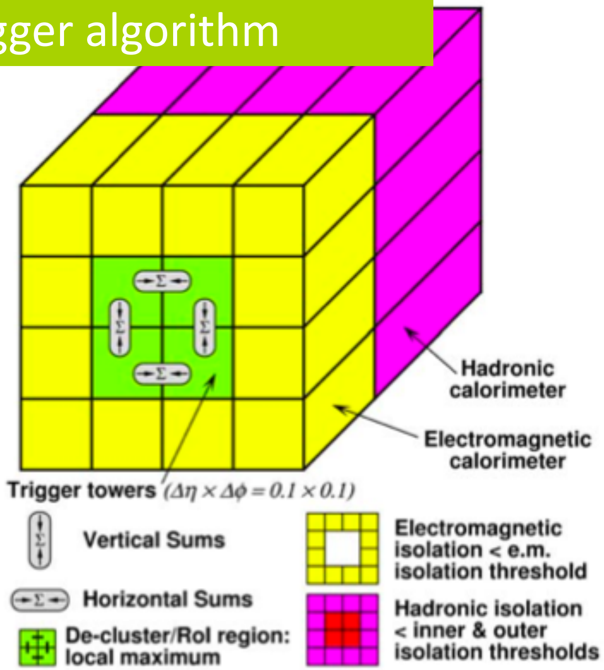
Pre-processor

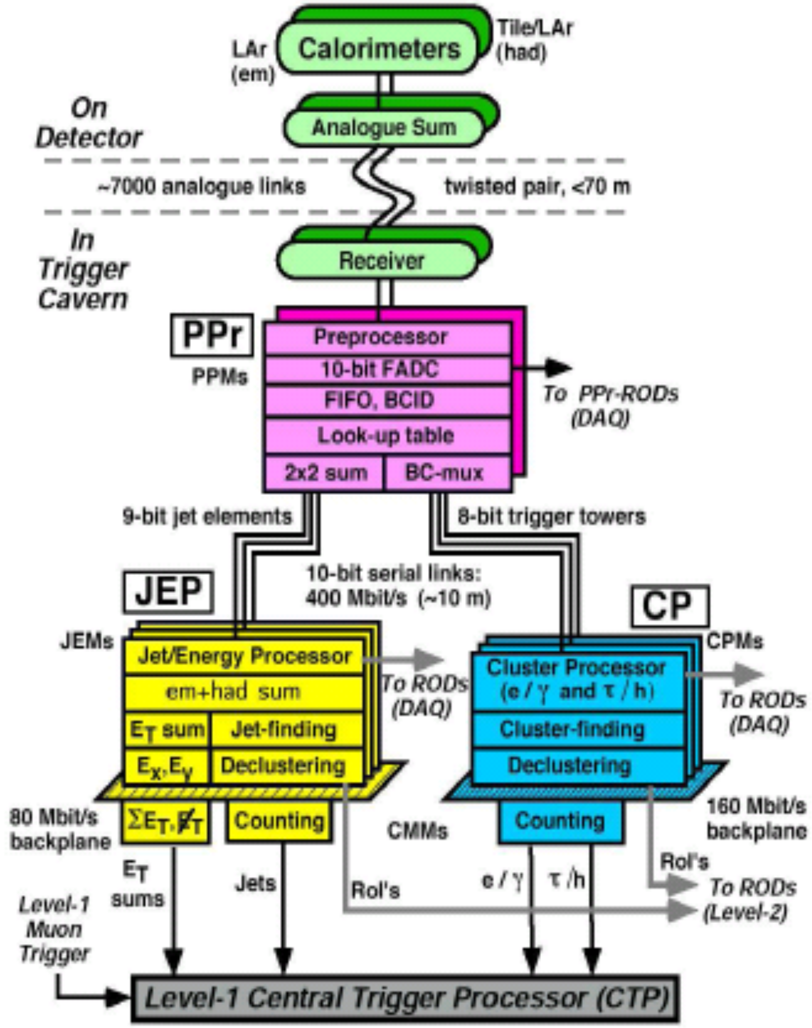
- Digitized pulse shape: 10-bit resolution
- Add the trigger algorithm
  - Assign each bin an energy (ET) via Look-Up tables
  - Apply trigger threshold on ET
- Signal over 8 BCs
  - Peak-finder algorithm to assign the correct BC

