



# Trigger and Data Acquisition

Summer Student Lectures  
15, 16 & 18 July 2002

Clara Gaspar CERN/EP-LBC



# Contents of the Lectures

## ⌘ Overview

- ☒ General framework
- ☒ Rates and data at collider experiments

## ⌘ Basic concepts

- ☒ Multi-level trigger systems and readout structures

## ⌘ Front-end essentials

- ☒ Digitizers
- ☒ Signal processing

## ⌘ Trigger algorithms and implementations

- ☒ Trigger design and performance
- ☒ Fast hardware trigger (Level-1)
- ☒ Software triggers (Level-2, Level-3, etc.)

← Break

## ⌘ Data Acquisition

- ☒ Readout networks (buses, switches, etc.)
- ☒ Event building and event filters
- ☒ Data storage

← Break

## ⌘ Configuration, Control and Monitoring

- ☒ Operating modes
- ☒ Run Control
- ☒ Data monitoring and quality control

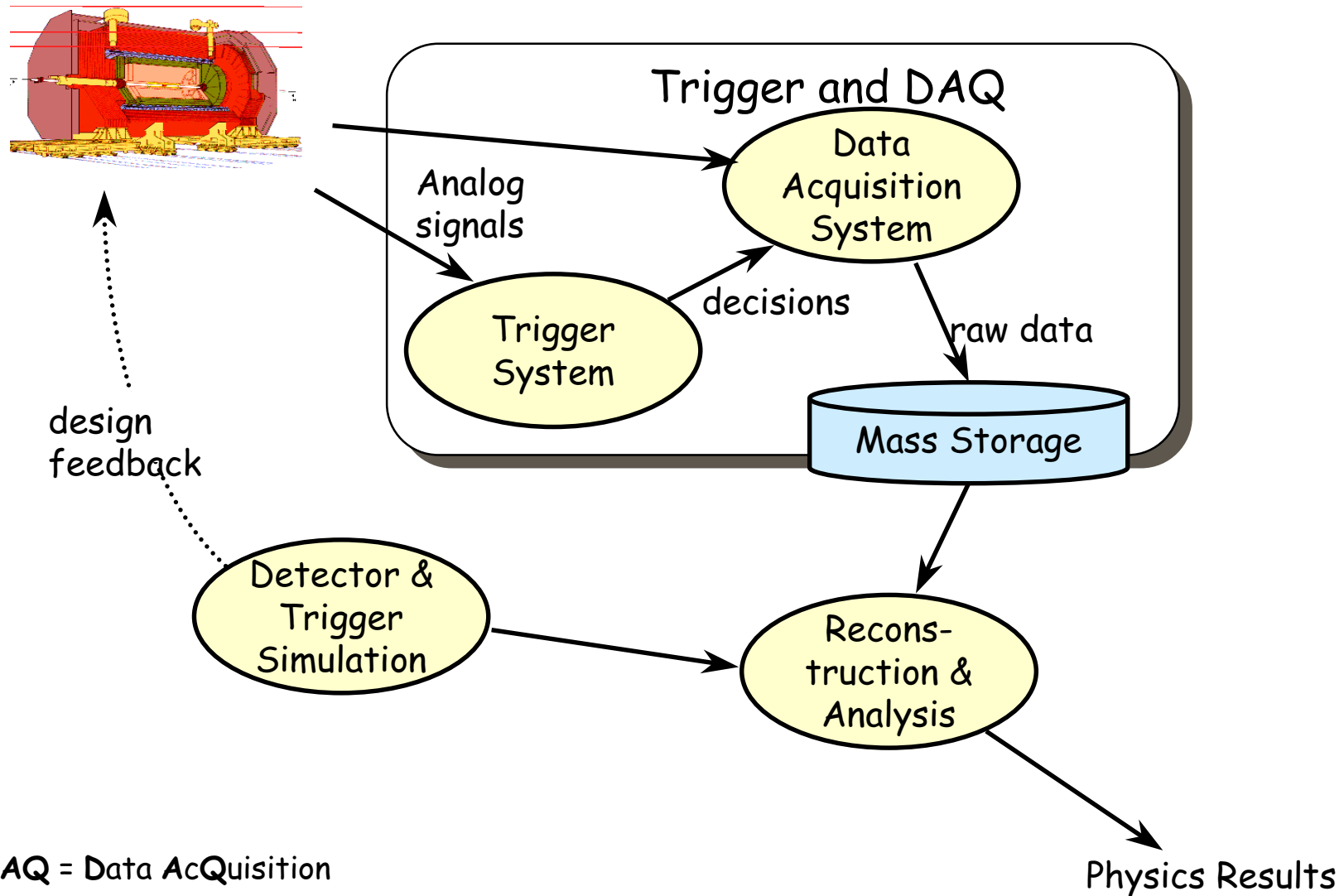


# Foreword

- ⌘ This course presents the experience acquired by **the work of many people** and many teams which have been developing and building trigger and data acquisition systems for high energy physics experiments for many years.
- ⌘ The subject of this course is **not an exact science**. The trigger and data acquisition system of each experiment is somehow unique (different experiments have different requirements and even similar requirements may be fulfilled with different solutions).
- ⌘ This is not a review of existing systems. The reference to existing or proposed systems are used as examples.
- ⌘ To prepare this course I have taken material and presentation ideas from my predecessors in previous years:  
P. Mato and Ph. Charpentier.



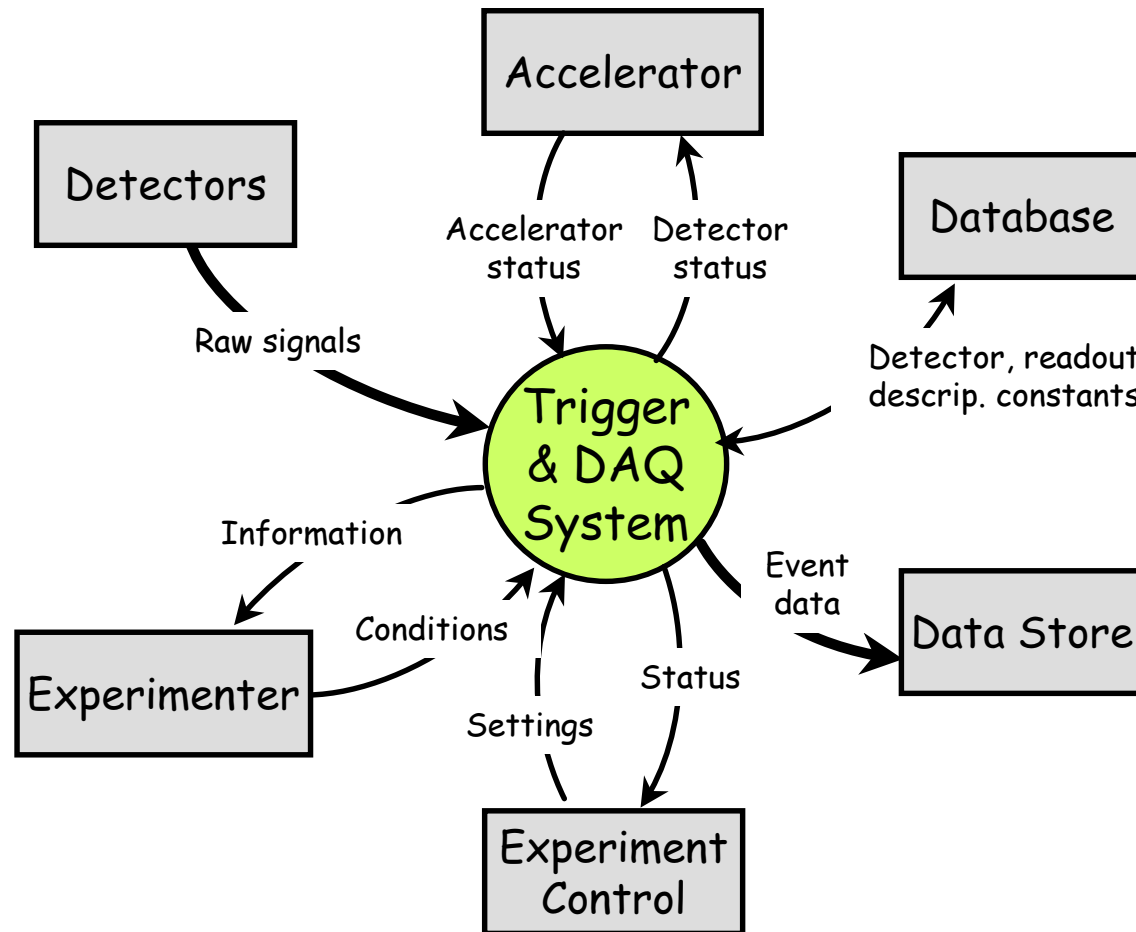
# General Framework





# Context Diagram

- ⌘ The main role of T & DAQ is to process the signals generated in the detector and write the information onto data storage, but:





# Trigger & DAQ

## ⌘ Trigger System:

- ⊞ Selects in Real Time "interesting" events from the bulk of collisions. - Decides if YES or NO the event should be read out of the detector and stored

## ⌘ Data Acquisition System

- ⊞ Gathers the data produced by the detector and stores it (for positive trigger decisions)

- ⊞ Front End Electronics:

