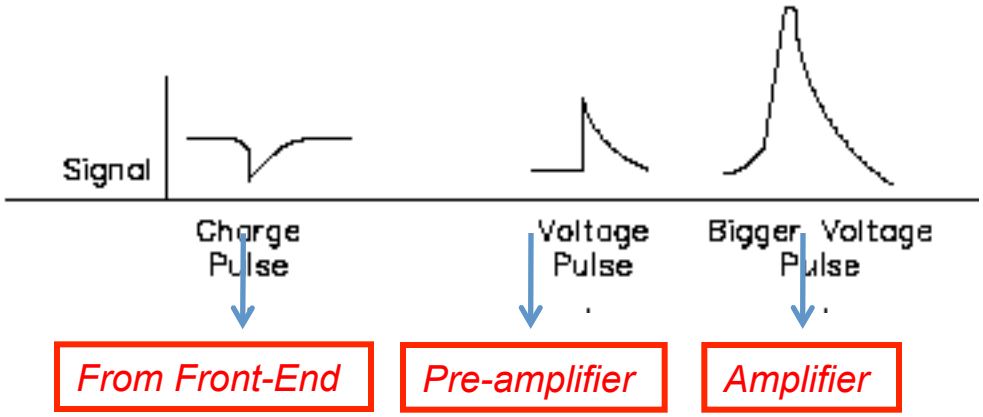
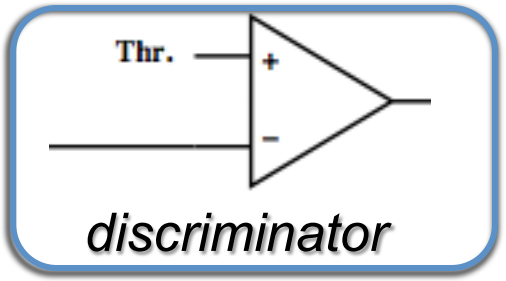
Trigger architectures

F.Pastore (Royal Holloway Univ. of London)
francesca.pastore@cern.ch

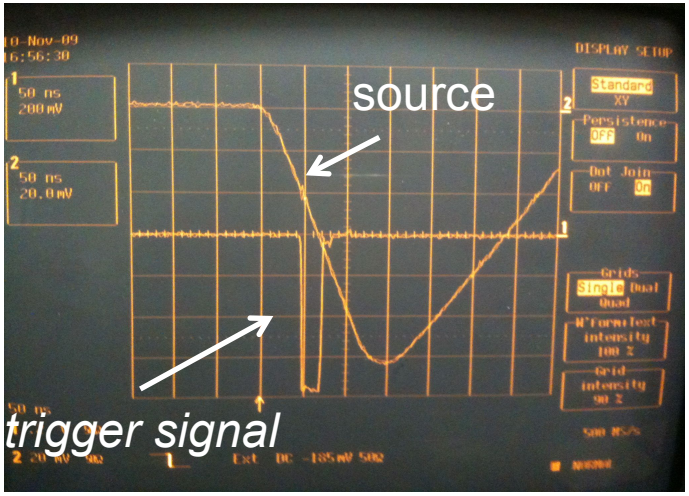
The simplest trigger system

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



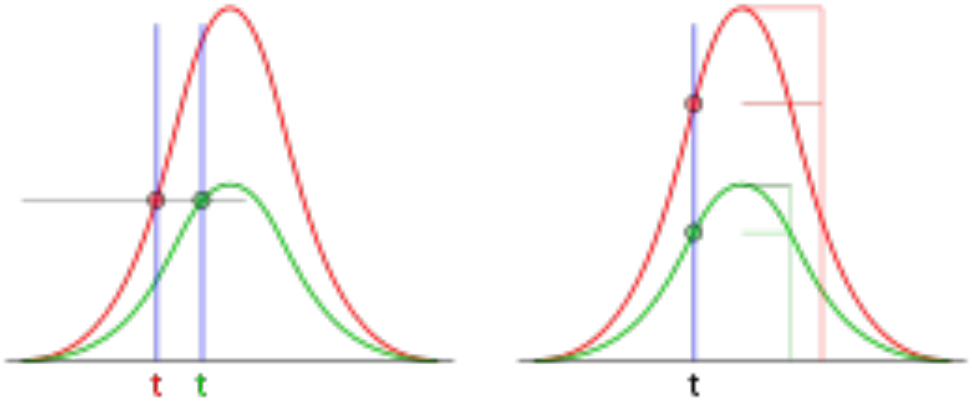


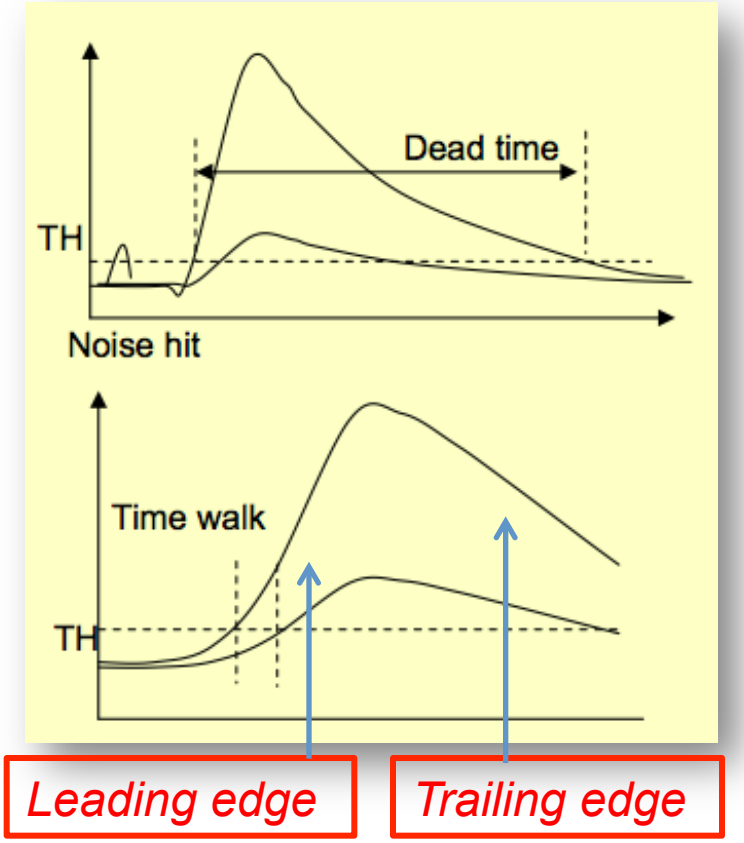
- The simplest trigger: apply a threshold
  - Look at the signal
  - Apply a threshold as low as possible, since signals in HEP detectors have large amplitude variation
  - Compromise between hit efficiency and noise rate



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 minimized in a good trigger system





- 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

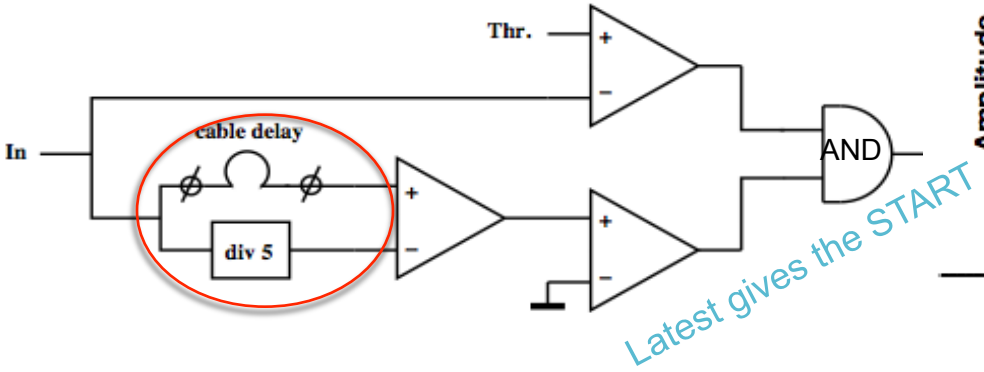
The constant fraction discriminator

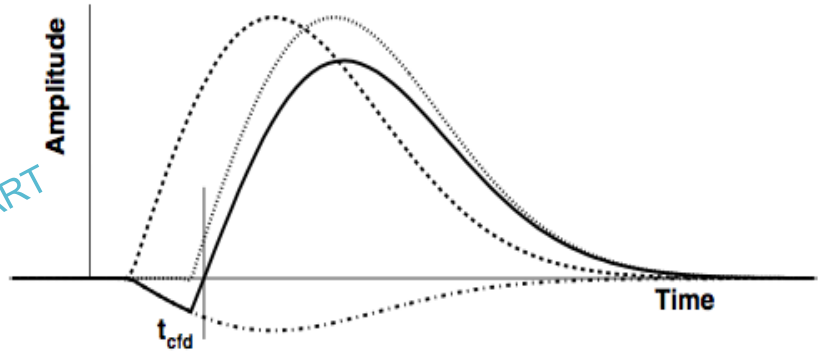
If two signals have the same rising time at a fraction **f**

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

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

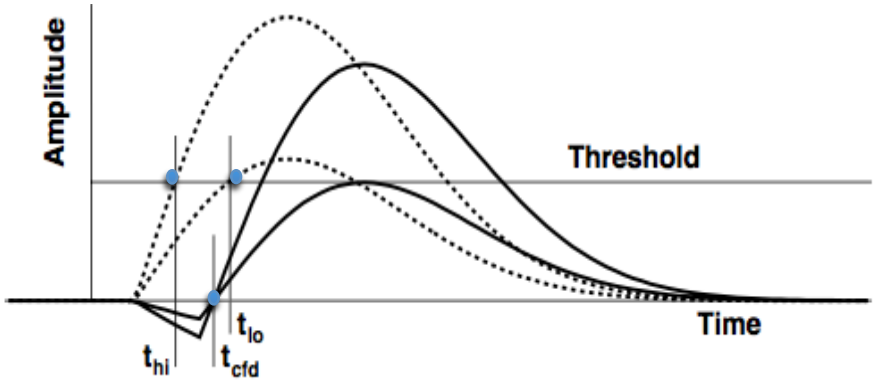
Input pulse
Delayed input pulse
Attenuated inverted input
Bipolar pulse





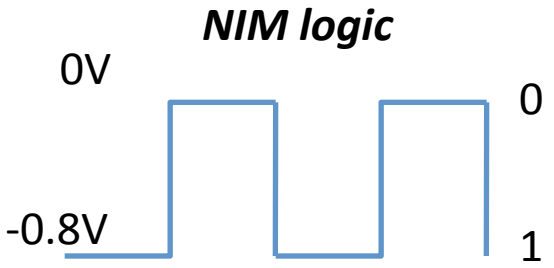
The output of the CFD fires when the bipolar pulse changes polarity

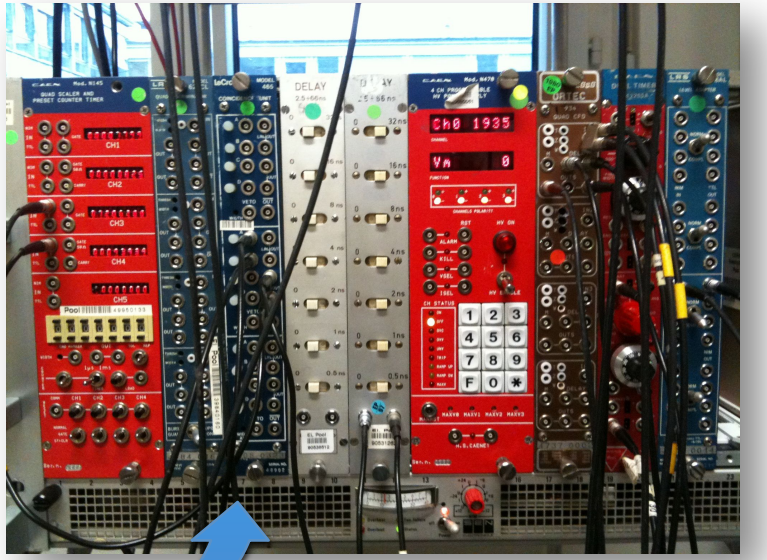
- Attenuation and delay (configurable) applied before the discrimination determines t_{CFD}
- If the delay is too short, the unit works as a normal discriminator because the output of the normal discriminator fires later than the CFD part

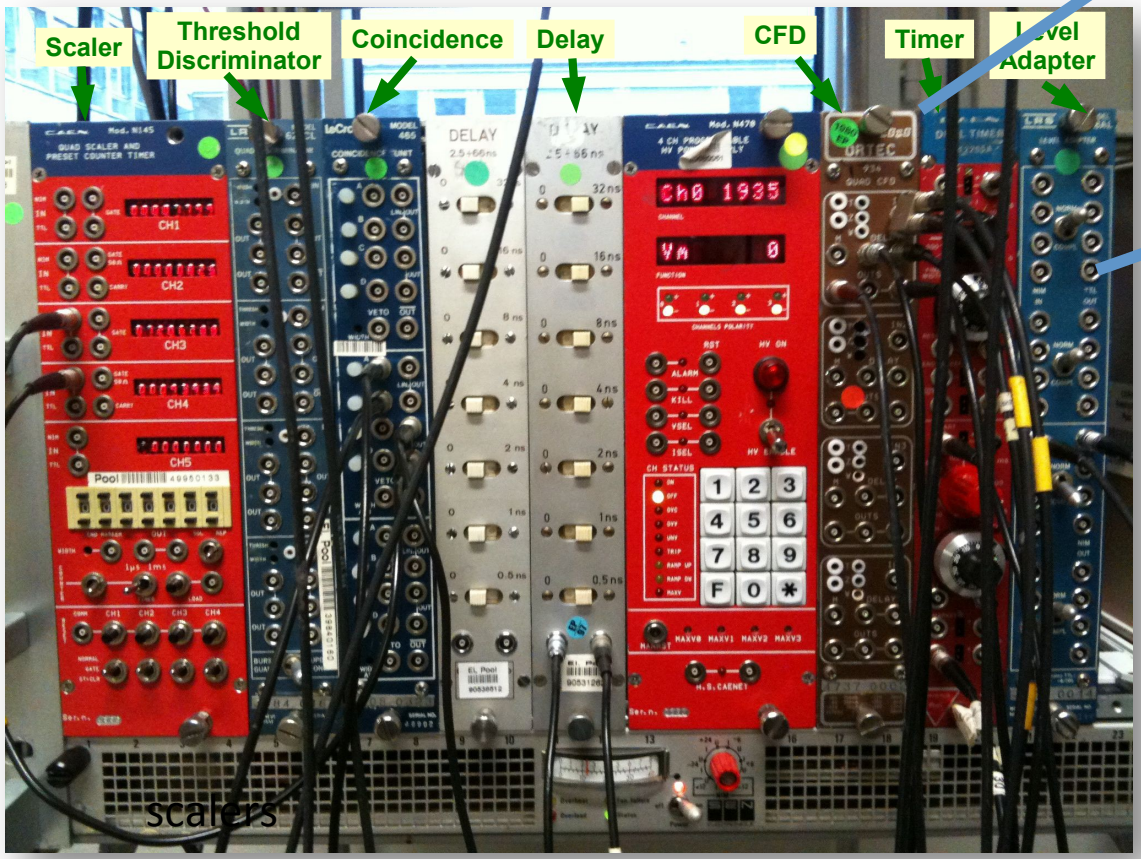


And now build your own trigger system

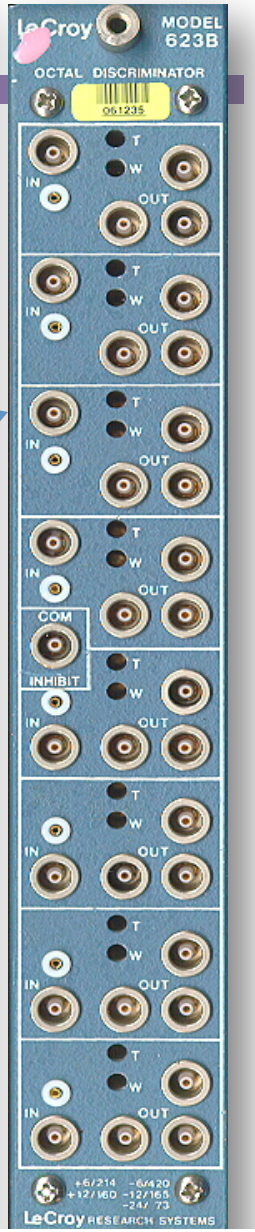
- A simple trigger system can start with a NIM crate
- Common support for electronic modules, with standard impedance, connections and logic levels: negative (at -16 mA into 50 Ohms = -0.8 Volts)