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

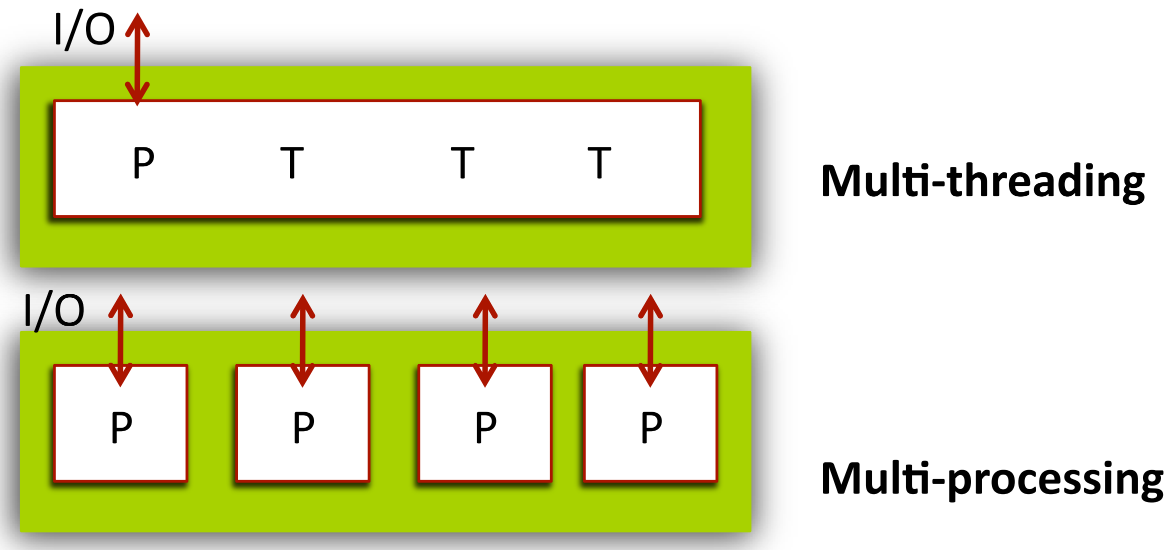


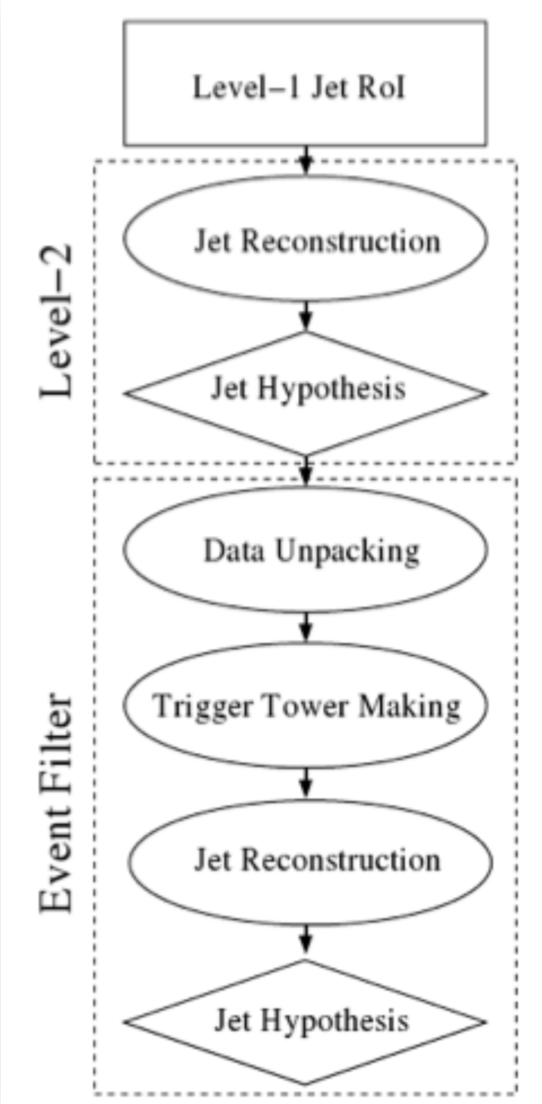


High level triggers

HLT design principles

- Early rejection
  - Alternate steps of feature extraction with hypothesis testing: events can be rejected at any step with a complex algorithm scheduling
- Event-level parallelism
  - Process more events in parallel, with multiple processors
  - Multi-processing or multi-threading
  - Queuing of the shared memory buffer within processors