- Receive detector, trigger and timing signals and produce digitized information

- ⊞ Readout Network

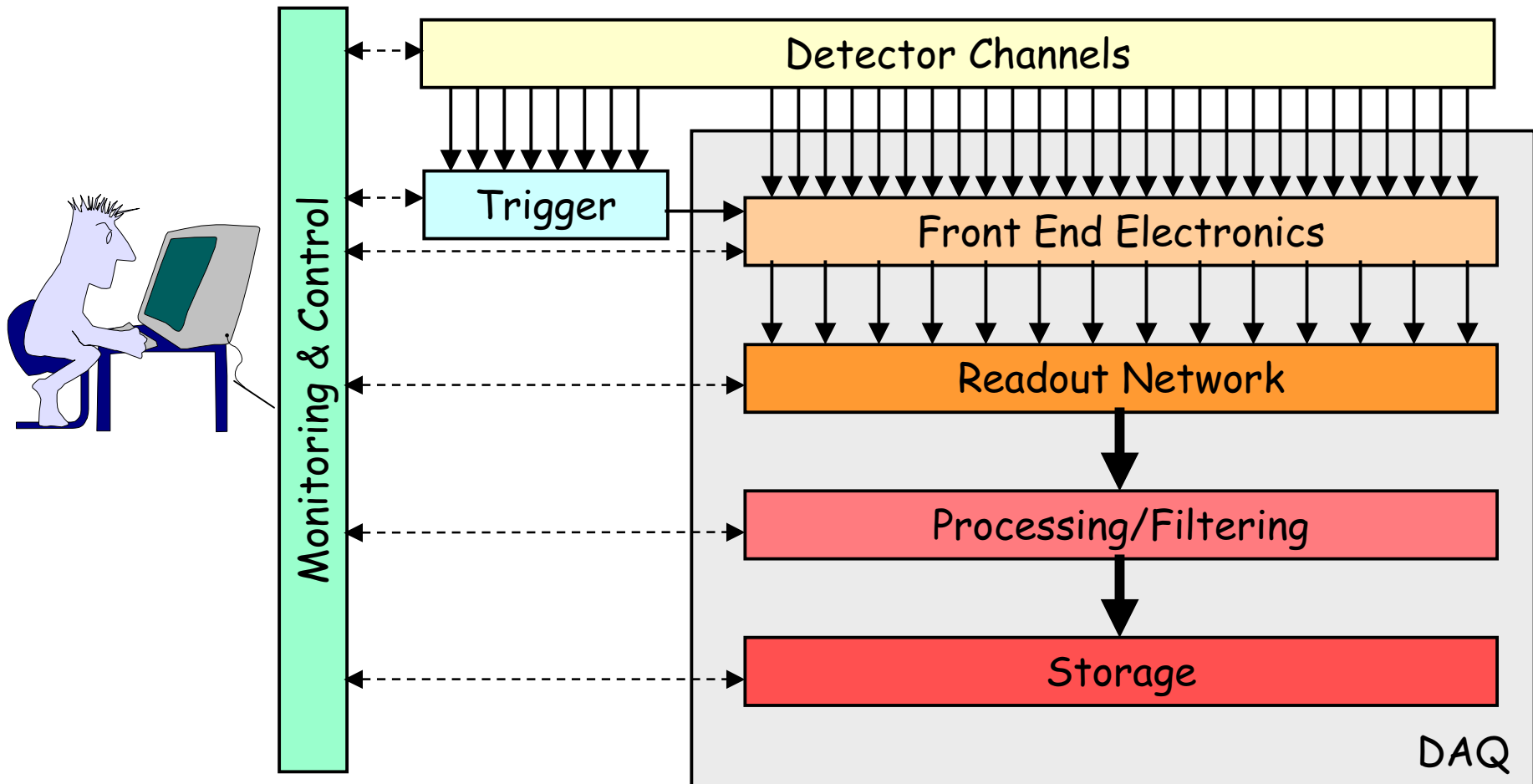
- Reads front end data and forms complete events (sometimes in stages) - Event building

- ⊞ Central DAQ

- Stores event data, can do data processing or filtering
- Overall Configuration Control and Monitoring



# Trigger, DAQ & Control





# LEP & LHC in Numbers

	LEP (1989/2000)	LHC (2007)	Factor
Nr. Electronic Channels	$\approx 100\ 000$	$\approx 10\ 000\ 000$	$\times 10^2$
Raw data rate	$\approx 100\ \text{GB/s}$	$\approx 1\ 000\ \text{TB/s}$	$\times 10^4$
Data rate on Tape	$\approx 1\ \text{MB/s}$	$\approx 100\ \text{MB/s}$	$\times 10^2$
Event size	$\approx 100\ \text{KB}$	$\approx 1\ \text{MB}$	$\times 10$
Bunch Separation	$22\ \mu\text{s}$	$25\ \text{ns}$	$\times 10^3$
Bunch Crossing Rate	$45\ \text{KHz}$	$40\ \text{MHz}$	$\times 10^3$
Rate on Tape	$10\ \text{Hz}$	$100\ \text{Hz}$	$\times 10$
Analysis	$0.1\ \text{Hz}$	$10^{-6}\ \text{Hz}$	$\times 10^5$
	( $Z_0, W$ )	(Higgs)	