ORTEC CFD



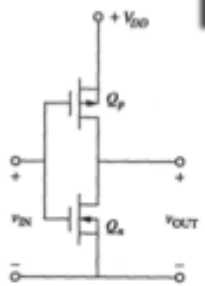
LeCroy discriminator

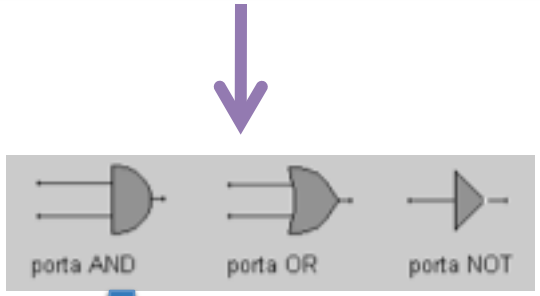
- Threshold levels configurable via screwdriver adjust

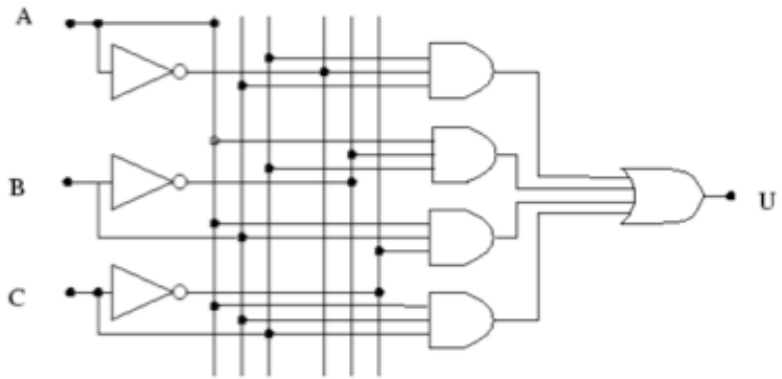
Trigger logic implementation

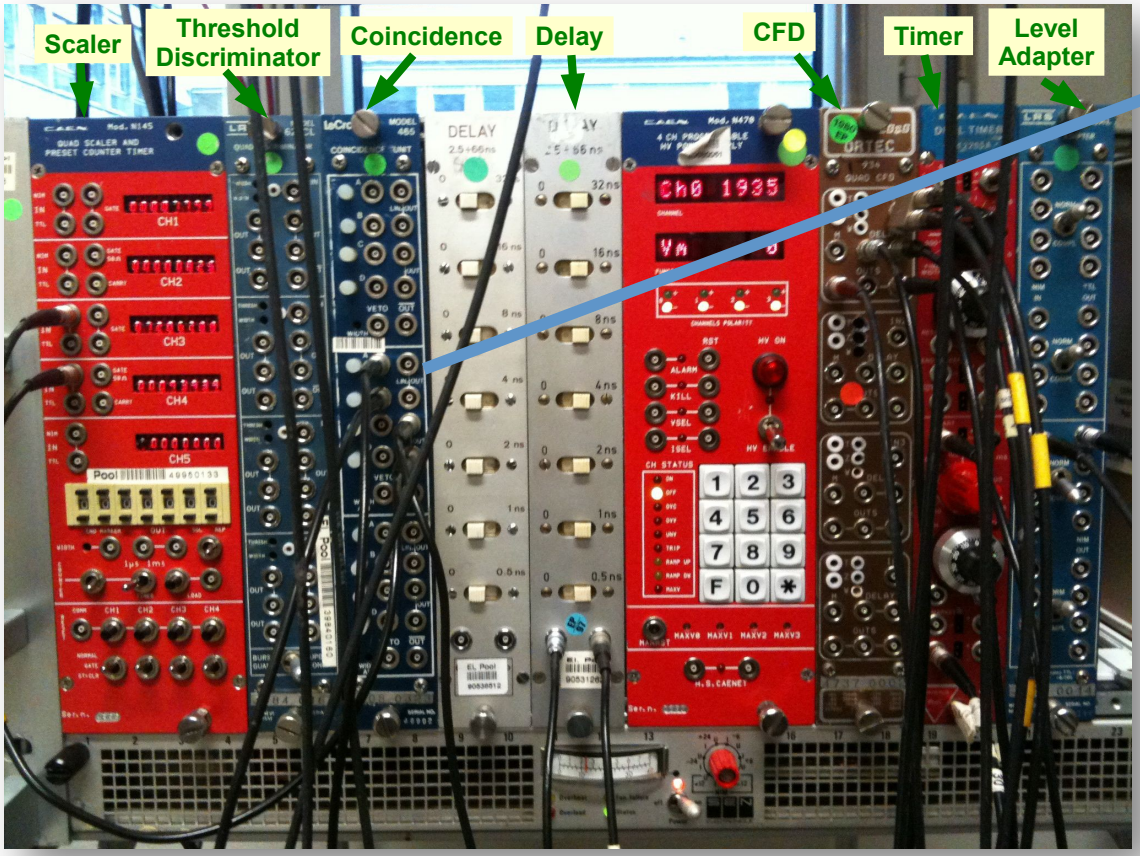
- Analog systems: amplifiers, filters, comparators, ….
- Digital systems:
  - Combinatorial: sum, decoders, multiplexers,….
  - Sequential: flip-flop, registers, counters,….
- Converters: ADC, TDC, …..

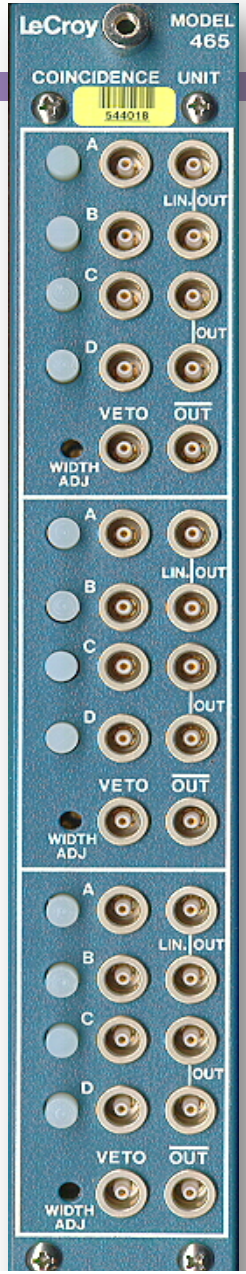
Any behavior of our system can be described logically with a sequence of mathematical operators











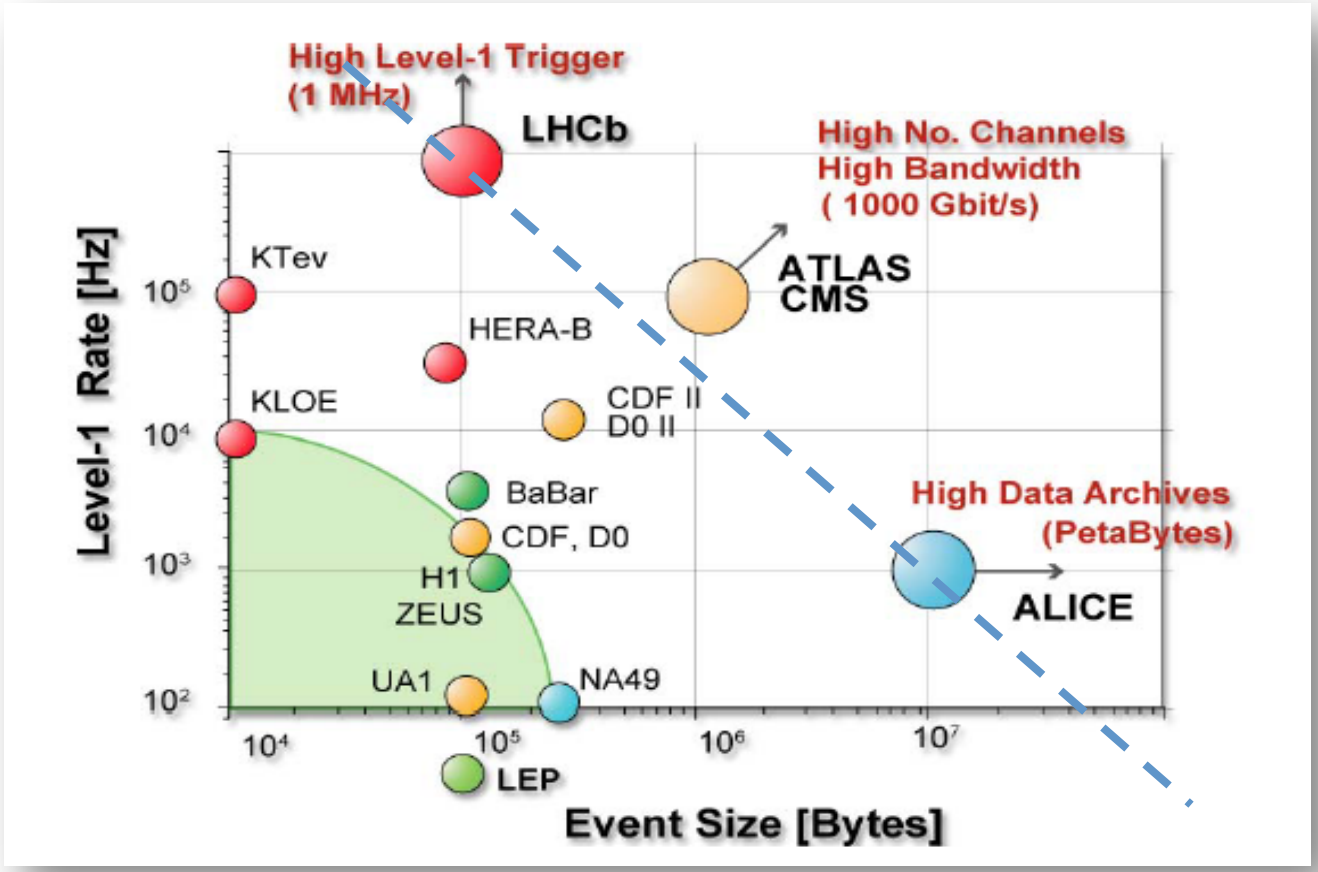
LeCtroy Concidence Unit

Summary of the trigger requirements

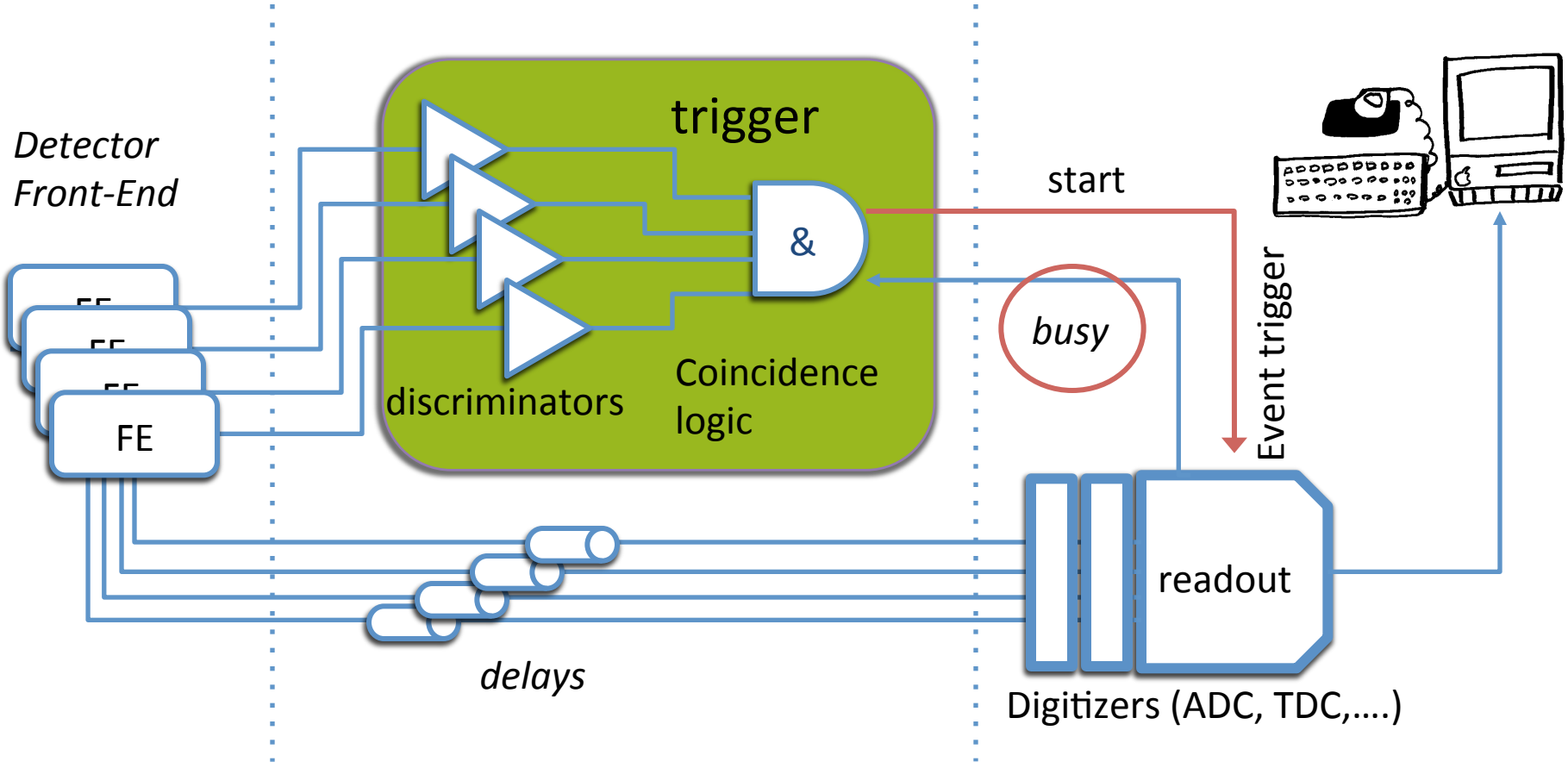
- High Efficiency
  - Low dead-time
  - Fast decision
- Reliability and robustness
- Flexibility

Trigger and data acquisition trends

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



A simple trigger system

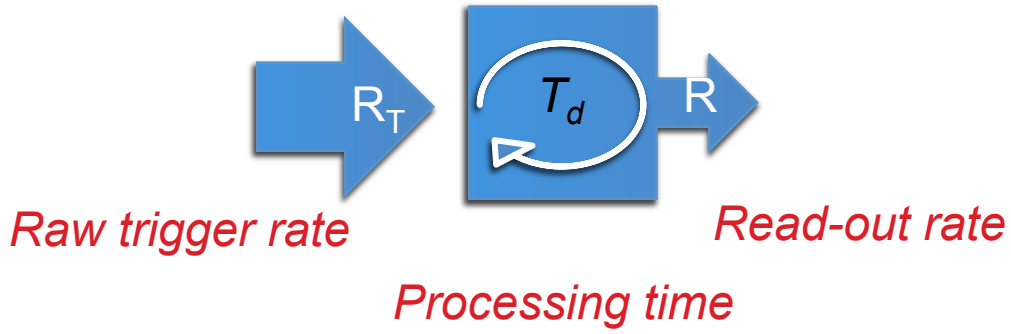


- Due to **fluctuations**, the incoming rate can be higher than processing one
- Valid interactions can be rejected due to system **busy**