- Algorithms are developed and optimized offline, often software is common to the offline reconstruction





High Level Trigger Architecture

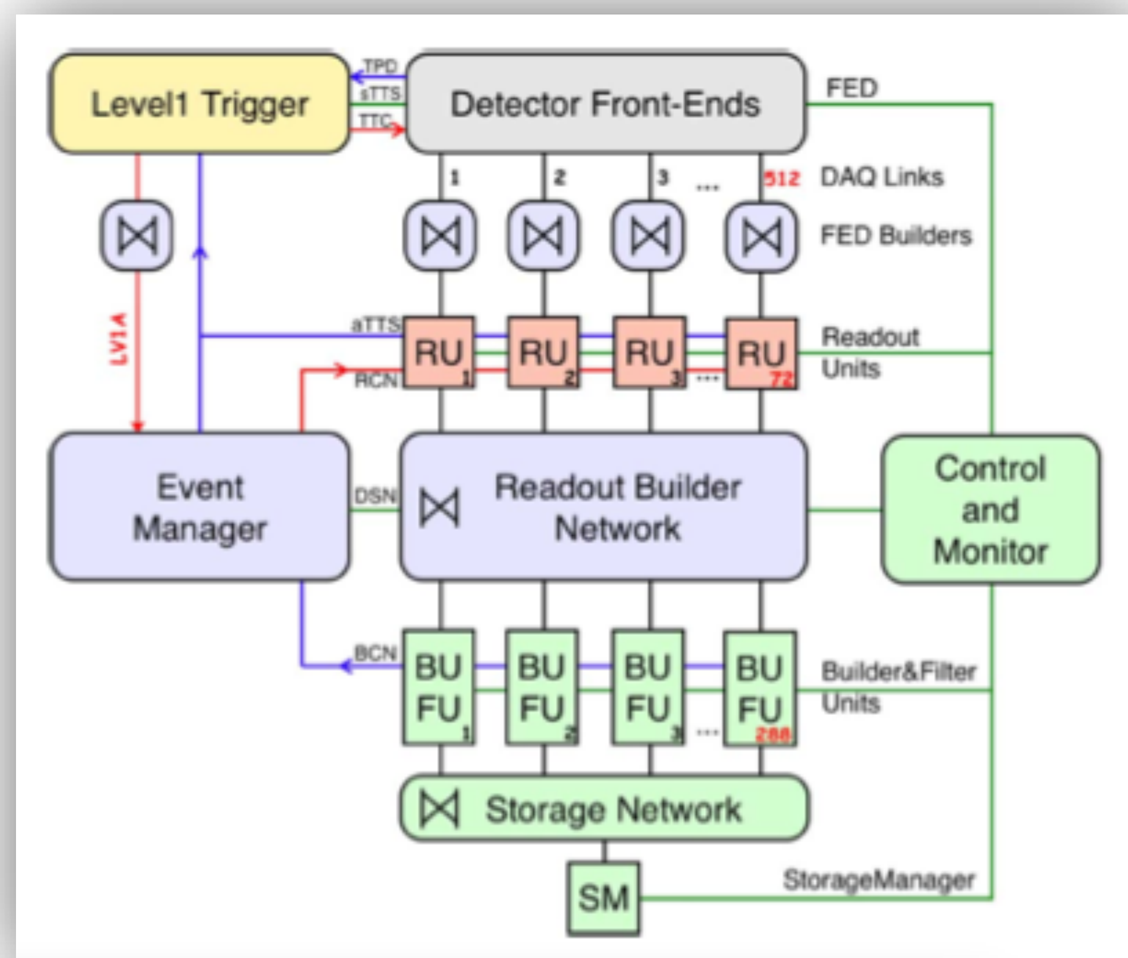
- After the L1 selection, data rates are reduced, but can be still massive
- Key parameter for the design is the **allowed bandwidth**, given by the average event-size and the trigger rate
  - **LEP**: 100 kByte event-size @ few Hz gives **few 100 kByte/s**
    - Supported by 40 Mbyte/s VME bus
  - **ATLAS/CMS**: 1 MByte event-size @100 kHz gives **~100 GByte/s**