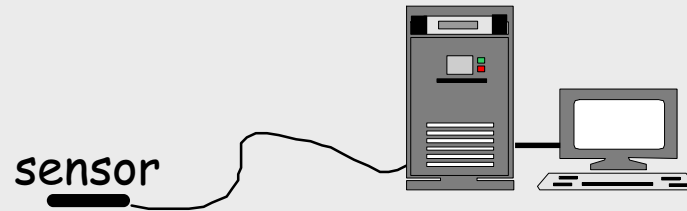


# Basic Concepts

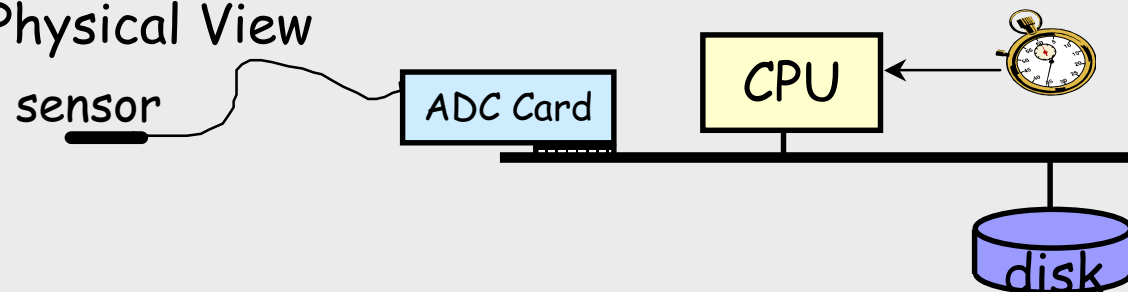


# Trivial DAQ

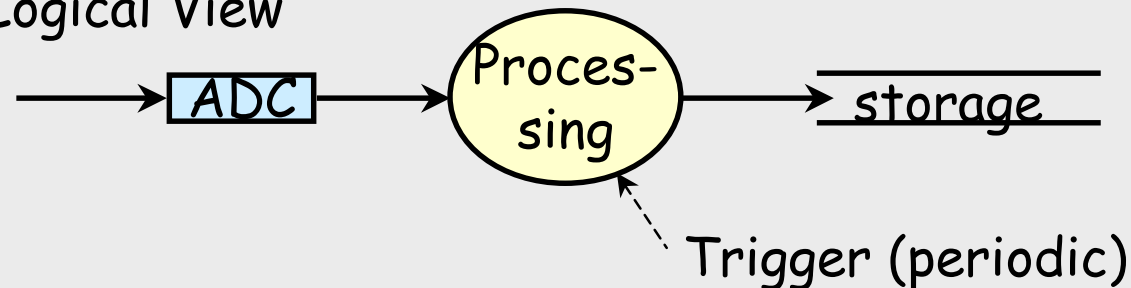
External View



Physical View

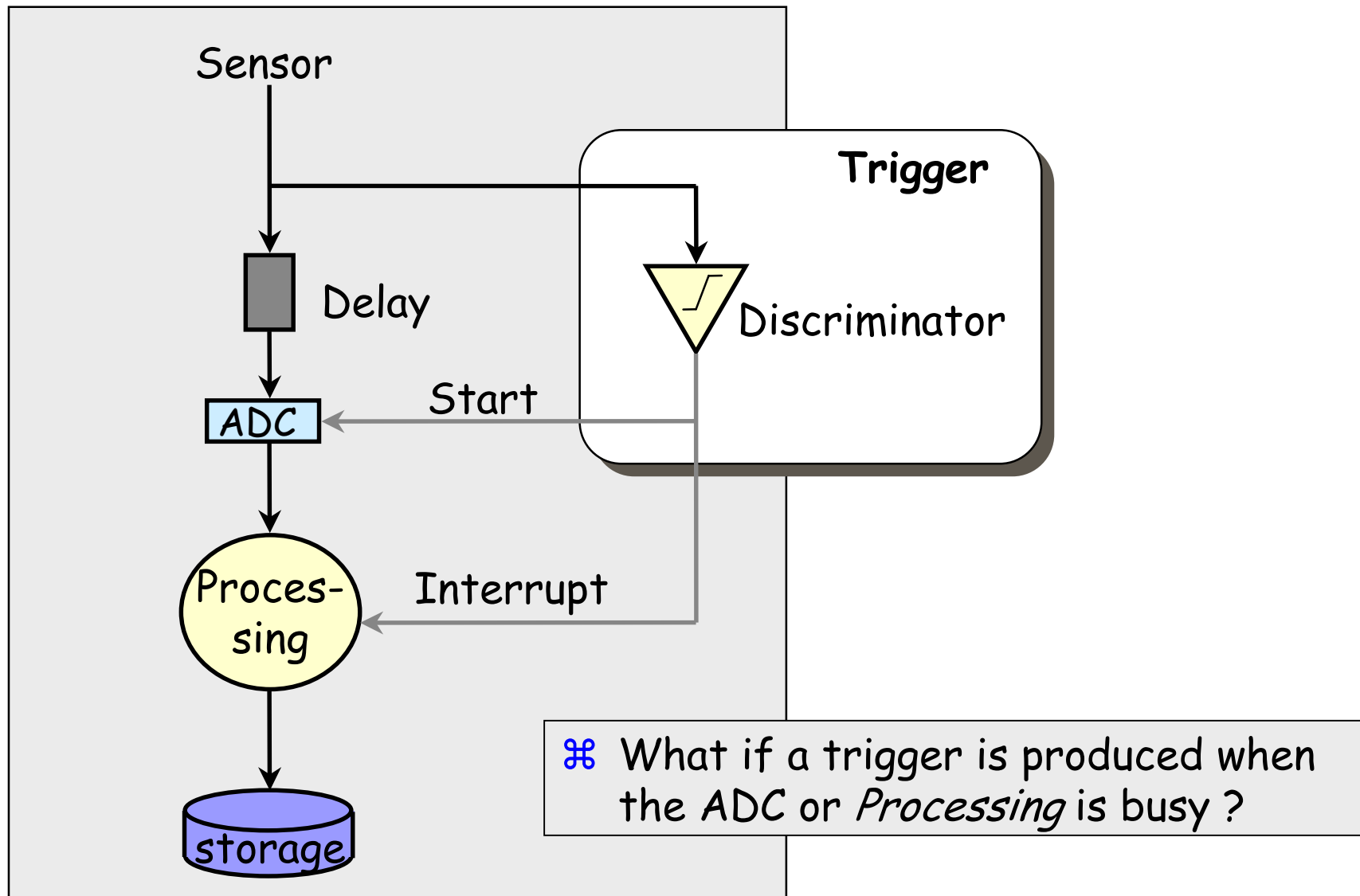


Logical View



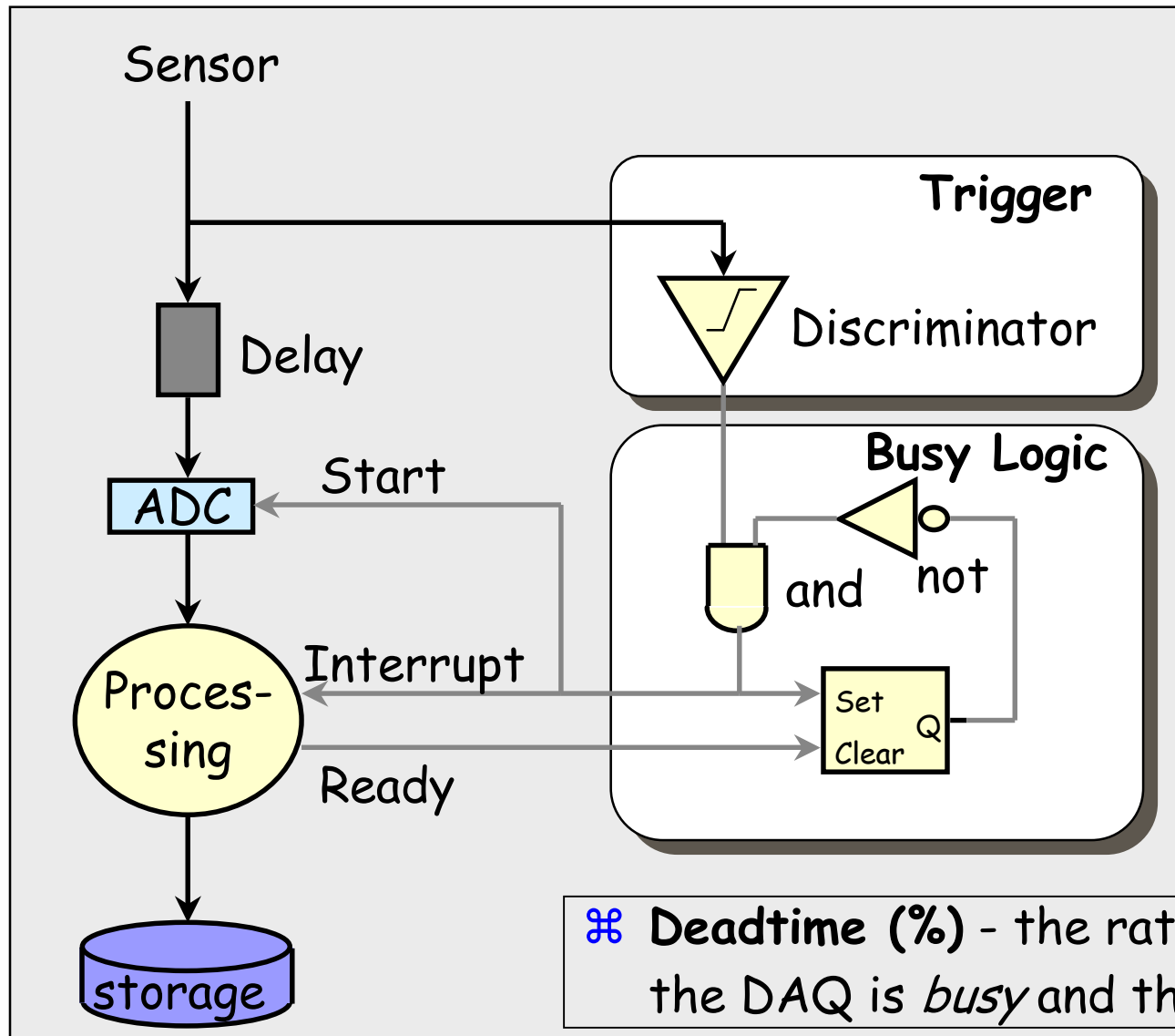


# Trivial DAQ with a real trigger





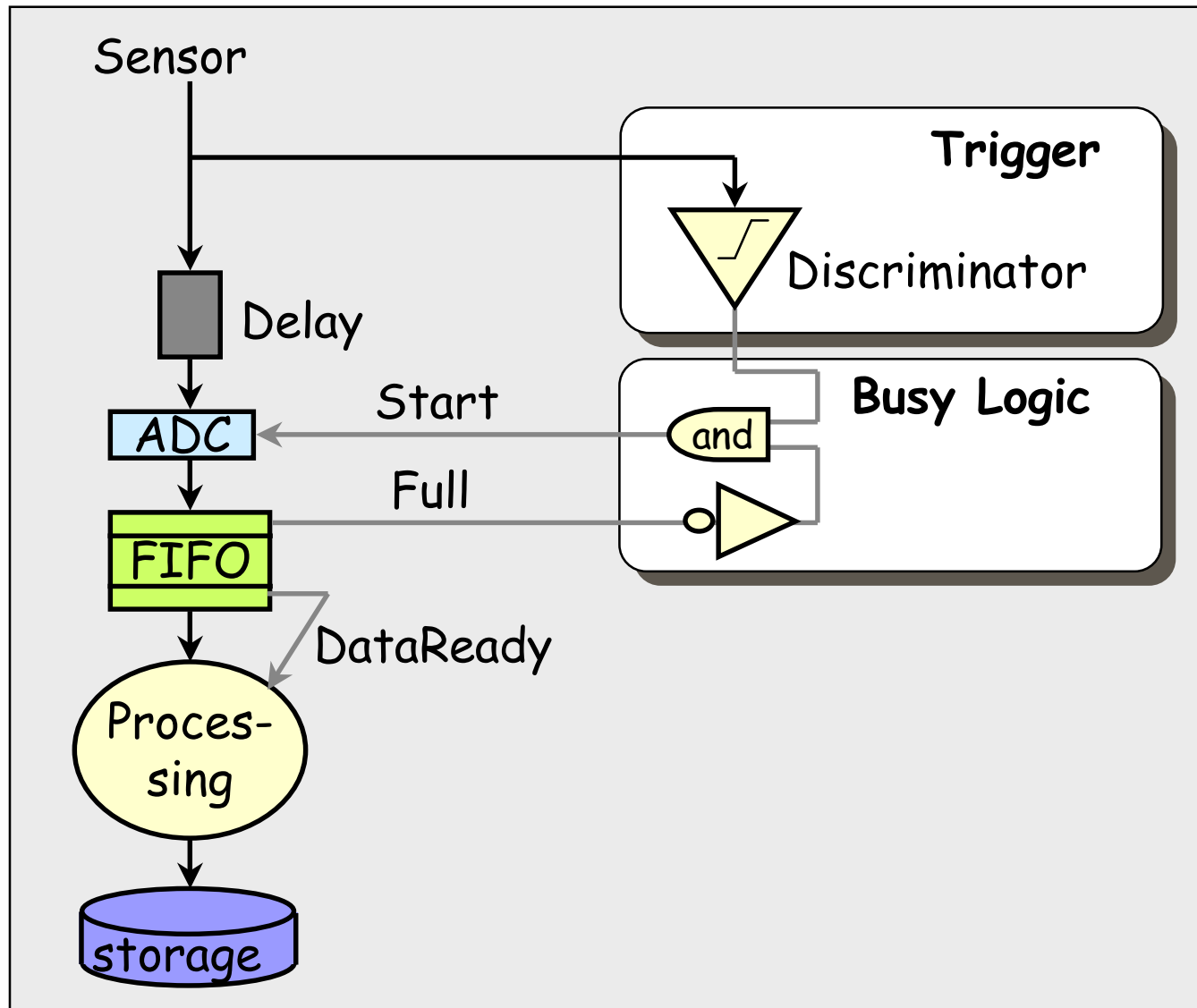
# Trivial DAQ with a real trigger (2)