Dead-time

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

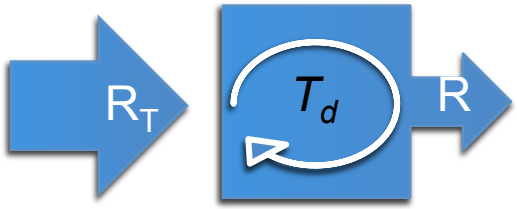
- **Fluctuations produce dead-time!**
  - Remind: due to fluctuations, the incoming rate can be higher than the processing one
- The most important parameter controlling the design and performance of high speed **T-DAQ systems**
  - Occurs whenever a given step in the processing takes a **finite amount of time**
  - The fraction of the acquisition time in which no events can be recorded can be typically of the order of **few %**



- Mainly three sources:
  - Readout dead-time
  - Trigger dead-time
  - Operational dead-time

Maximize event recording rate

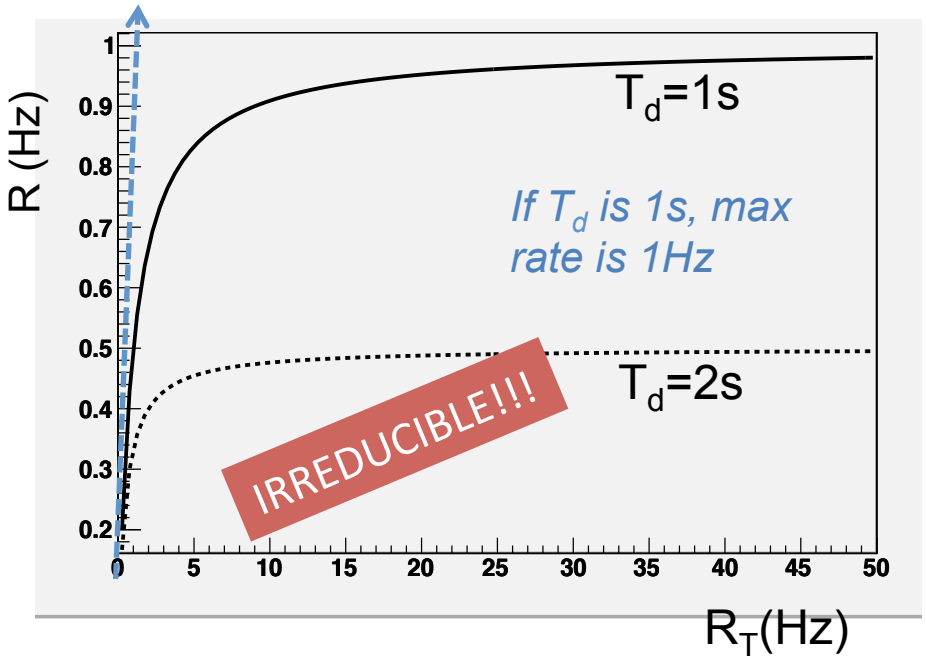
R_T = raw trigger rate (average)
R = number of events read per second (DAQ rate)
T_d = readout time interval per event



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

Fraction of surviving events!

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

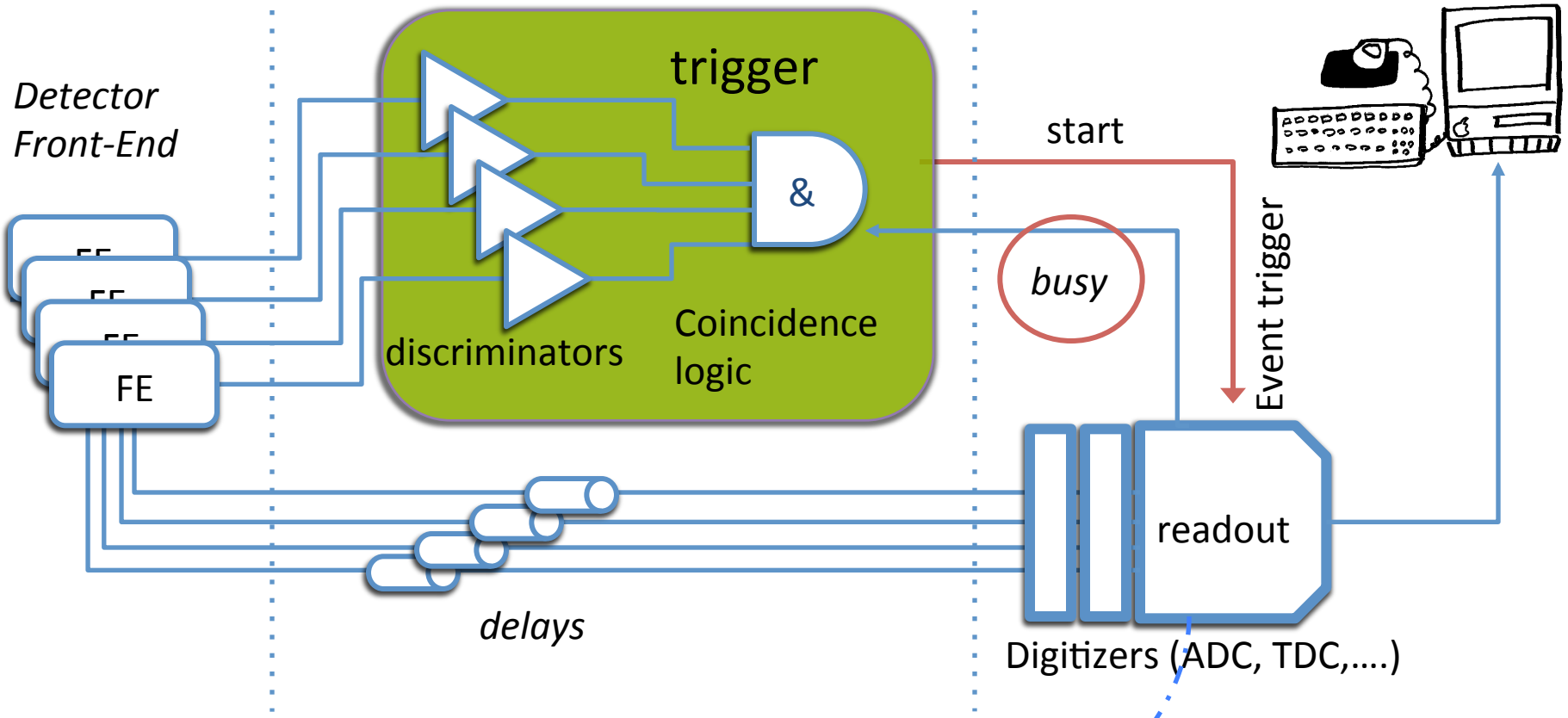


- We always lose events if $R_T > 1/T_d$
- If exactly $R_T = 1/T_d$ -> dead-time is 50%

The trick is to make both R_T and T_d as small as possible (R~R_T)

FAST TRIGGER!
LOW INPUT RATE!

A simple trigger system



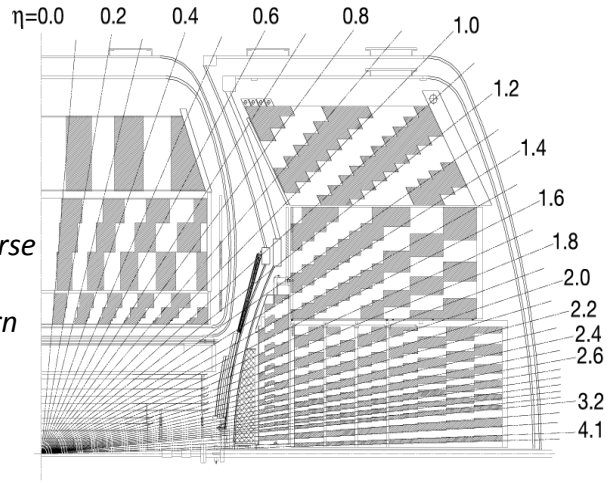
$$D_t = R \cdot T_{RO}$$

Fraction of lost events due to finite readout

Features to minimize dead-time

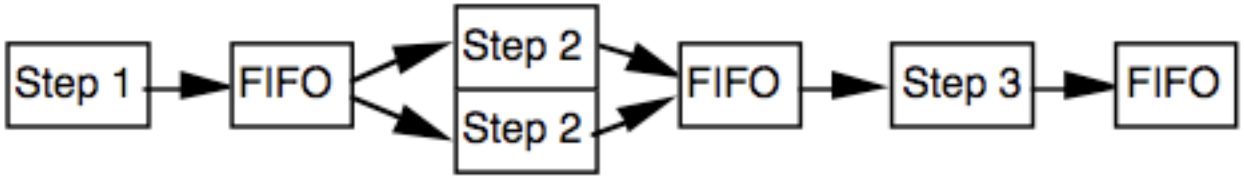
- **1: Parallelism**
  - Independent readout and trigger processing paths, one for each sensor element
  - Digitization and DAQ processed in parallel (as many as affordable!)

Segment as much as you can!



DZero calorimeters showing the transverse and longitudinal segmentation pattern

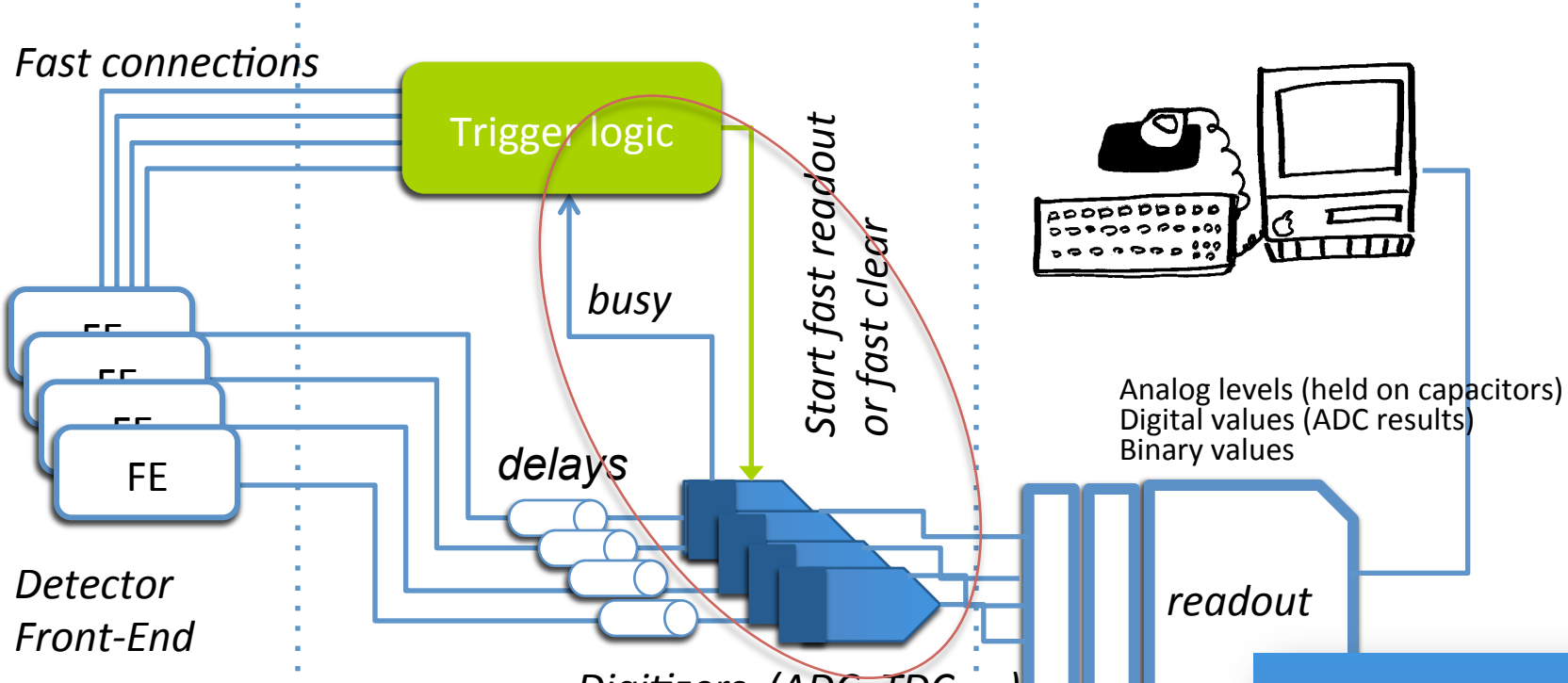
- **2: Pipeline processing** to absorb fluctuations
  - Organize the process in different steps
  - Use local **buffers** (FIFOs) between steps with different timing (big events processed during short events)
  - The depth of local buffers limits the processing time of the subsequent step