	N.Levels	L1 rate (Hz)	Event size (Byte)	Readout bandw. (GB/s)	Filter out MB/s (Event/s)
ATLAS	3	L1: 10^5	10^6	10	~100 (10^2)
		L2: 10^3			
CMS	2	10^5	10^6	100	~100 (10^2)

- **Latest technologies** in processing power, high-speed network interfaces, optical data transmission
- High data rates are held by using
  - Network-based event building
  - Seeded reconstruction of data

Network-based HLT: CMS

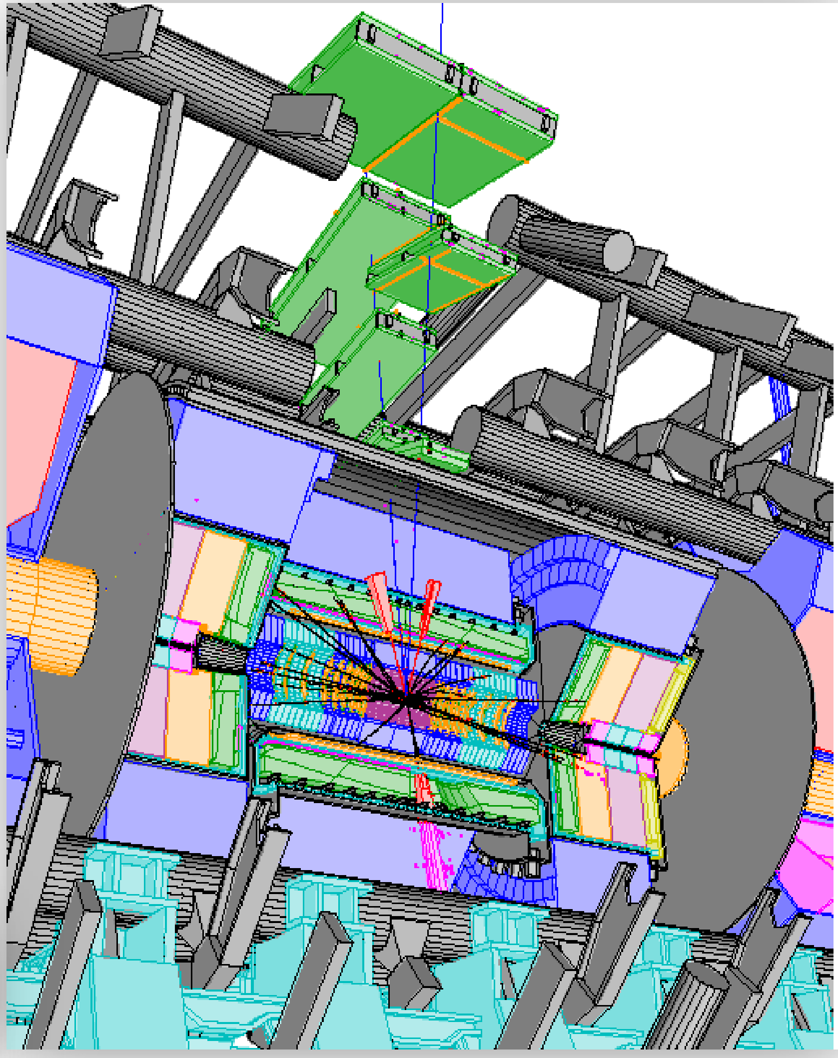
- Data from the readout system (RU) are transferred to the filters (FU) through a builder network
- Each filter unit processes only a fraction of the events
- Event-building is factorized into a **number of slices**, each one processing only 1/nth of the events
  - Large total bandwidth still required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