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



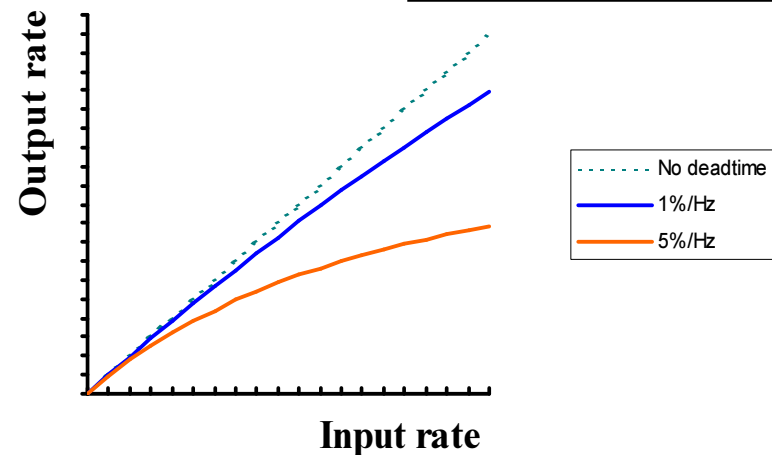
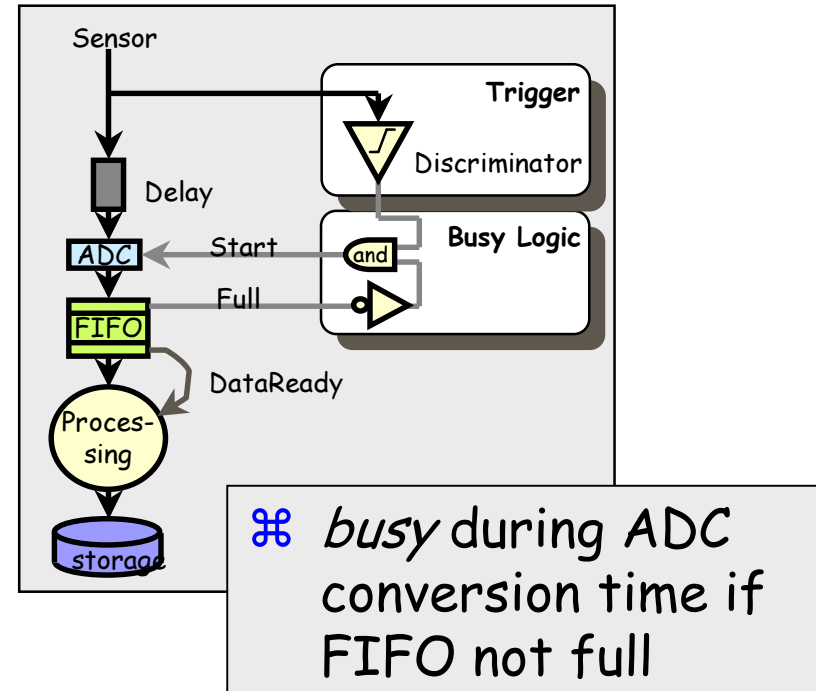
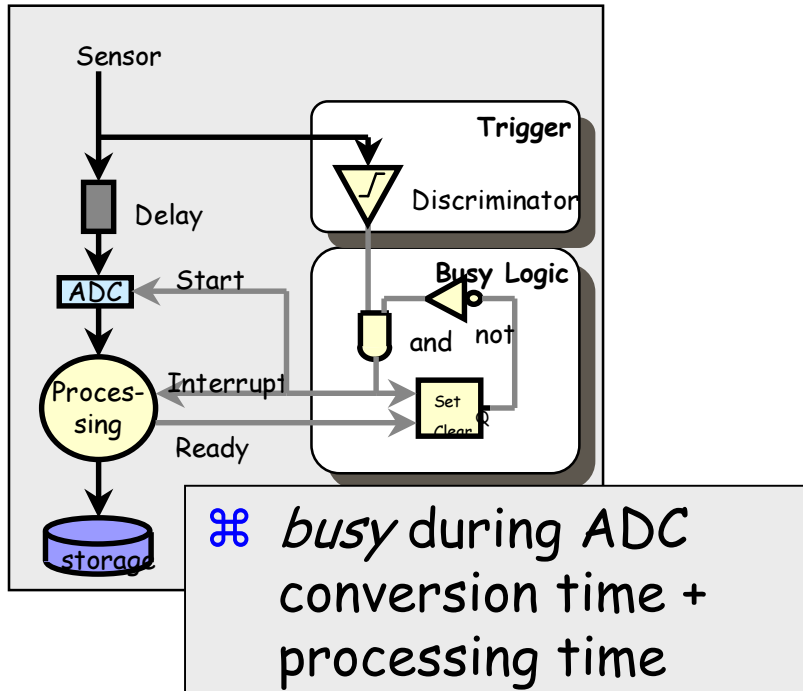
# Trivial DAQ with a real trigger (3)



-> decouple data  
ption



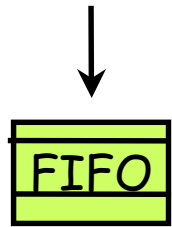
# Derandomizer buffers (queues)



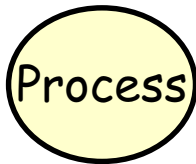


# Queue Theory

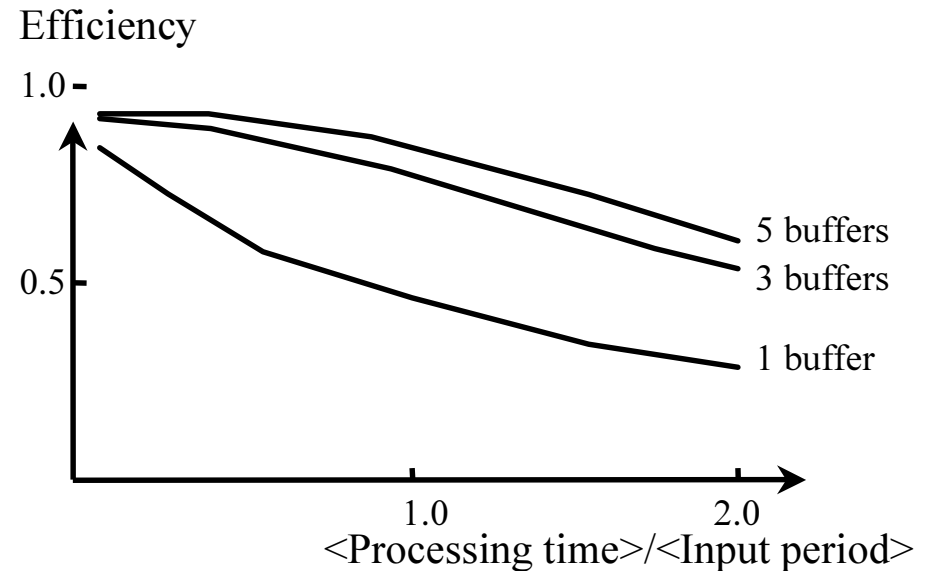
Arrival time following exponential distribution (Poisson process)