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

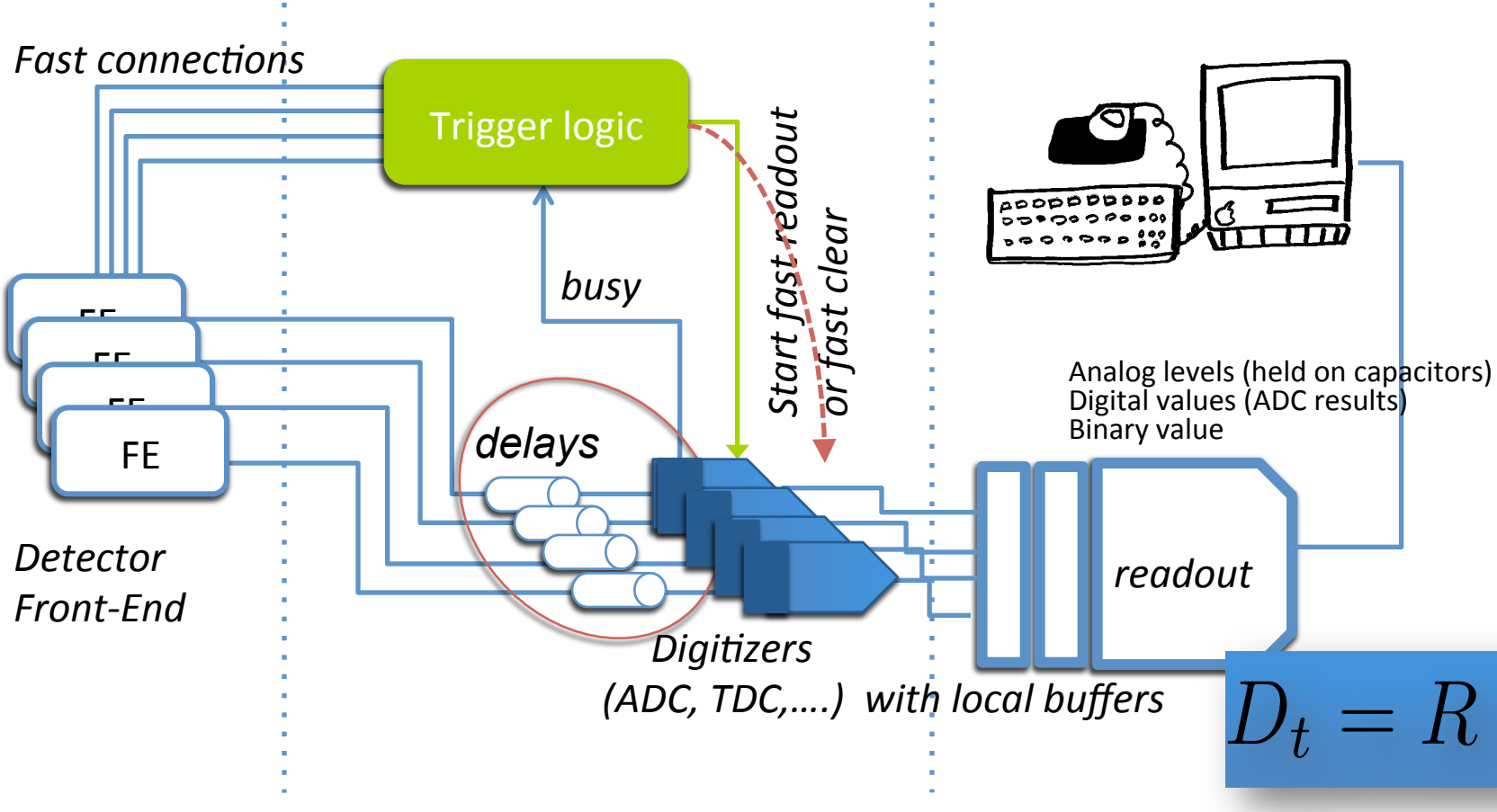
Try to absorb in capable buffers

Minimizing readout dead-time…



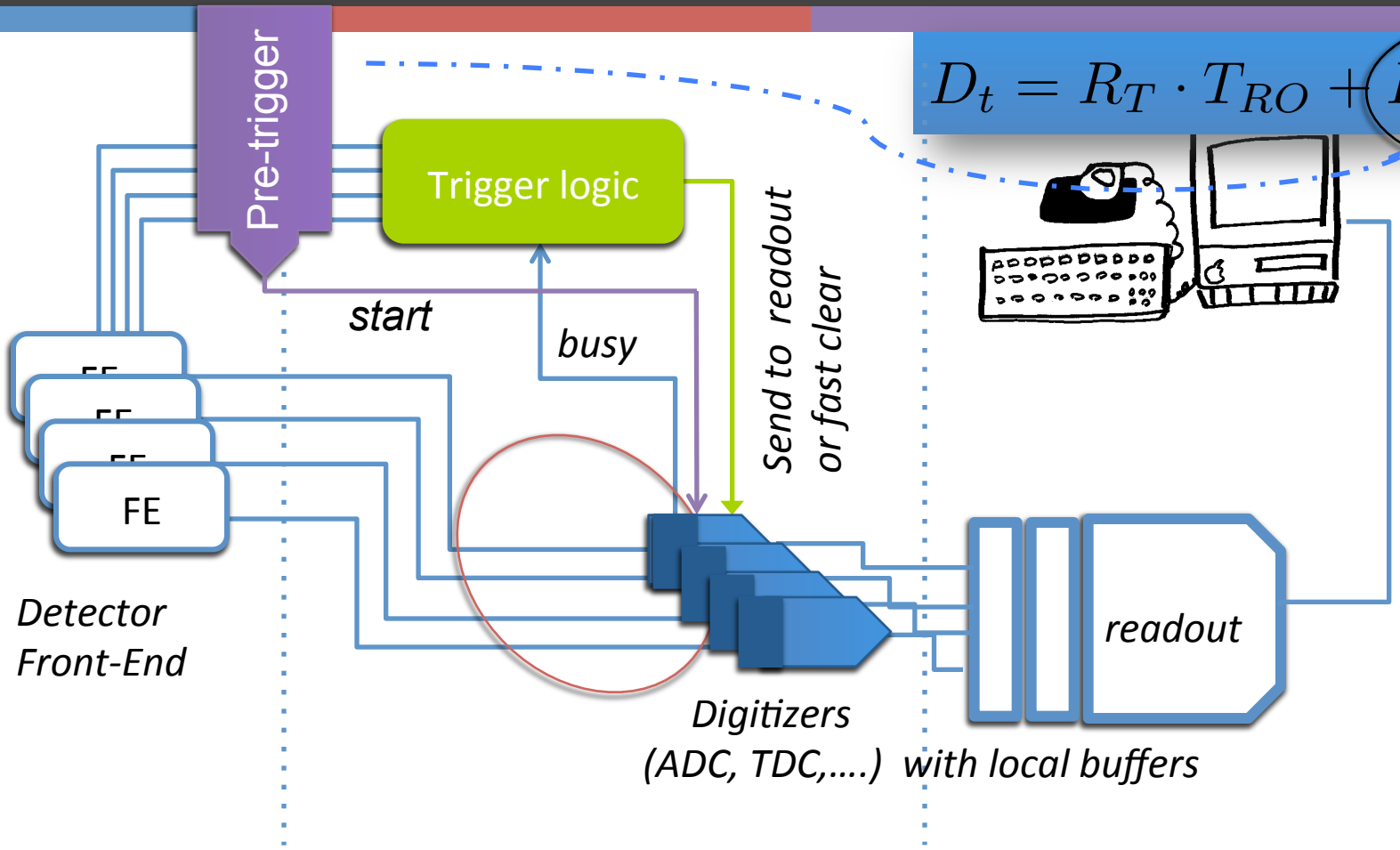
- **Parallelism**: Use multiple digitizers
  - The trigger sends a fast readout or a fast clear command to all local buffers
- **Pipelining**: Different stages of readout: fast local readout + global event readout (slow)
  - Dead-time is the product of the trigger rate and the **fast readout time**

Trigger latency



- 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
  - Must be available when the signals from the detectors arrive at the digitizers
  - **START** the digitizers, confirmed later by the main trigger (send to readout or fast-clear)
    - The main trigger can 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

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

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

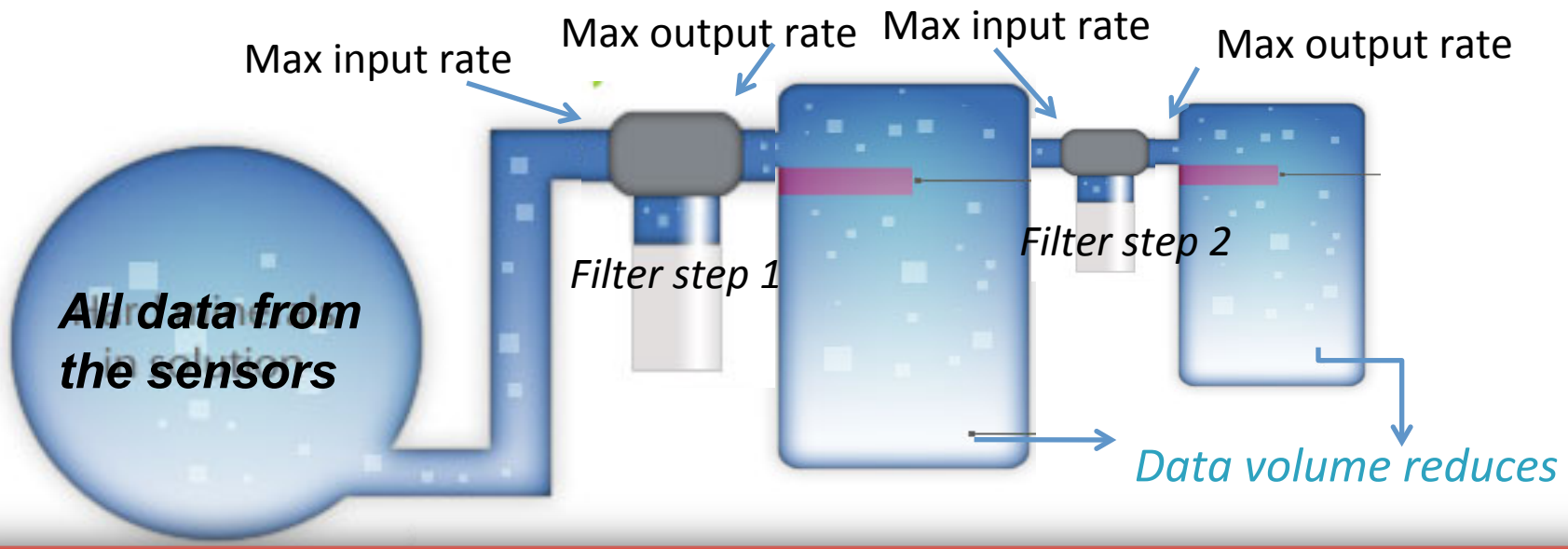
T_{LRO} = Local readout time

Try to minimize each factor!

Buffering and filtering

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

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



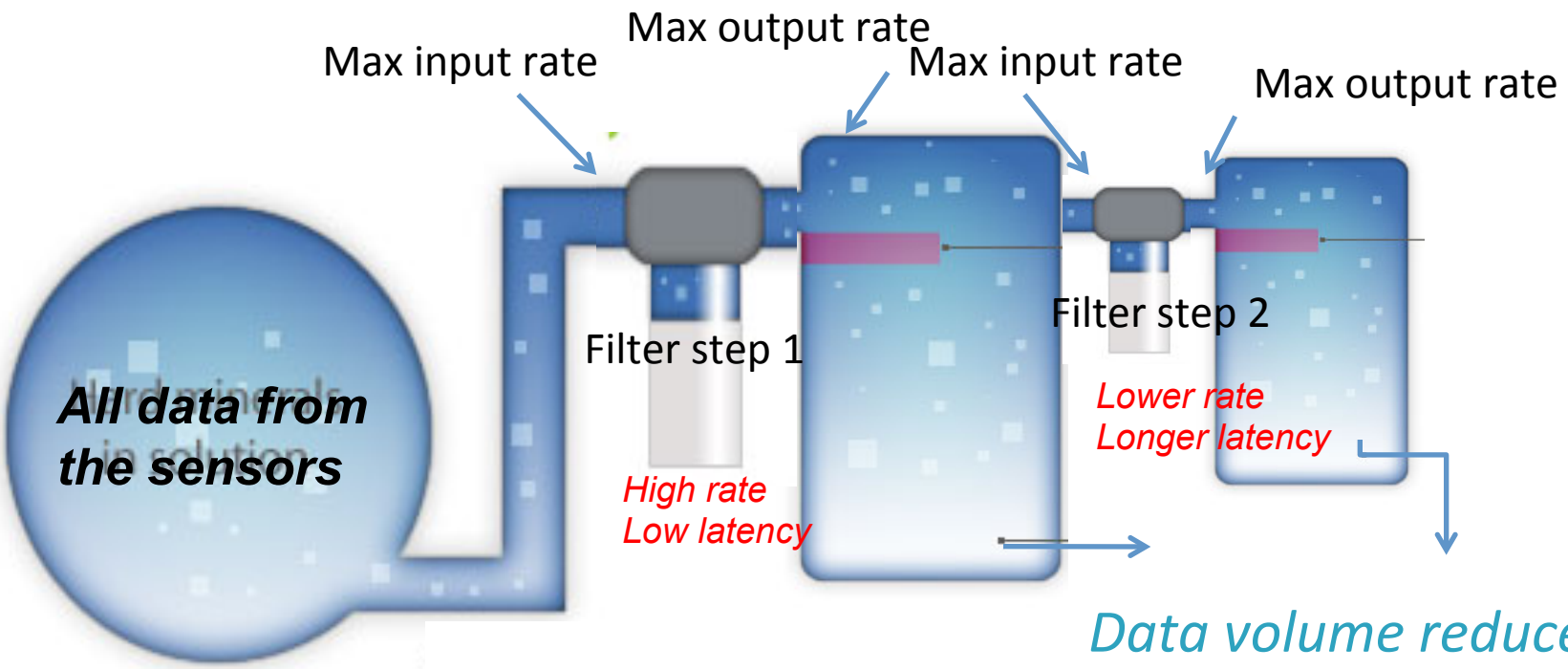
As long as the buffers do not fill up (overflow), no additional dead-time is introduced!

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)

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



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)

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

LHC experiments @ Run1

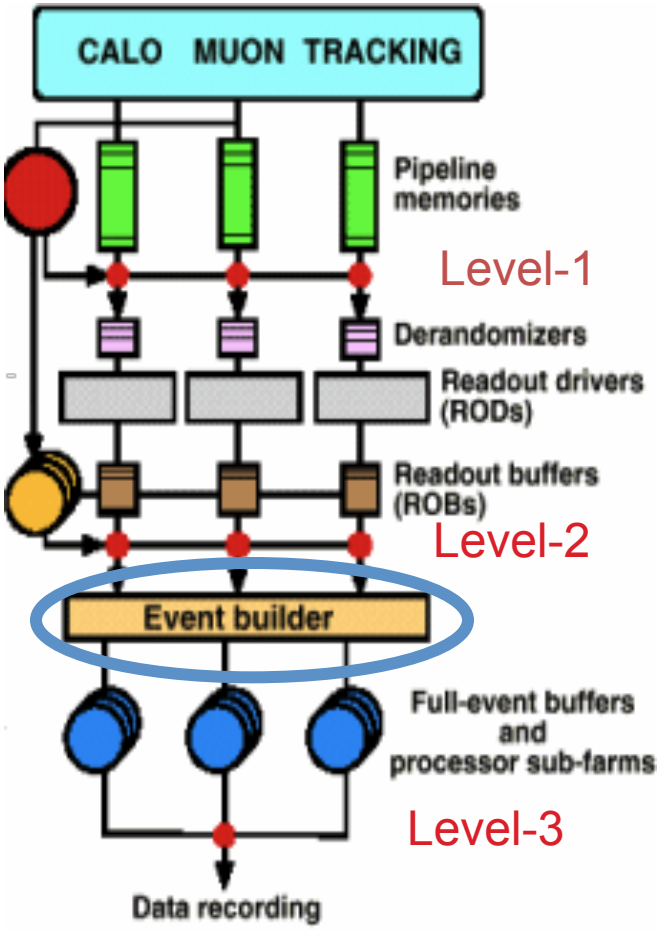
Exp.	N.of Levels
ATLAS	3
CMS	2
LHCb	3
ALICE	4

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

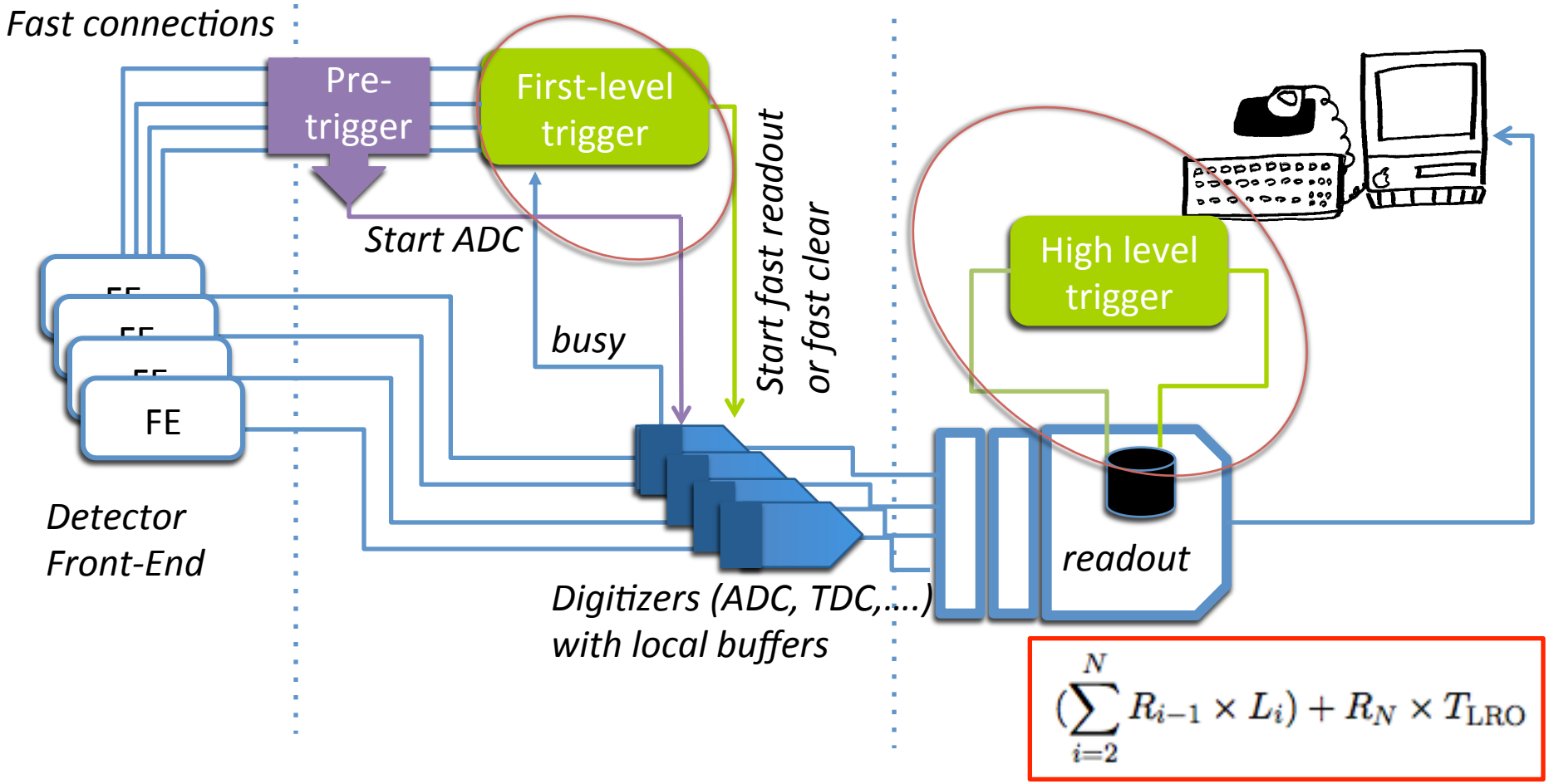
Logical division between levels

- **First-level**: Rapid rejection of high-rate backgrounds
  - **Fast custom electronics**
  - **Coarse granularity** data from detectors
    - Calorimeters for e/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**
    - Full-precision and **full-granularity** information
    - **Fast tracking** in the inner detectors (for example to distinguish e/γ)