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

Seeded reconstruction HLT: ATLAS

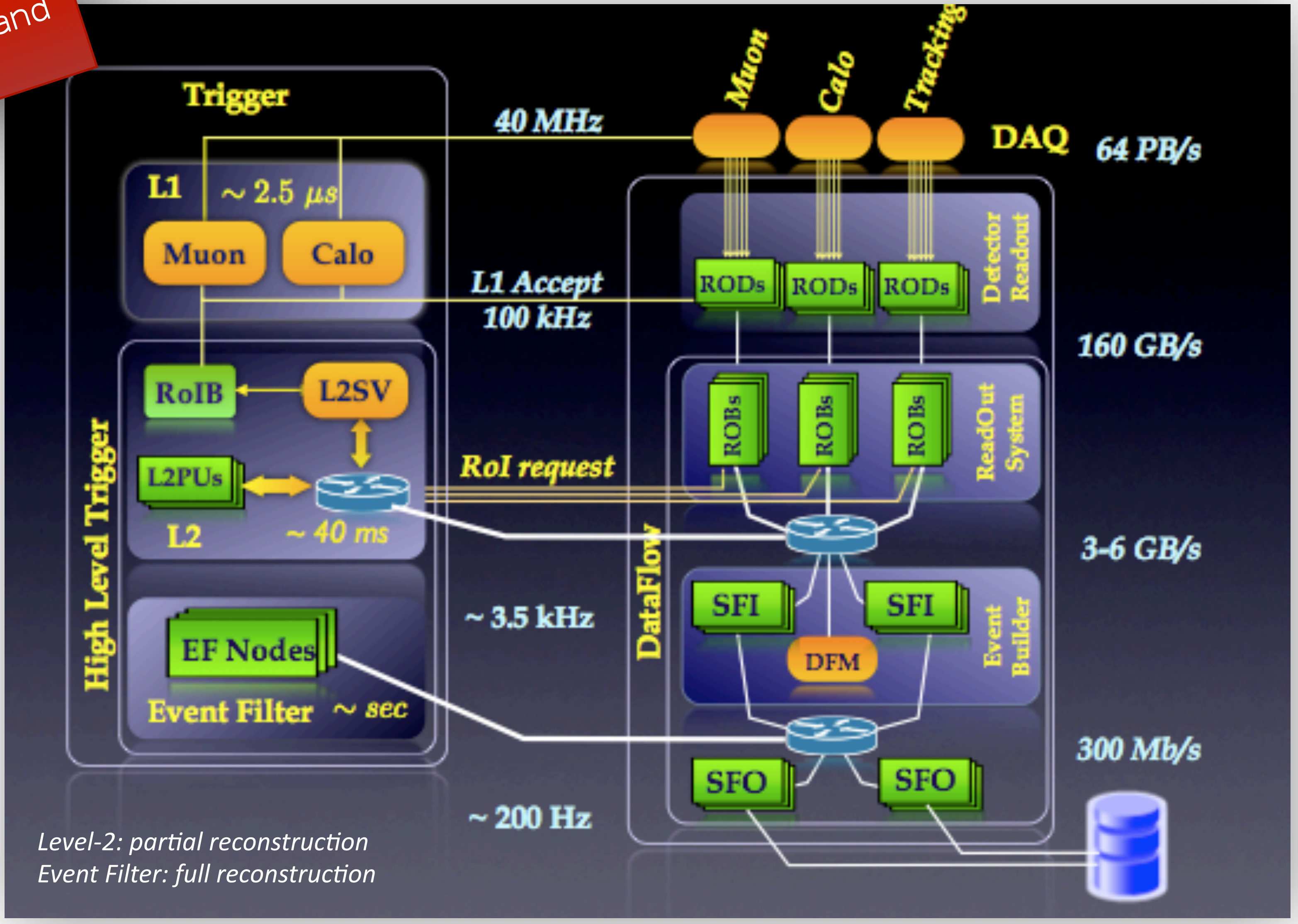
- Level-2 uses the information seeded by level-1 trigger
  - Only the data coming from the region indicated by the level-1 is processed, called **Region-of-Interest (RoI)**
  - The resulting total amount of RoI data is minimal: a few % of the Level-1 throughput
  - Level-2 can use the full granularity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