Depth 1, 3, 5



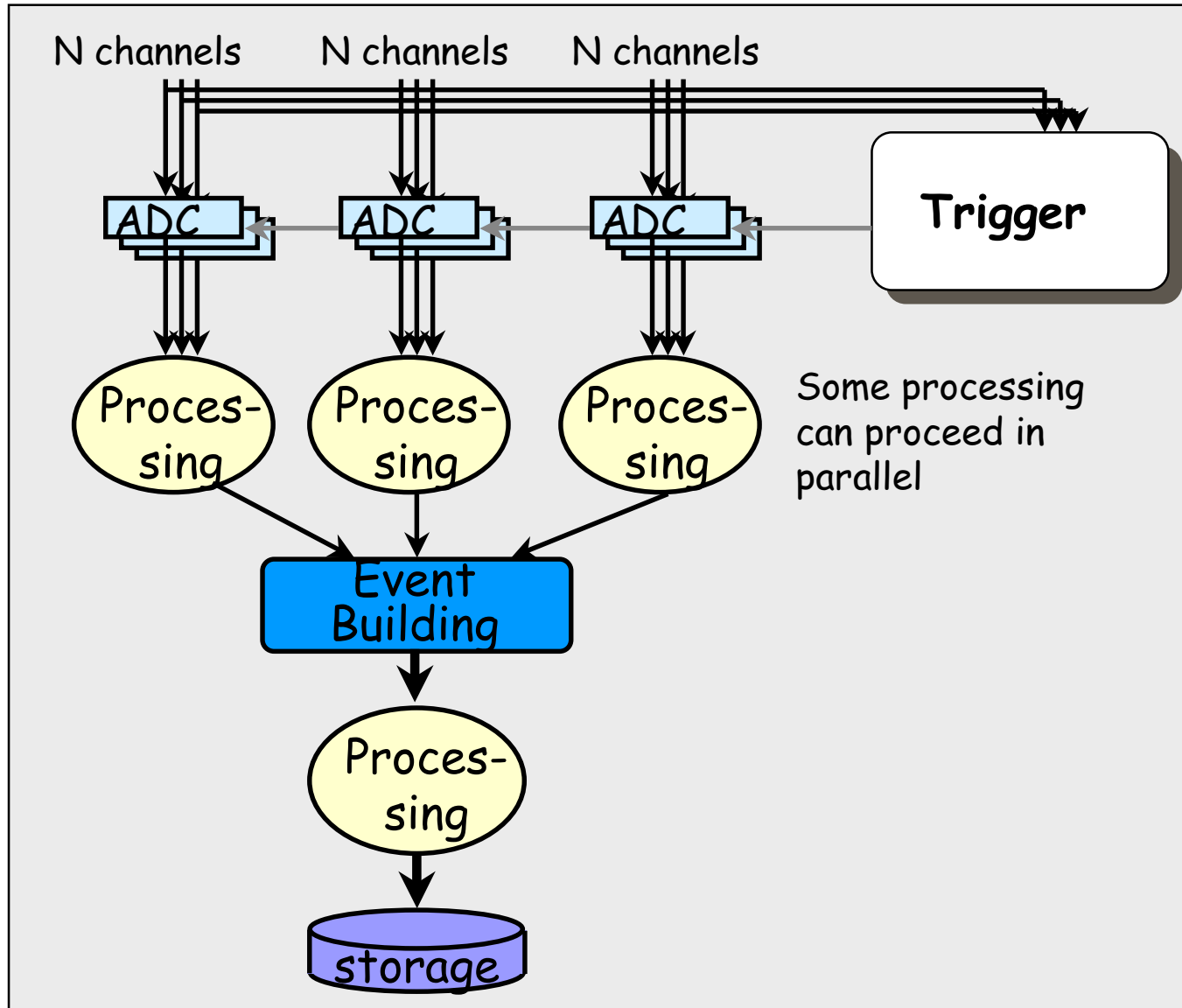
Processing time following gaussian distribution



⌘ For simple cases, the behavior of the system can be calculated analytically using *queue theory*. Soon you have to use simulation techniques.



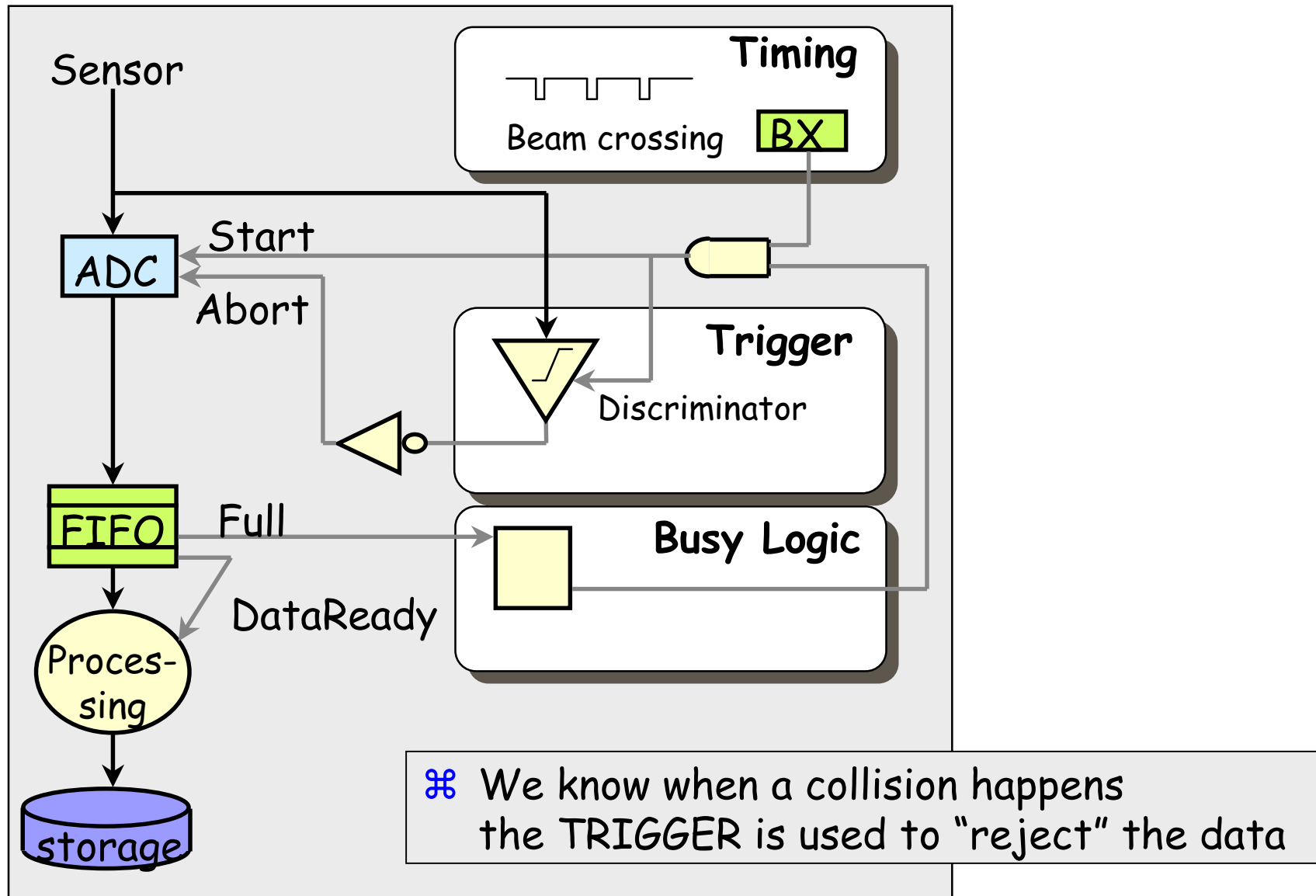
# Less trivial DAQ







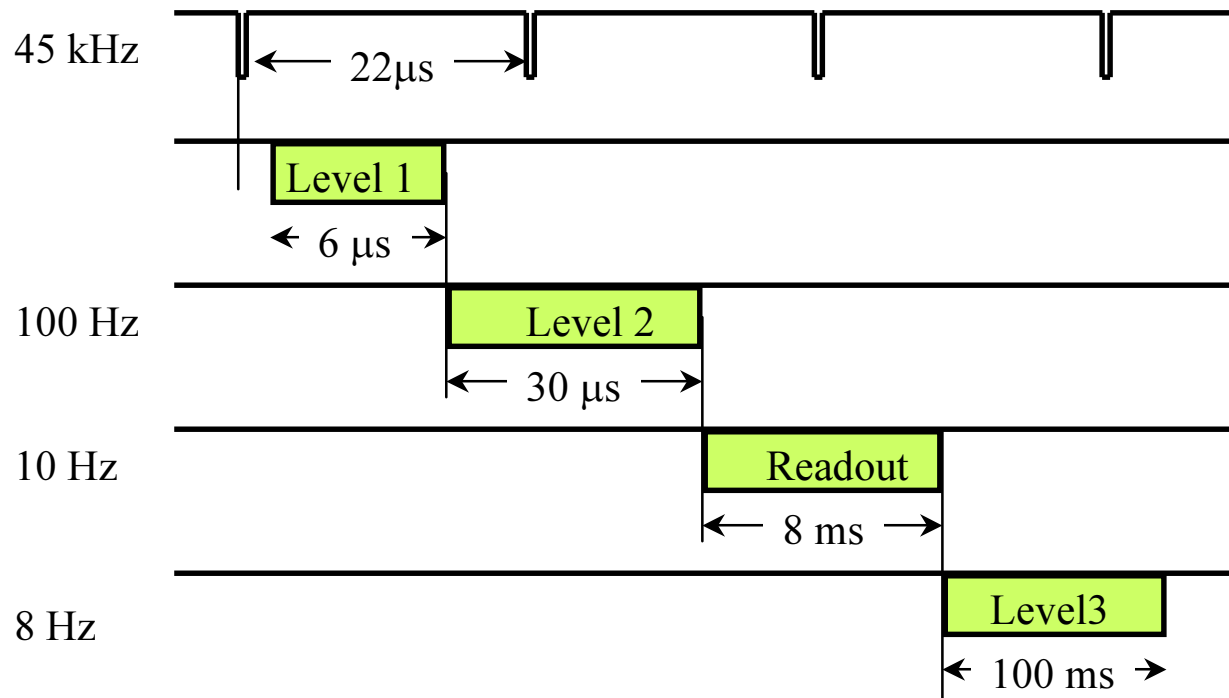
# Trivial DAQ in collider mode





# LEP collider timing

$e^+e^-$  Crossing rate 45 kHz (4 bunches)