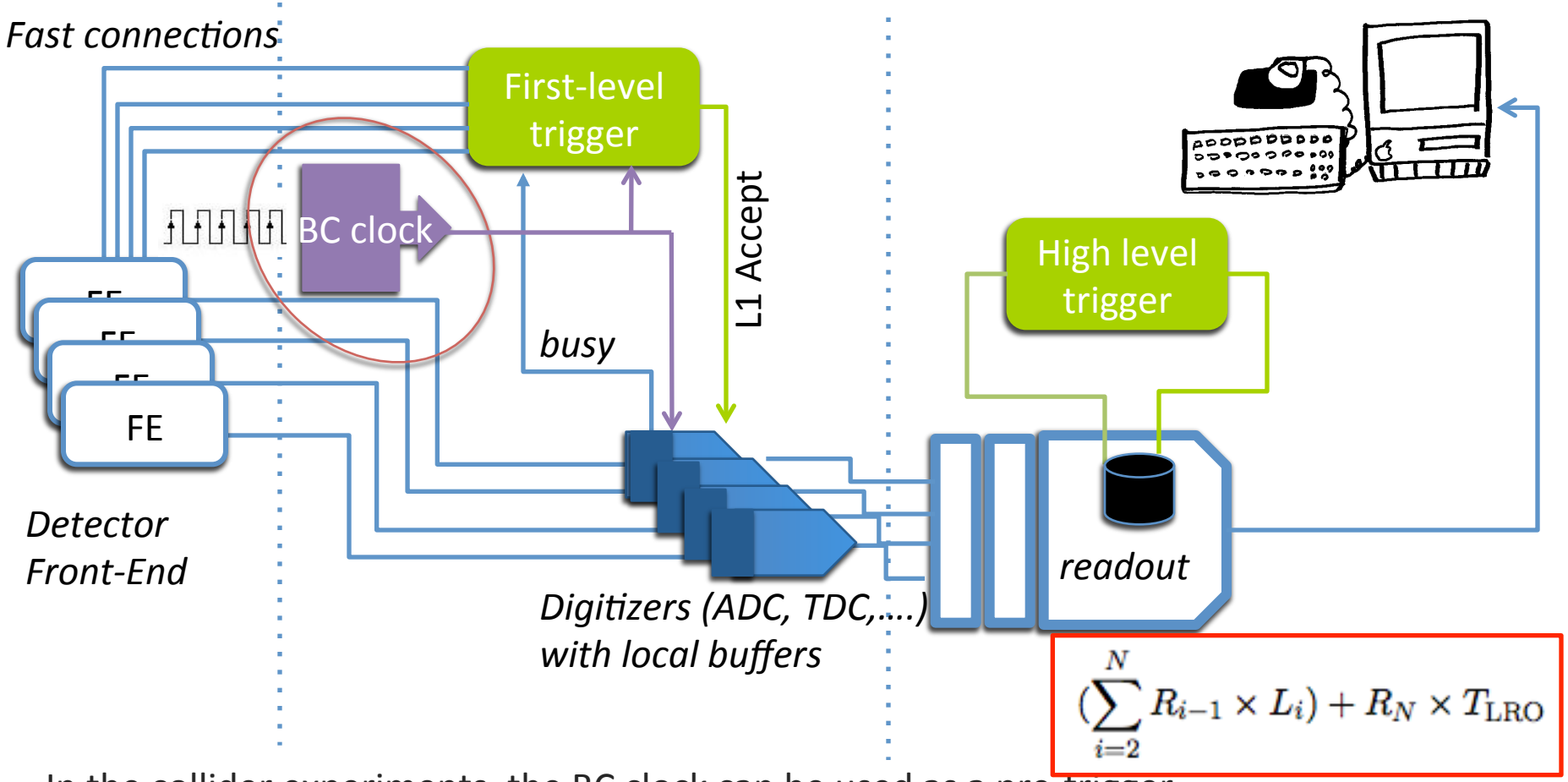


Schema of a multi-level trigger



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

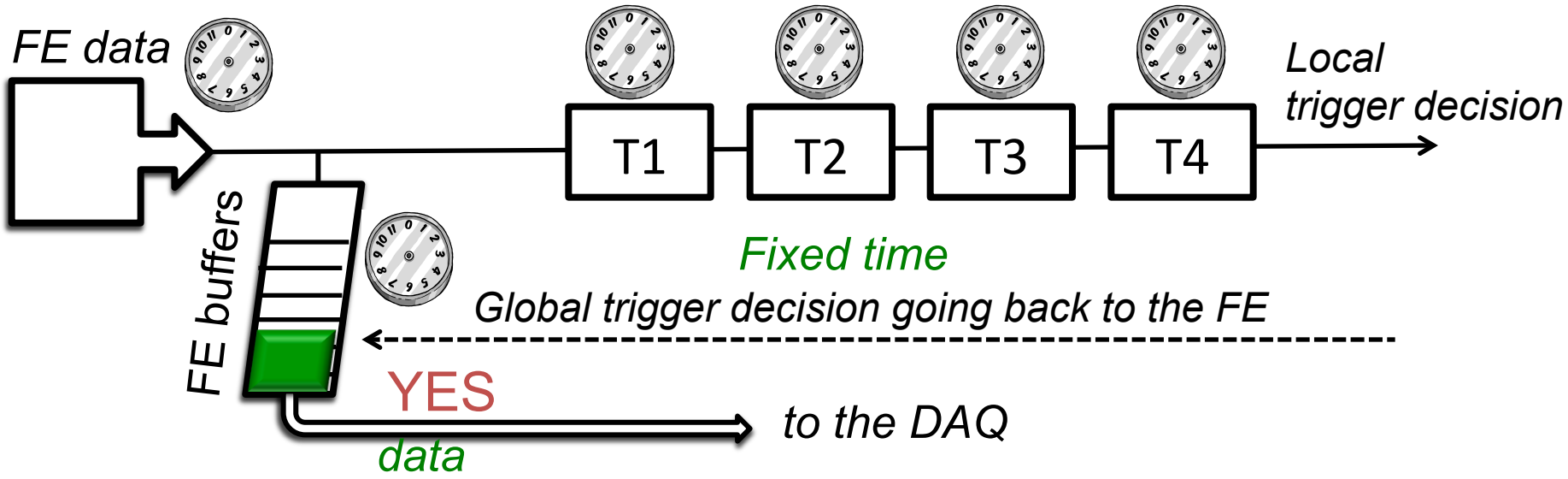
Schema of a multi-level trigger @ colliders



- 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

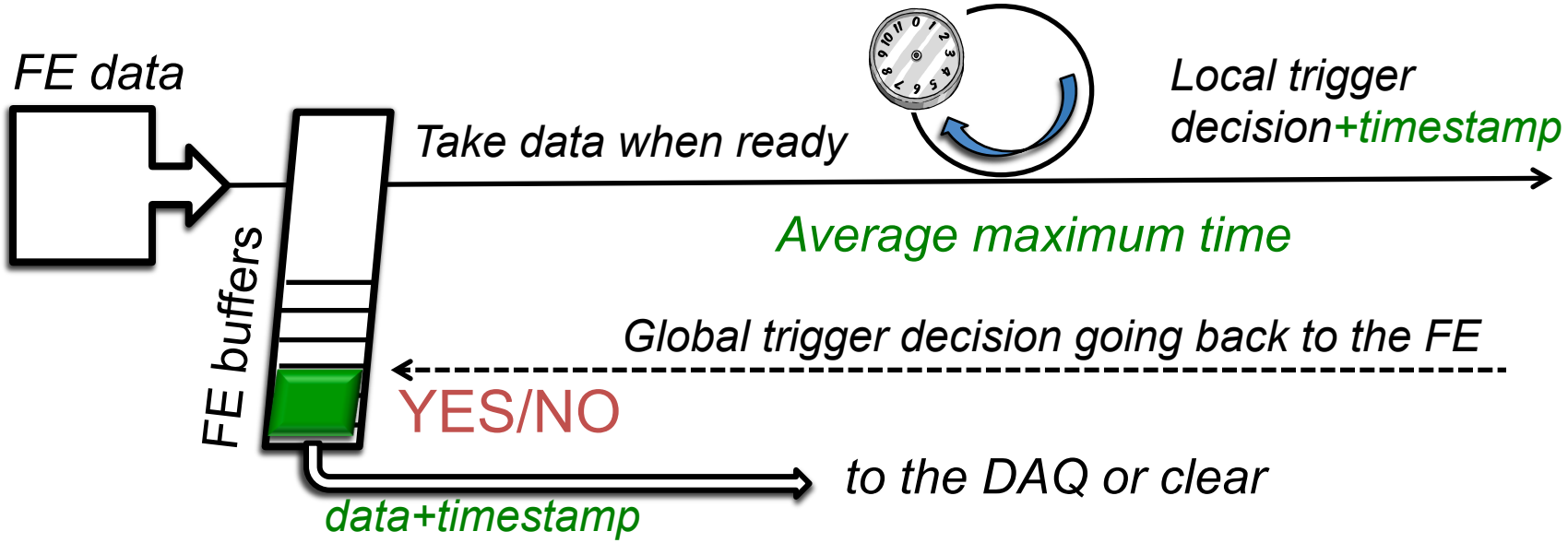
Synchronous or asynchronous?

- **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 ≠ latency → 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 ≠ dead-time → 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 synchronized to the FE to latch the data, needs time-marker stored in the data, data transfer protocol is more complex

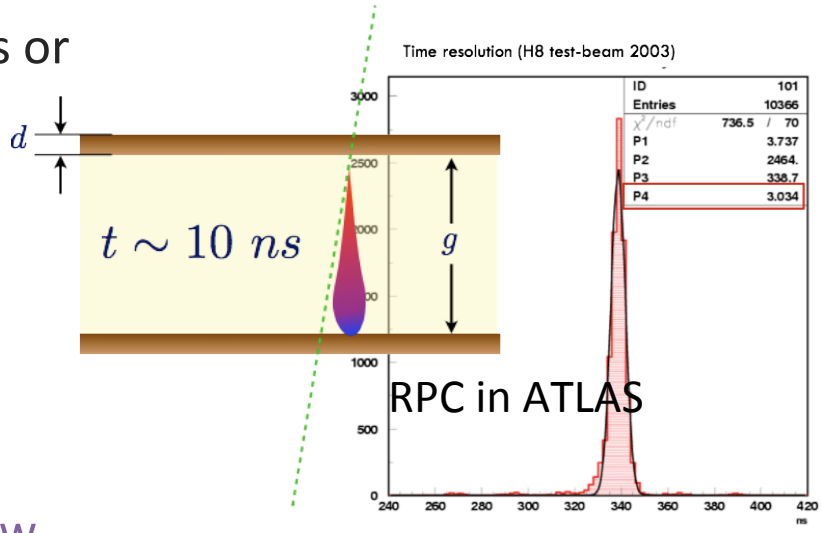


Level-1: reduce the latency

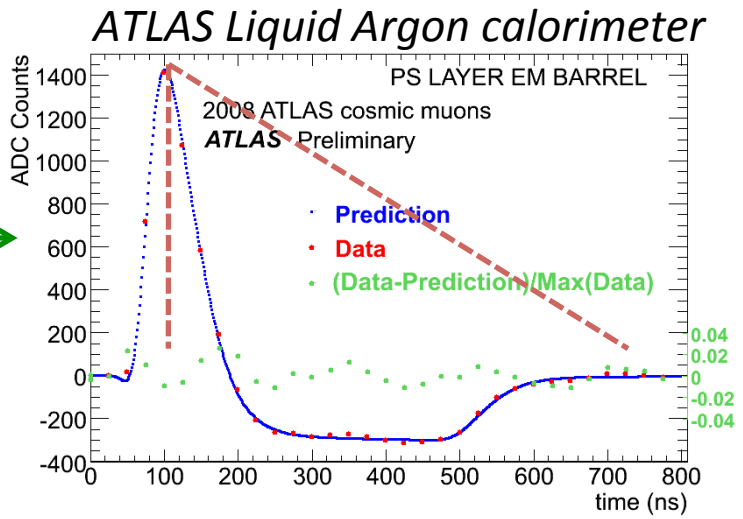
- Pipelined trigger
- Fast processors
- Fast data movement