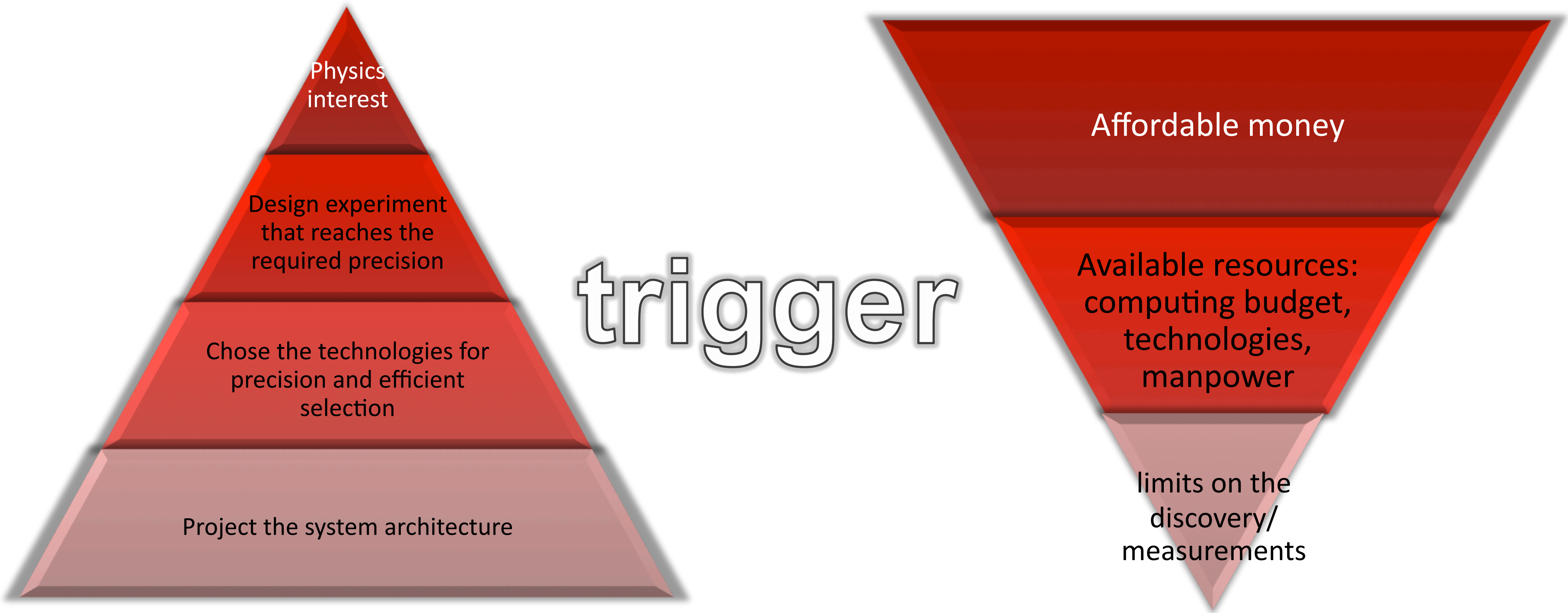
Typically, there are less than 2 RoIs per event accepted by LVL1

ATLAS TDAQ system

Note rates and latencies



The trigger connections



- The trigger system is connected with all of them: needs that the experts on each field work together to maximize the available resources

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 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!!