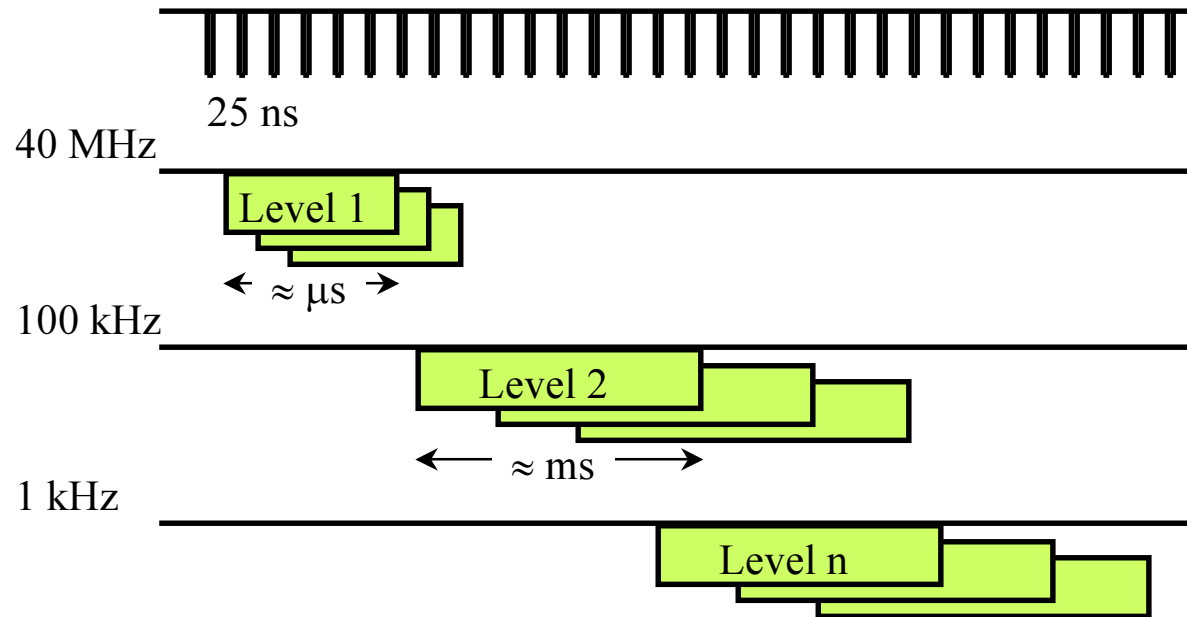


- ⌘ Level 1 trigger latency  $<$  inter bunch crossings  $\rightarrow$  No deadtime
- ⌘ No event overlapping
- ⌘ Most of the electronics outside the detector



# LHC timing

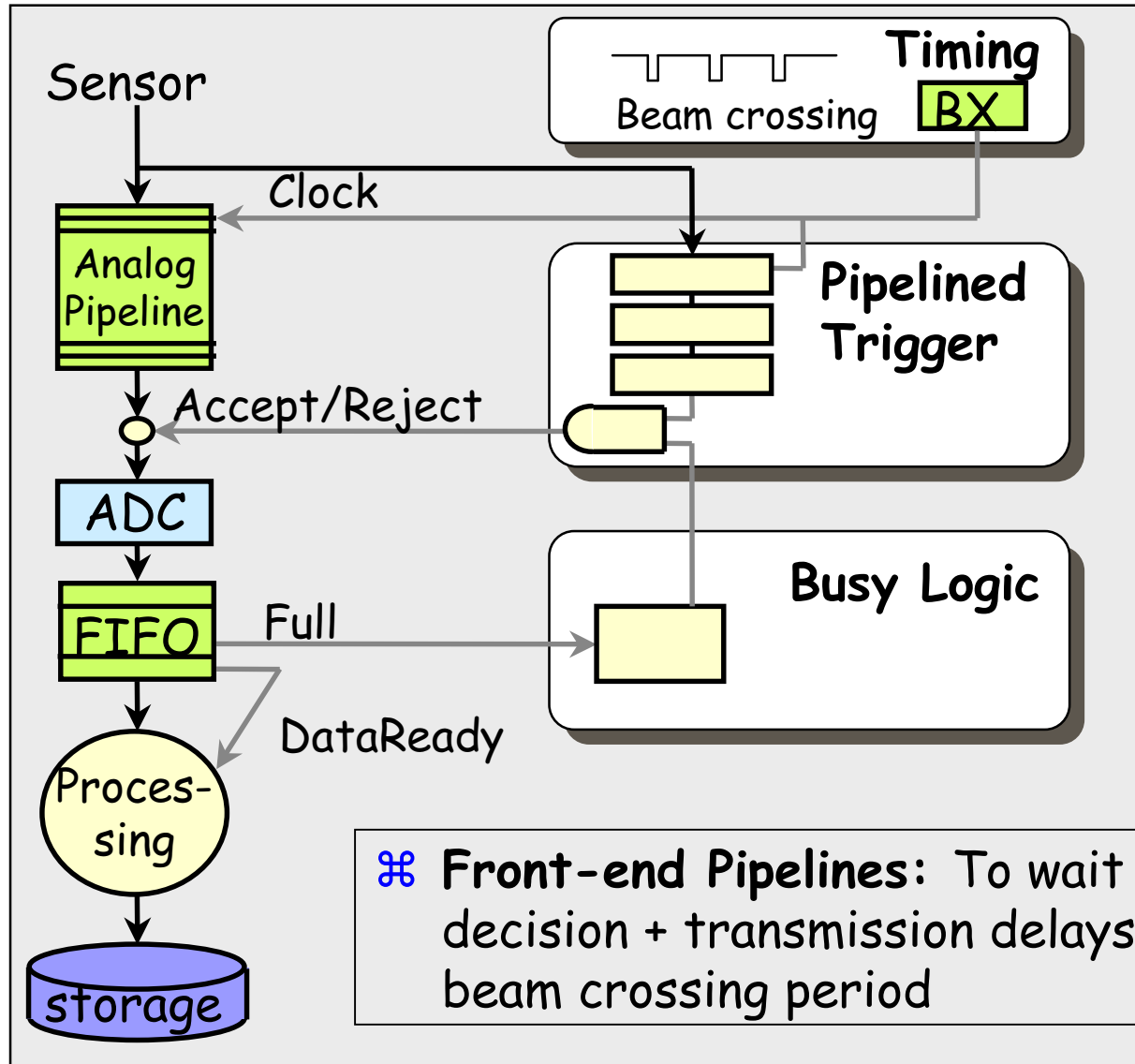
p p crossing rate 40 MHz ( $L=10^{33}-4\cdot 10^{34}\text{cm}^{-2}\text{s}^{-1}$ )



- ⌘ Level 1 trigger time exceeds bunch interval
- ⌘ Event overlap & signal pileup (multiple crossings since the detector cell memory greater than 25 ns)
- ⌘ Very high number of channels