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**

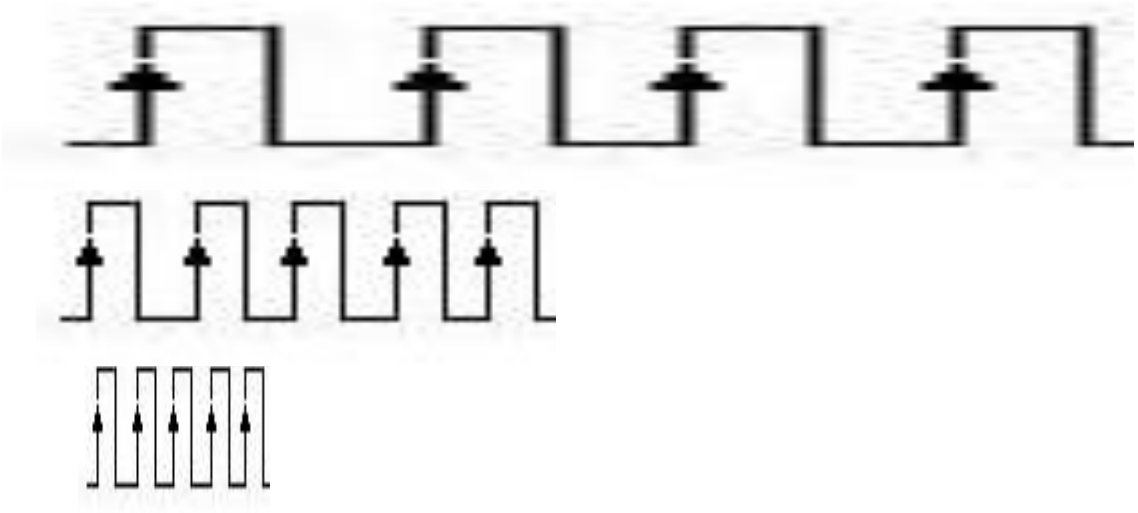
Synch level-1 trigger @ colliders

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

Tevatron: 396 ns

HERA: 96 ns

LHC: 25 ns

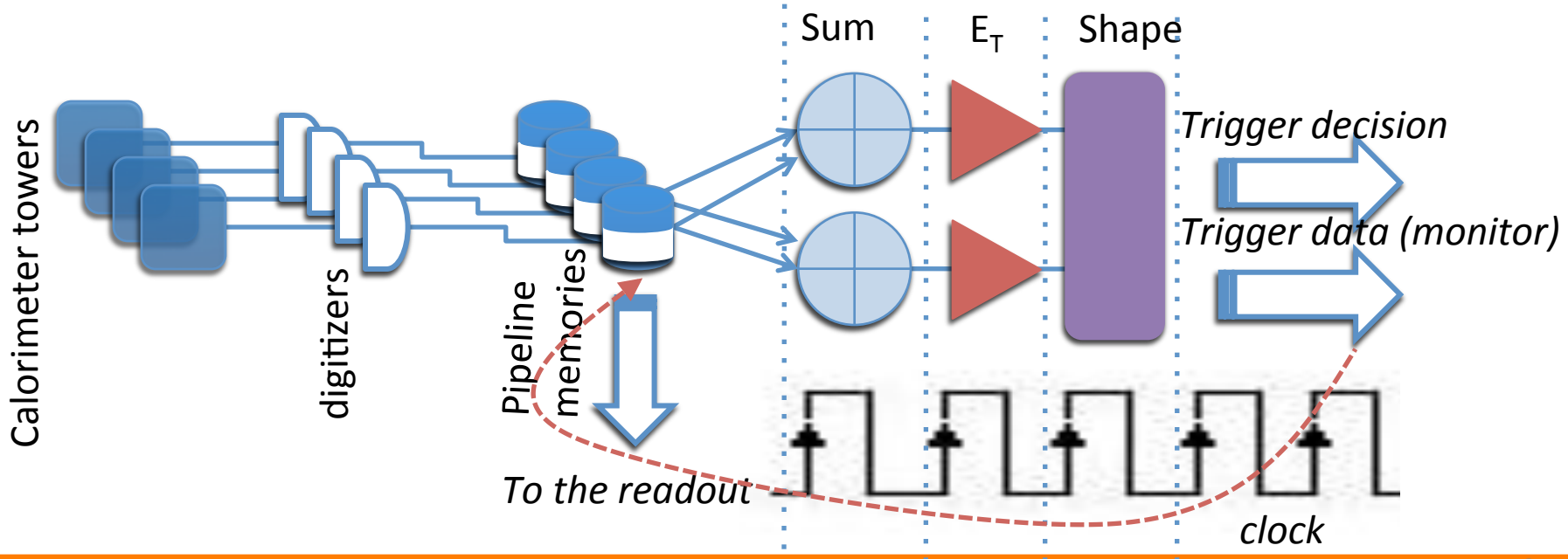


- @LEP, BC interval = **22 μs**: complicated trigger processing within few μs latency was allowed
- In modern colliders: the required high luminosity is driven by high rate of bunch-crossing, then the BC period is short
  - **It's not possible to make a trigger decision within this short time!**

Level-1 pipeline trigger

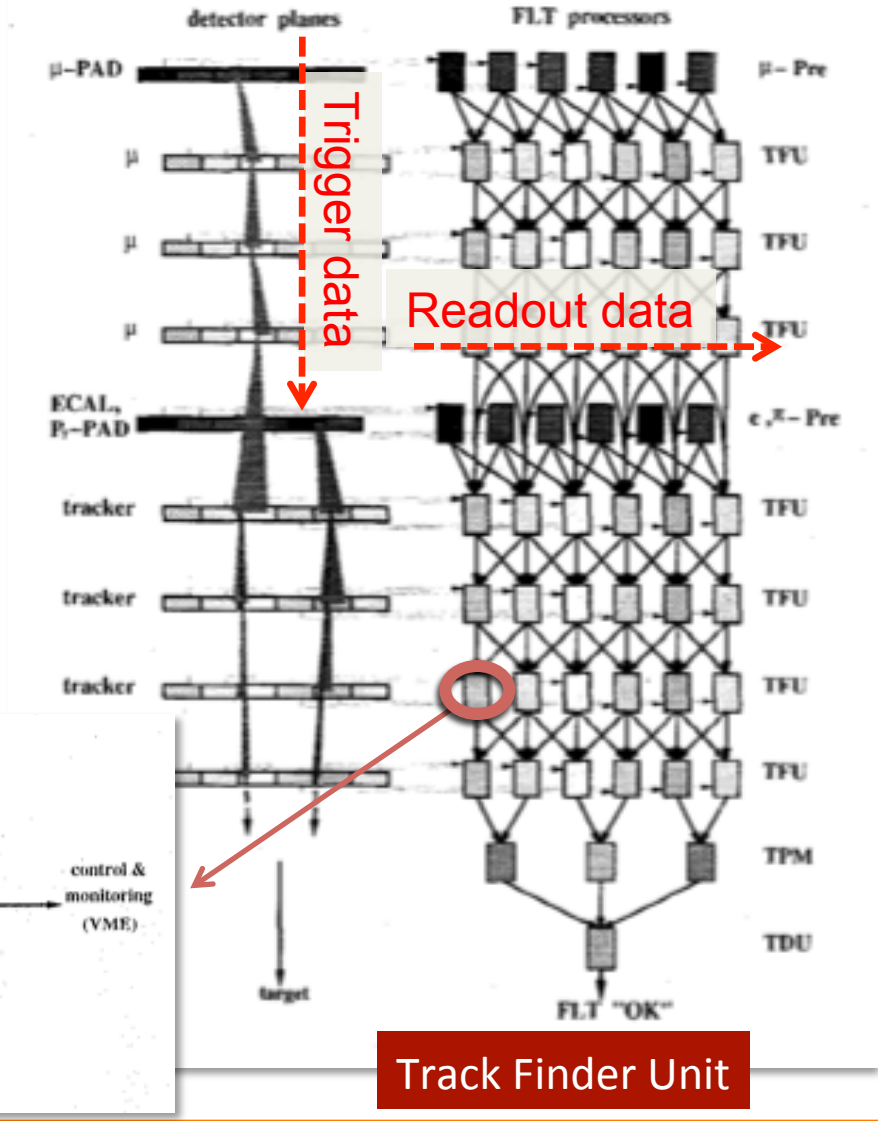
- At each clock tick, a L1 trigger decision must be issued: since data are buffered in pipelines, the decision can be taken later, within **a fixed trigger latency**
  - Latency is given by the sum of the processing time of each step and the data transmission time

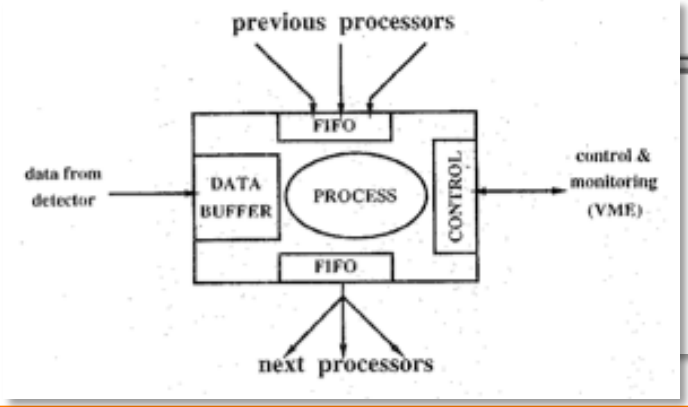
- It's necessary that the trigger **concurrently** processes many events
  - Perform operations in parallel within different processors
  - Divide the processing in steps, each performed within one BC



Example: HERA-B track finder

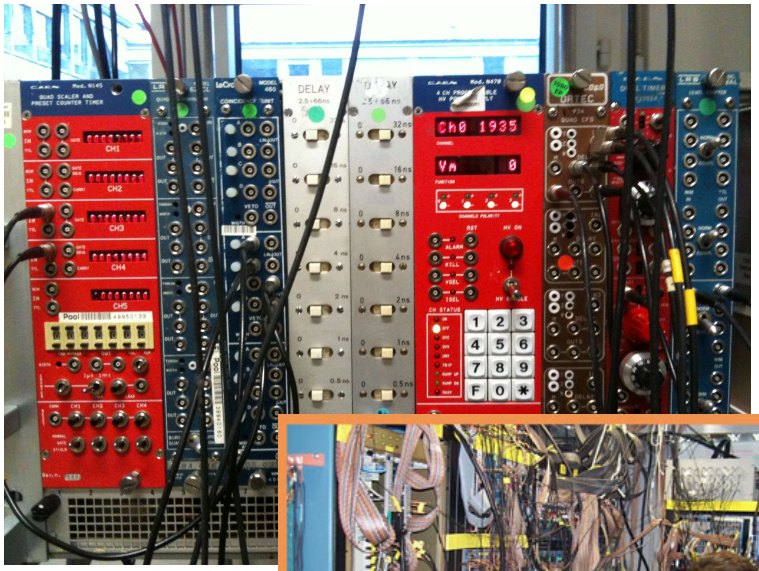
- **Iterative algorithm**: each step processes only a small *Region of Interest* (RoI) defined by the previous step
  - Each unit handles only the hit information corresponding to a small part of the detector
  - Only units whose region is touched by the RoI will process it
- Two data streams:
  - Detector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



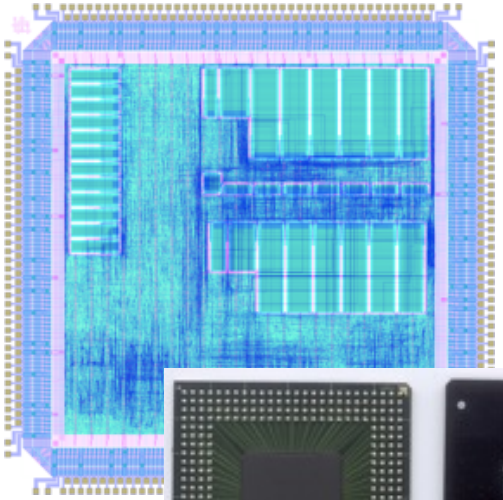


Choose your L1 trigger system

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

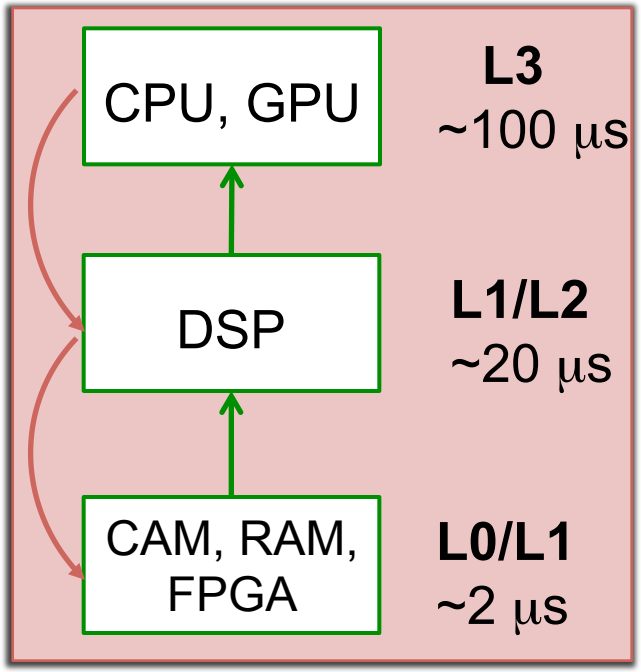


- Digital integrated systems
  - Highly complex algorithms
  - Fast signals processing
  - Specific knowledge of digital systems



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)
  - Offer advantage in terms of reliability, reduced power usage, reduced board size and better performance
  - Can perform the tasks of a computer (CPU, memory, I/O…)

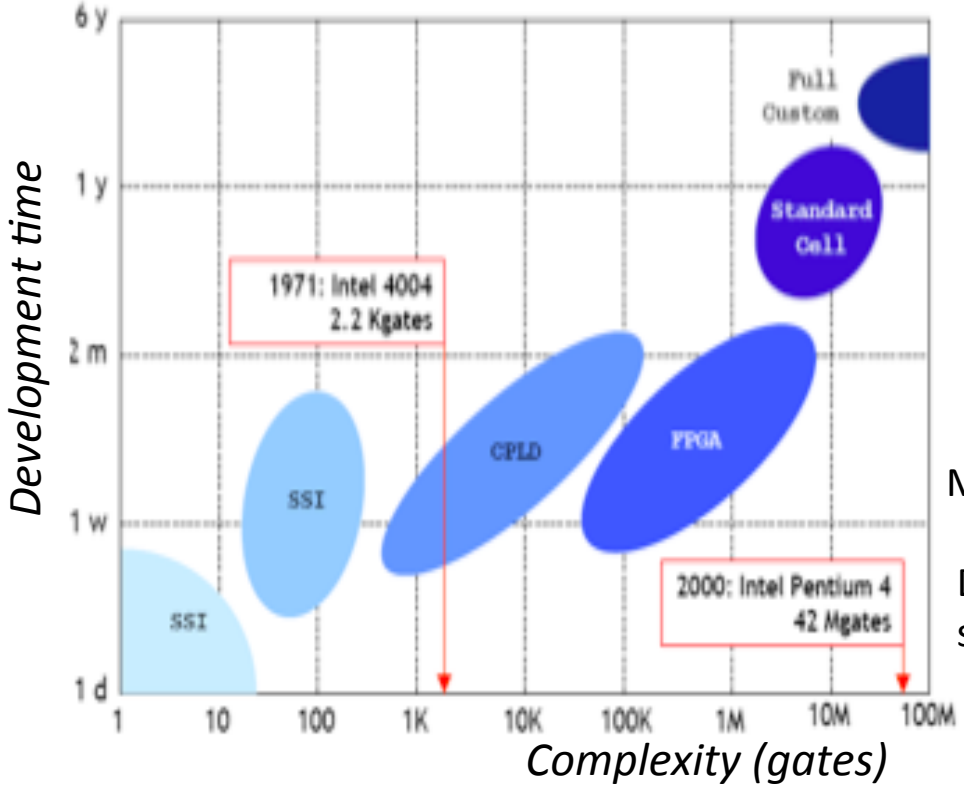
- Different families of IC on the market:
  - Microprocessors (CPUs, DSPs=Digital Signal Processors,..)
    - Available on the market or specific, programmed only once
  - Programmable logic devices (FPGAs, CAMs,…)
    - Can perform more operations per clock cycle, but not suitable for all algorithms (problems with floating points). Other drawbacks are the general difficulties associated with developing the software and the cost

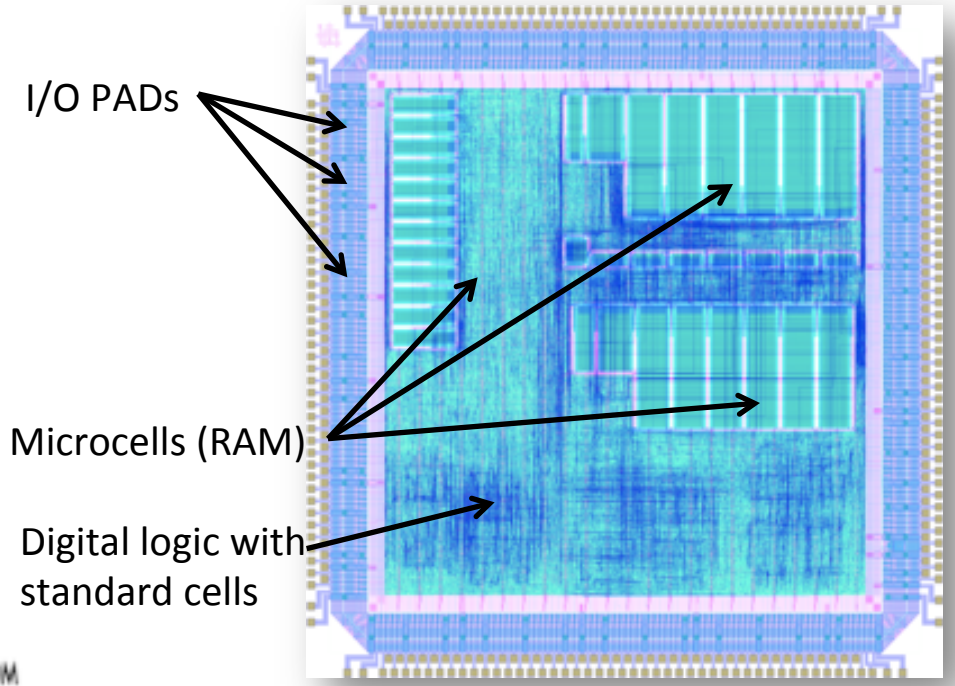
need instructions

already learned the task

Custom trigger processors?