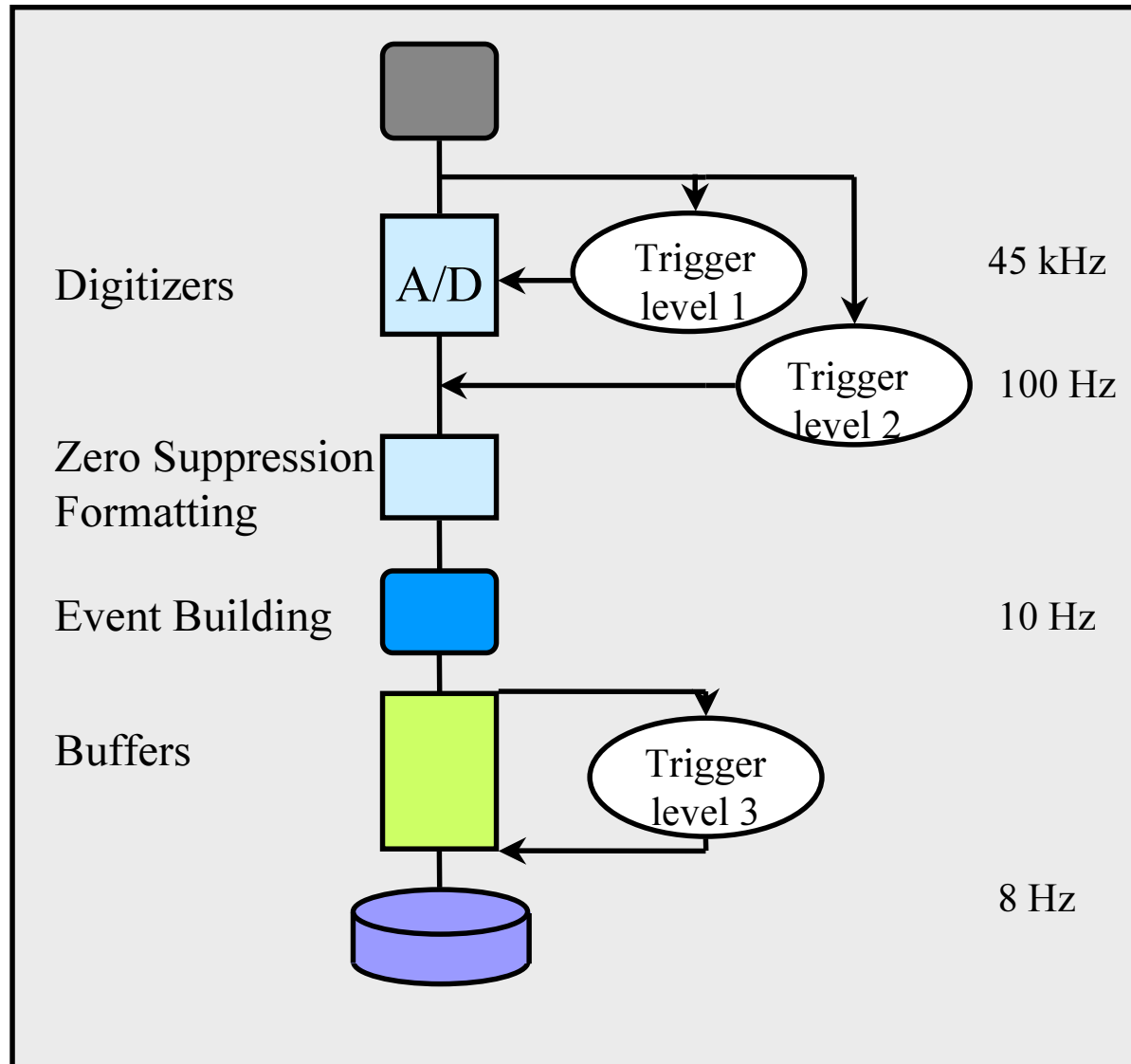
# Trivial DAQ in LHC



⌘ Front-end Pipelines: To wait for the trigger decision + transmission delays are longer than beam crossing period

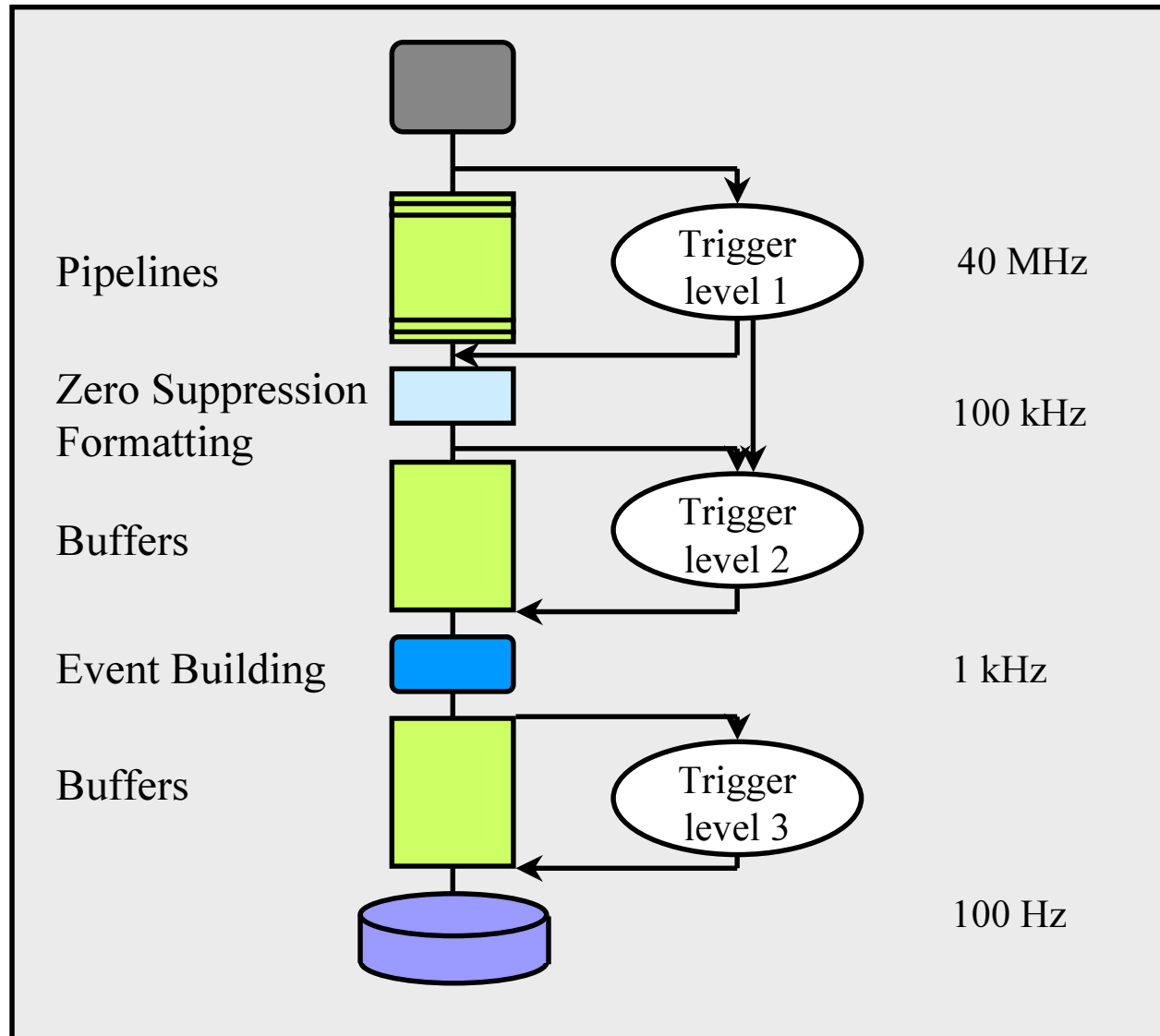


# LEP readout structure



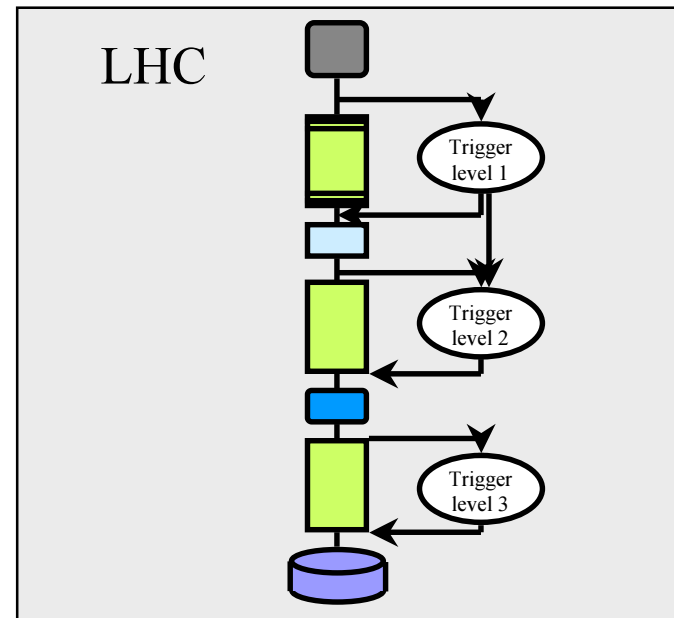
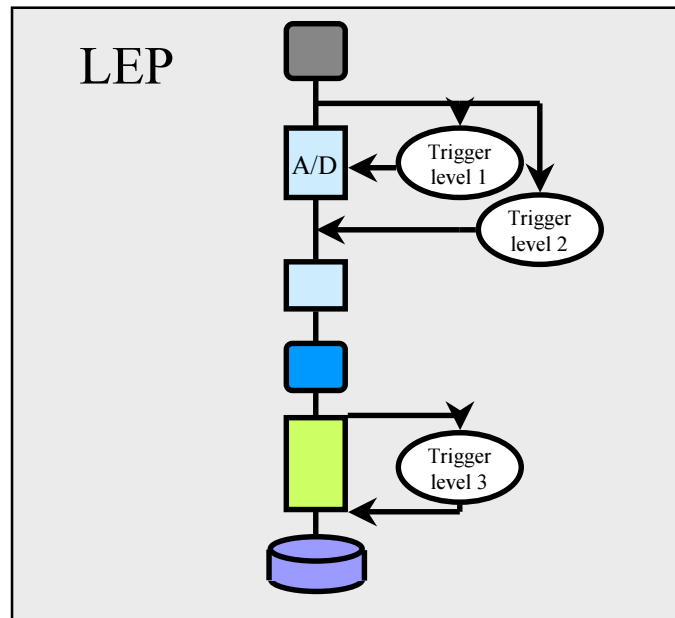


# LHC readout structure





# Readout structures differences



- ⌘ Rates are very different.
- ⌘ At LEP the readout of the digitizers is only done after Level-2 accept (few 10  $\mu$ s).
- ⌘ At LEP the trigger data for Level-2 comes from detector.
- ⌘ At LHC you need data pipelines and data buffers to store the events during Level-1 and Level-2 processing



# Front-end essentials

Clara Gaspar CERN/EP-LBC





# Front-Ends

## ⌘ Detector dependent (Home made)

### ☑ On Detector

- ☒ Pre-amplification, Discrimination, Shaping amplification and Multiplexing of a few channels