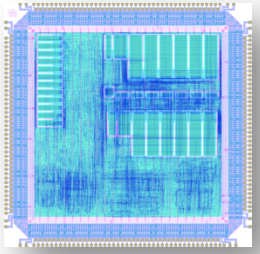
- Application-specific integrated circuits (**ASICs**): optimized for fast processing (Standard Cells, full custom)
  - Intel processors, ~ GHz
- Programmable ASICS (like Field-programmable gate arrays, **FPGAs**)
  - Easily find processors @ 100 MHz on the market (1/10 speed of full custom ASICs)

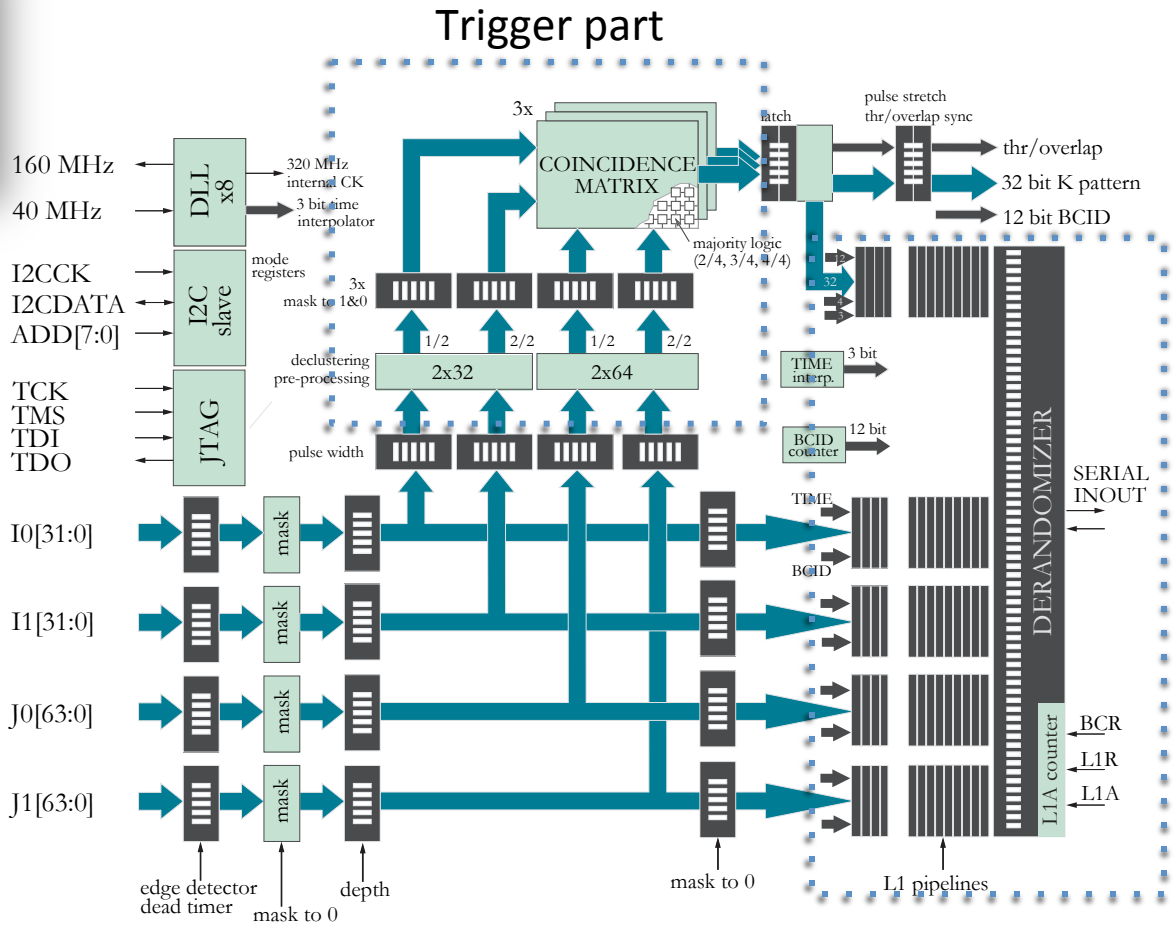




Layout of the CM ASIC

Example: logic of a trigger ASIC





Digital inputs from detector

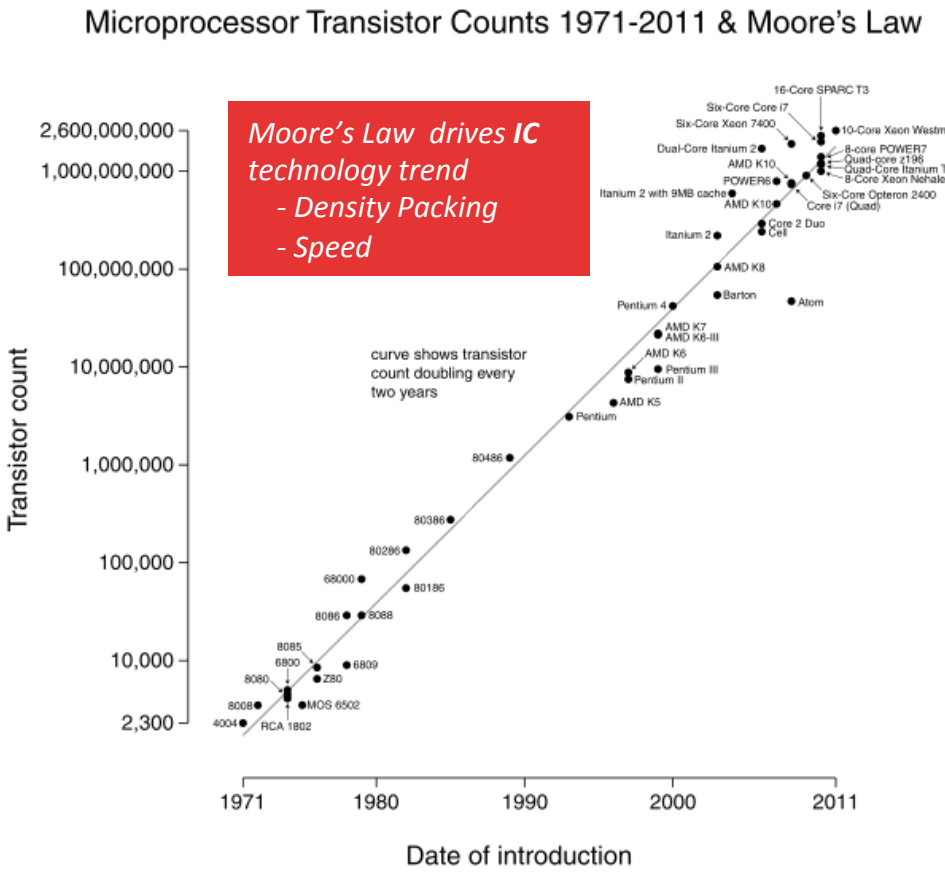
Trigger output

Readout output

Input: synchronization signals

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)

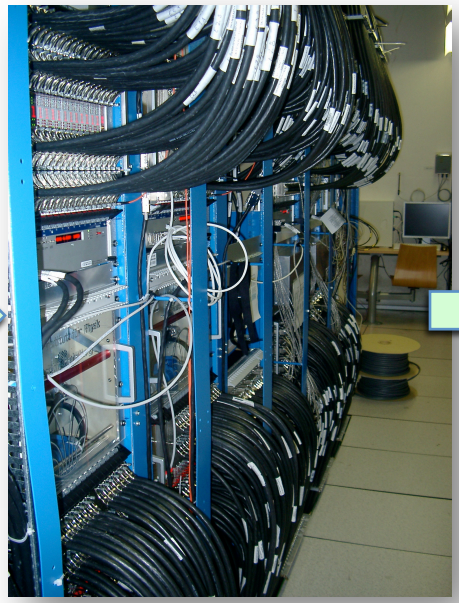
Data movement technologies

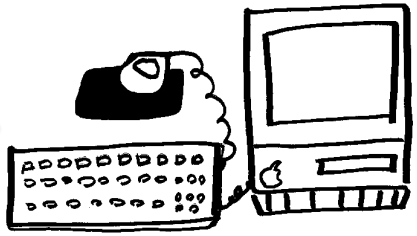
- Faster data processing are placed on-detector (close or joined to the FE)
- Intermediate crates are good separation between FE (long duration) and PCs

On-detector

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

Off-detector



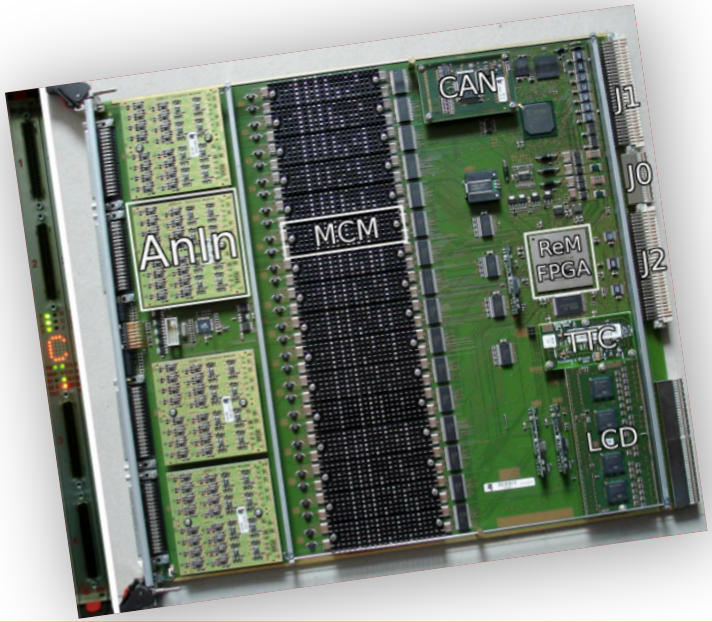


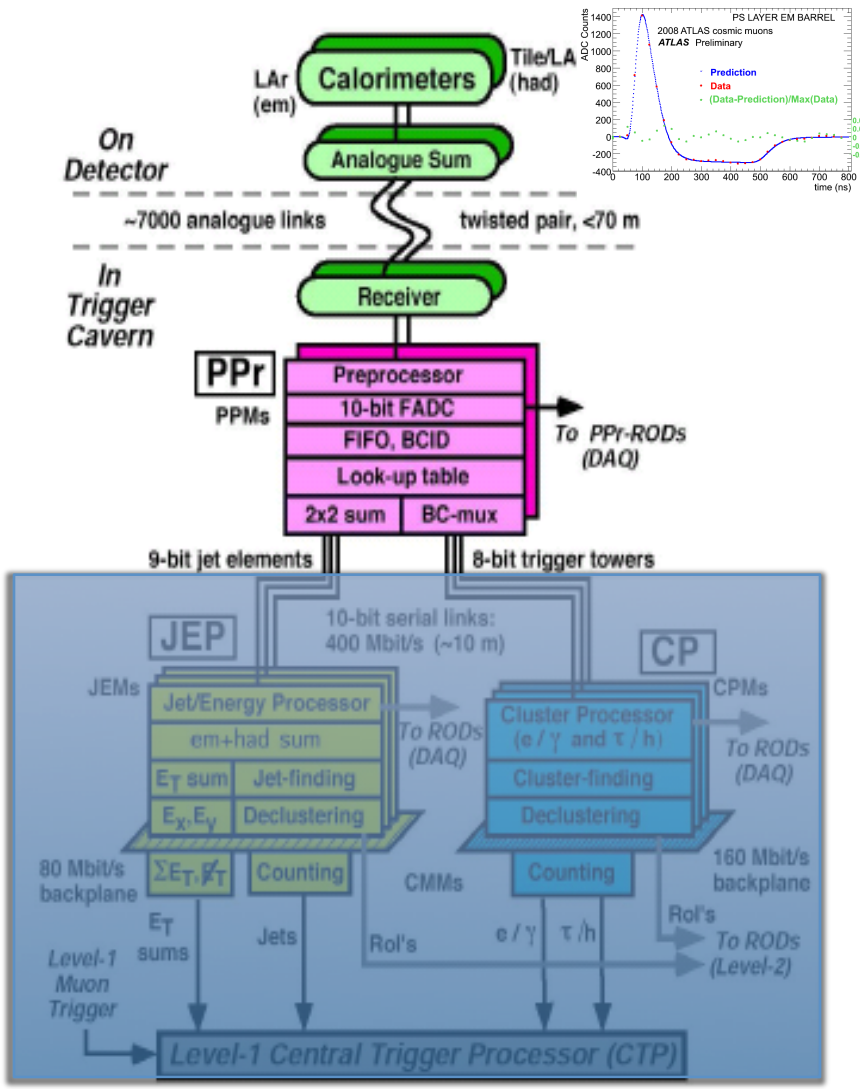
- **High-speed** serial links, electrical and optical, over a variety of distances
  - Low cost and low-power LVDS links, @400 Mbit/s (up to 10 m)
  - Optical GHz-links for longer distances (up to 100 m)

- **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



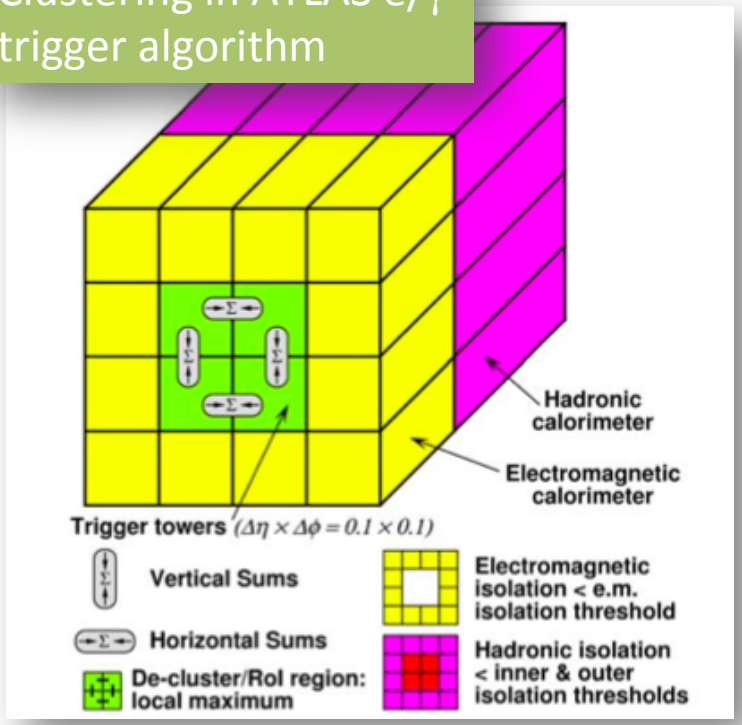


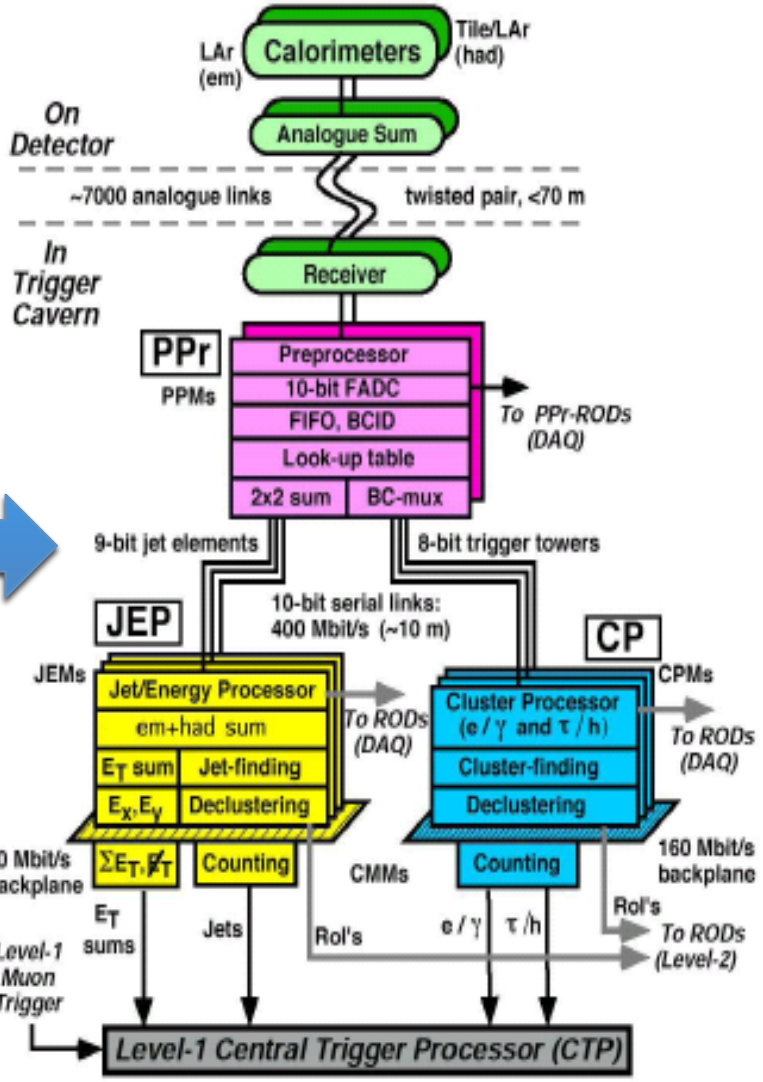
Example: ATLAS calorimeter trigger

- 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

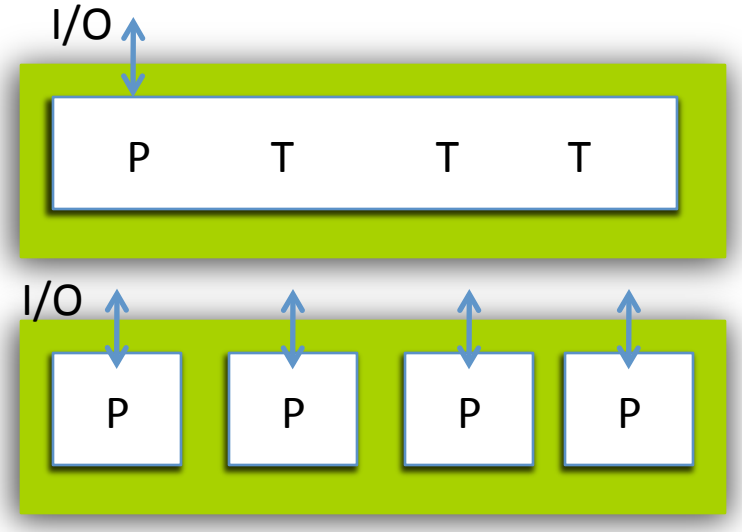




High level triggers

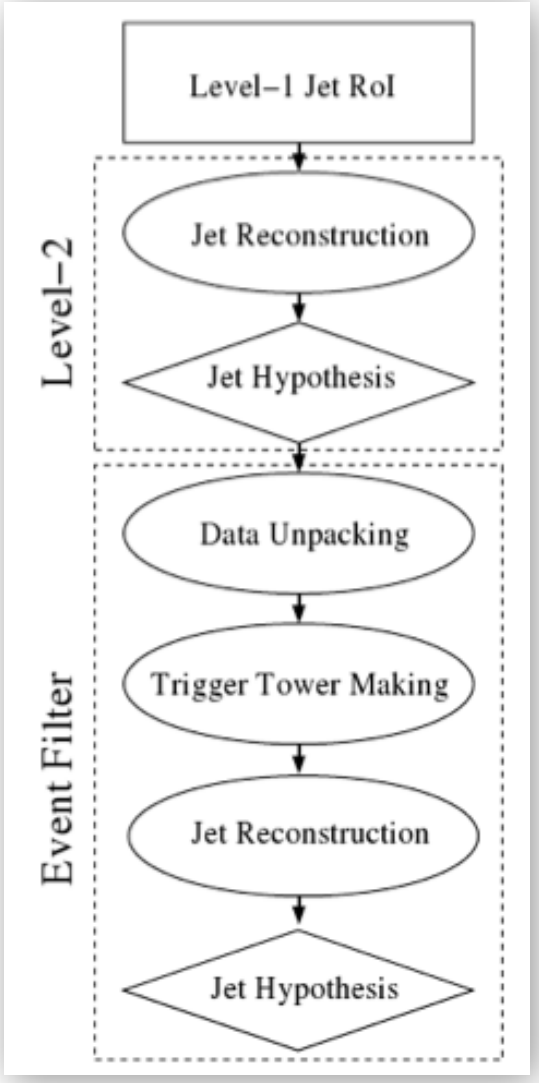
HLT design principles

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



Multi-threading

Multi-processing



High Level Trigger Architecture

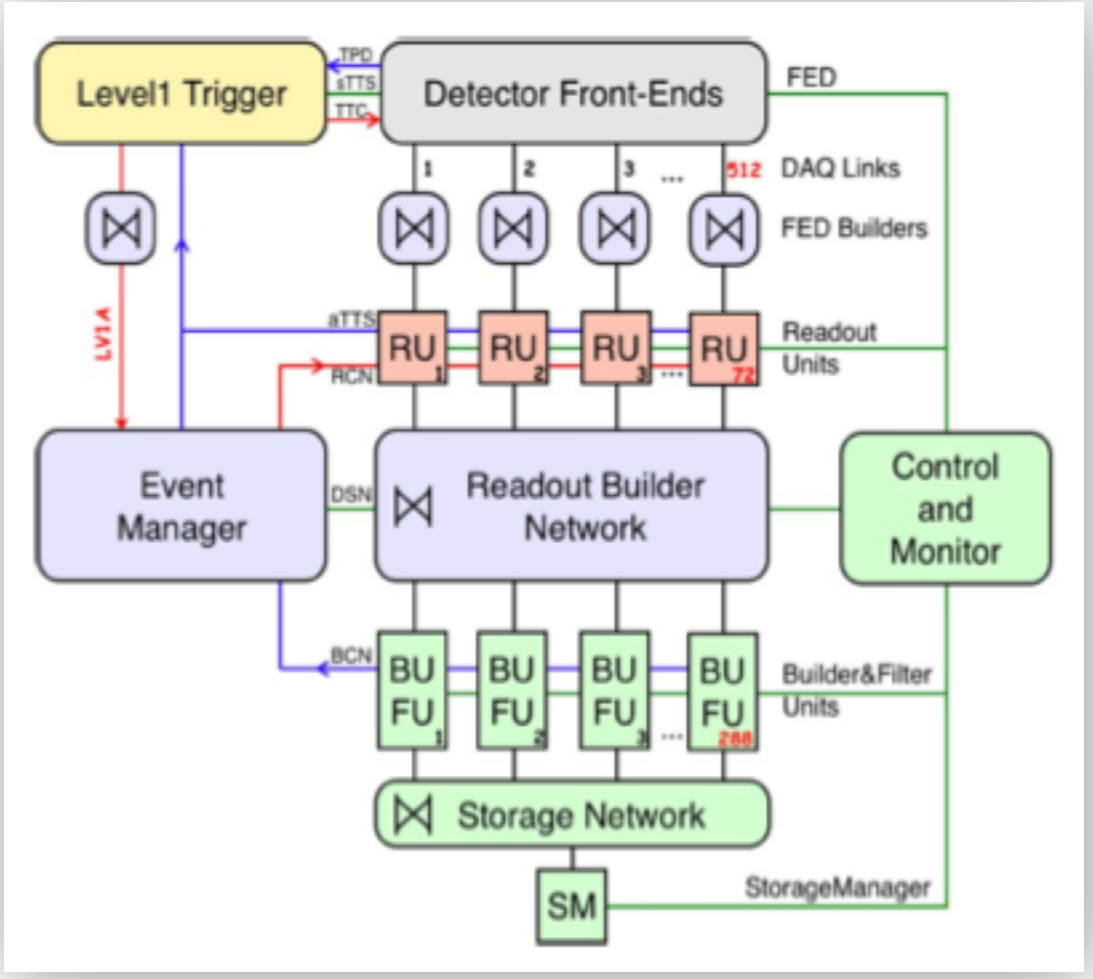
- After the L1 selection, data rates are reduced, but can be still massive
- Key parameter for the design is the **allowed bandwidth after L1**
  - **LEP**: 100 kByte event-size @ few Hz gives **few 100 kByte/s**
    - Supported by 40 Mbyte/s VME bus
  - **ATLAS/CMS**: 1 MByte event-size @100 kHz gives **~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 L2: 10^3	10^6	10	~200
CMS	2	10^5	10^6	100	~200

- **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

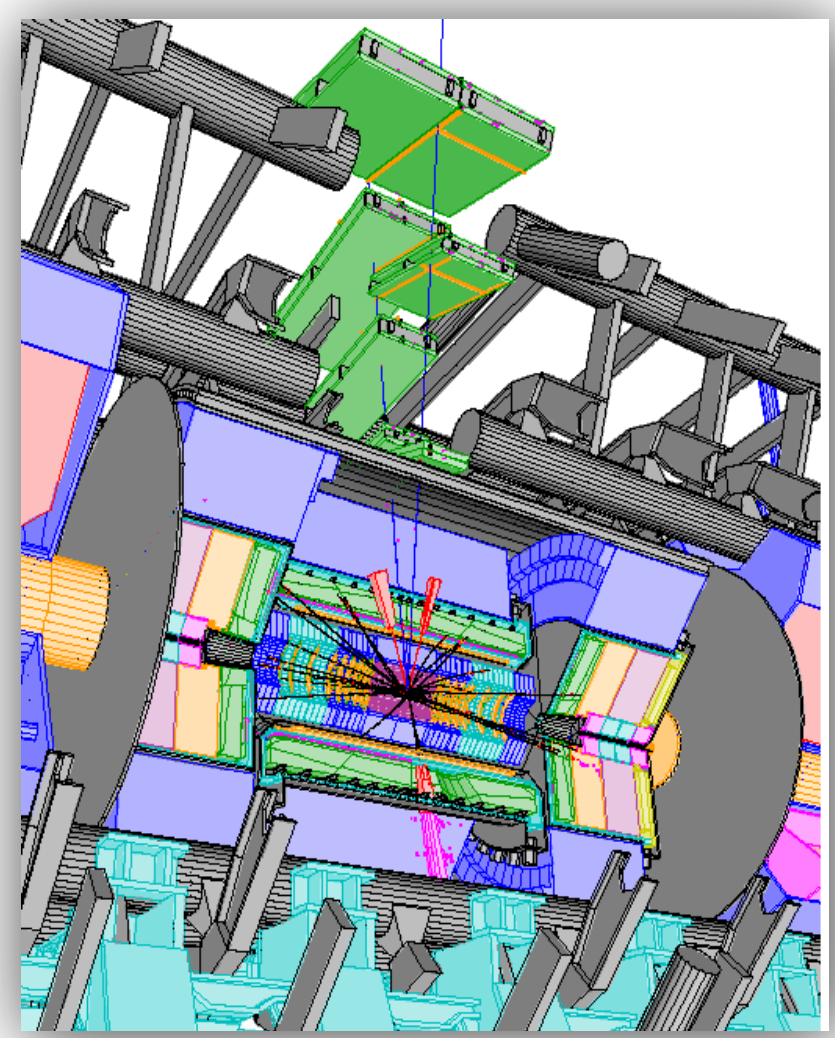
- Data from the readout system (RU) are transferred to the filters (FU) through a builder network
- Each filter unit processes only a fraction of the events
- Event-building is factorized into a **number of slices**, each processing only 1/nth of the events
  - Large total bandwidth still required
 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



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

Seeded reconstruction HLT: ATLAS

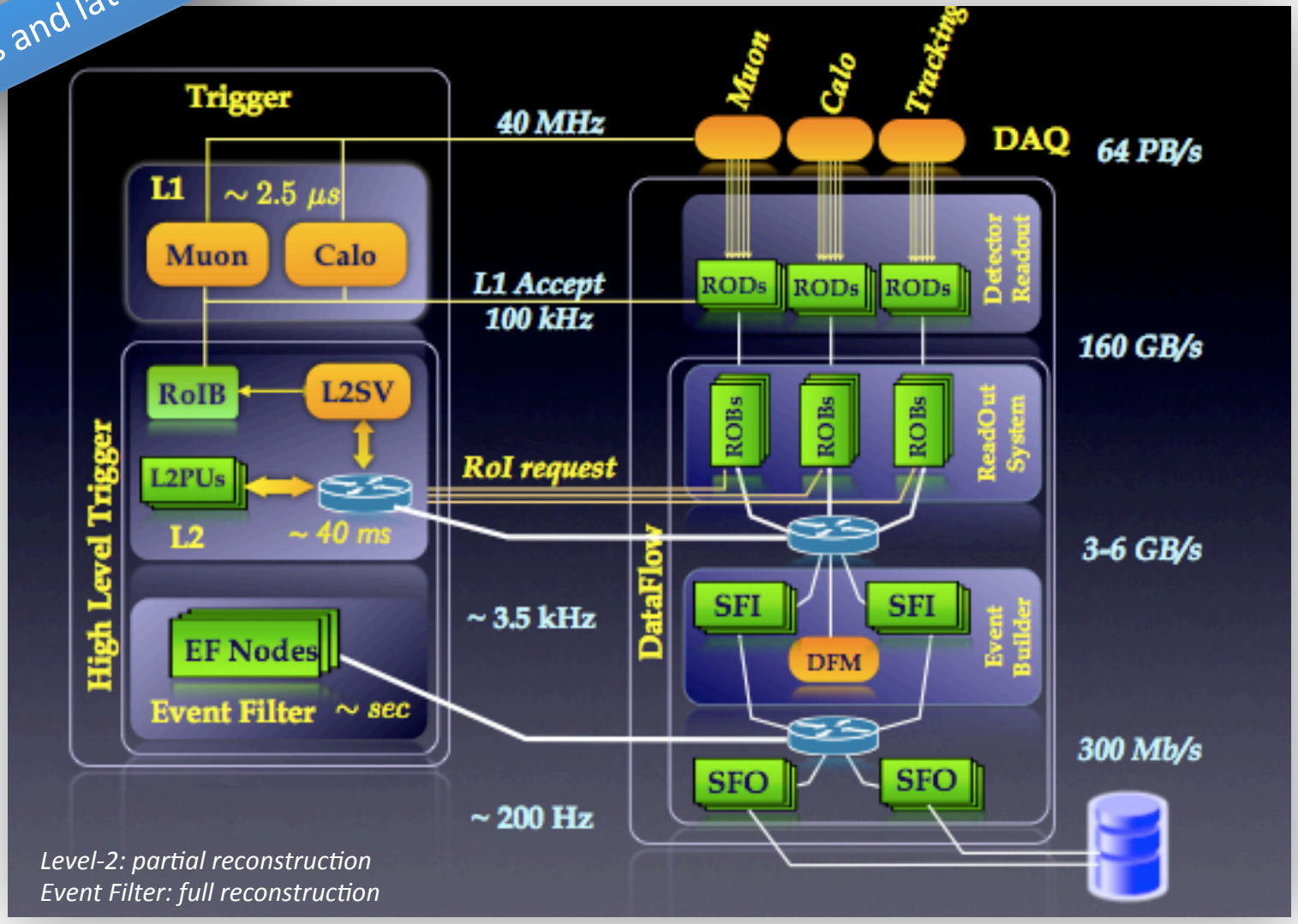
- Level-2 uses the information seeded by level-1 trigger
  - Only the data coming from the region indicated by the level-1 is processed, called **Region-of-Interest (RoI)**
  - The resulting total amount of RoI data is minimal: a few % of the Level-1 throughput
  - Level-2 can use the full granularity 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



Level-2: partial reconstruction
Event Filter: full reconstruction

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!!