### ☑ Transmission

- ☒ Long Cables (50-100 m), electrical or fiber-optics

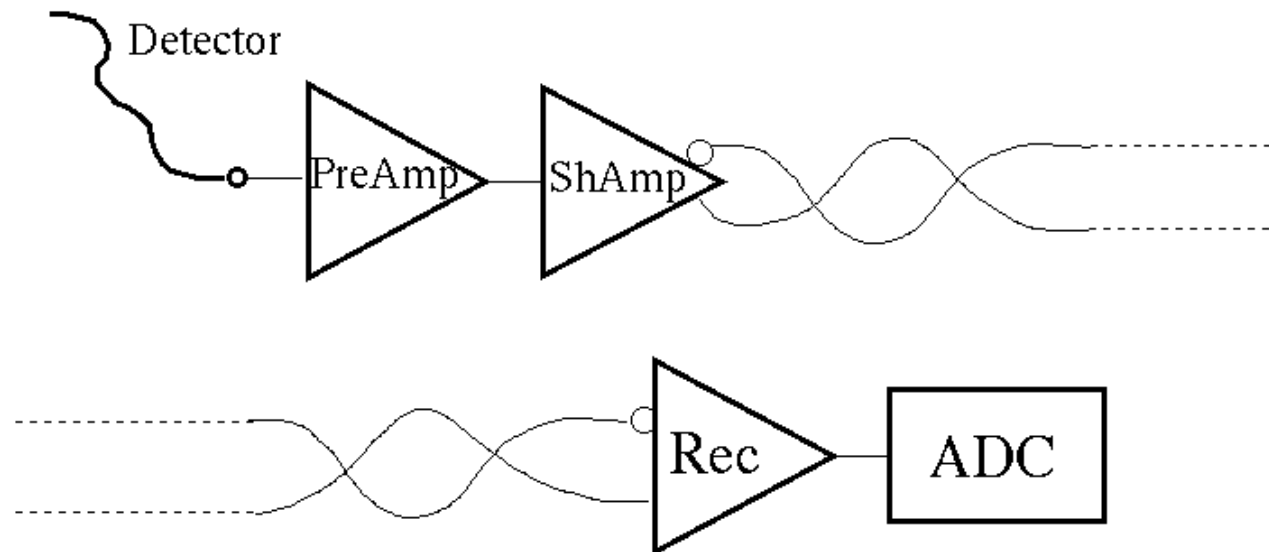
### ☑ In Counting Rooms

- ☒ Hundreds of FE crates :  
Reception, A/D conversion and Buffering



## Front-Ends (2)

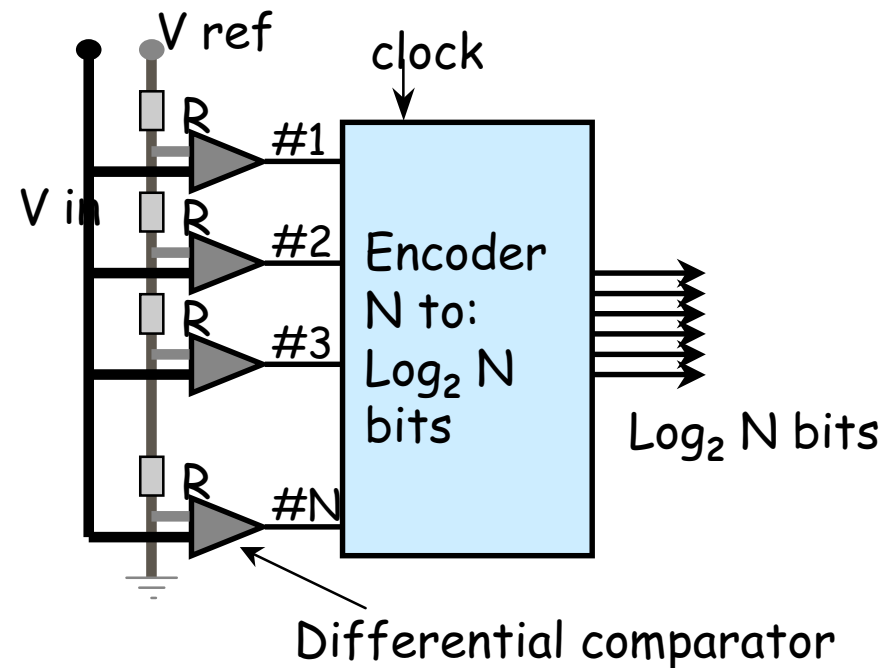
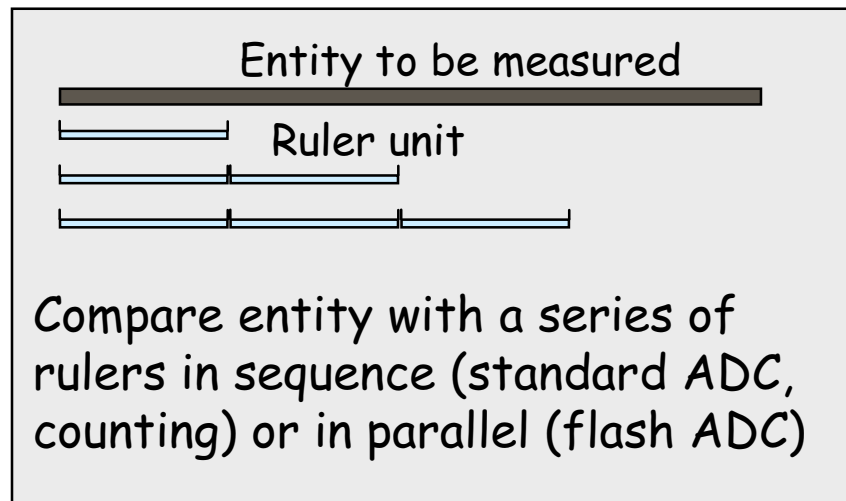
⌘ Example: Charged particle creates ionization  $\rightarrow$   $e^-$  drifted by electric field



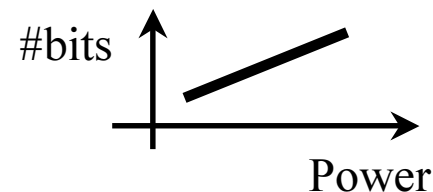


# Analog to digital conversion

- ⌘ Digitizing means measuring something (charge, amplitude, time, ...) comparing it with a reference unit.
- ⌘ Ex: Flash ADC

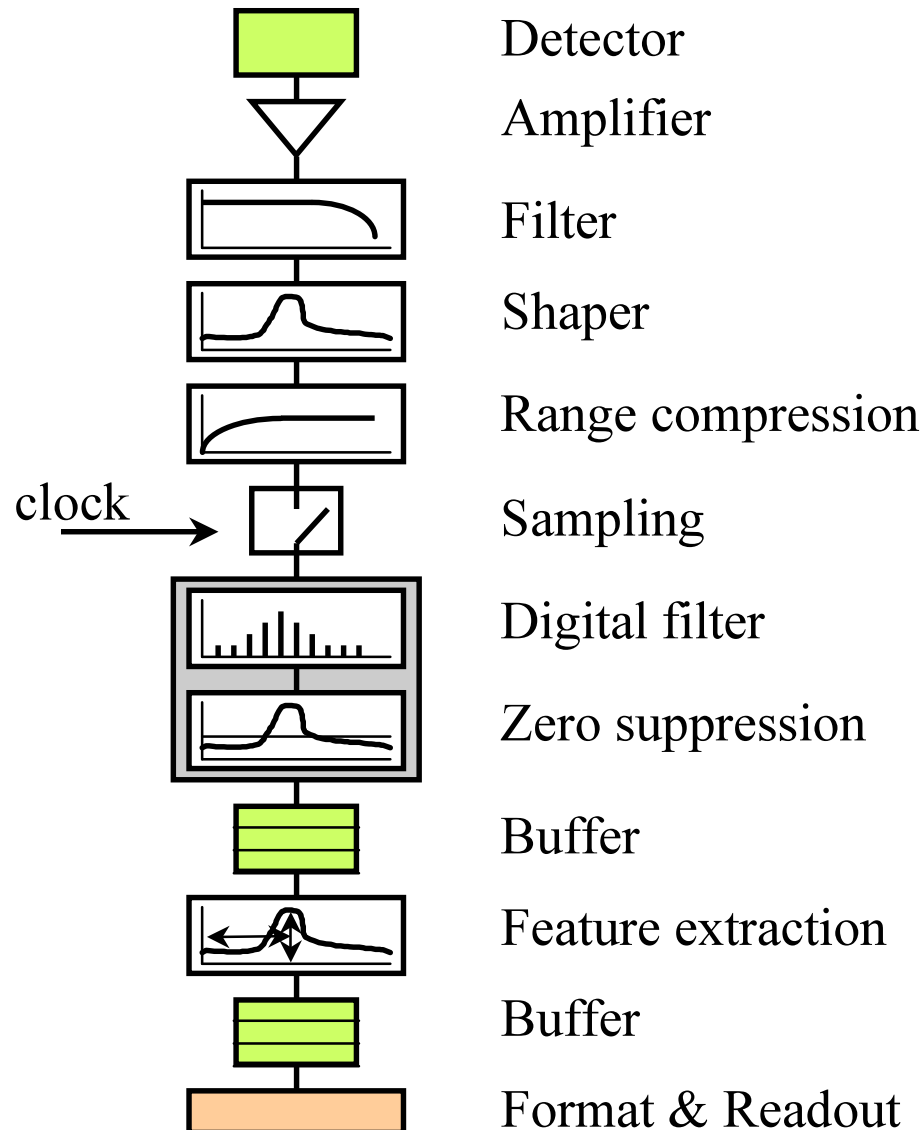


- ⌘ The fastest method
- ⌘ Prohibitive for  $> 10$  bits



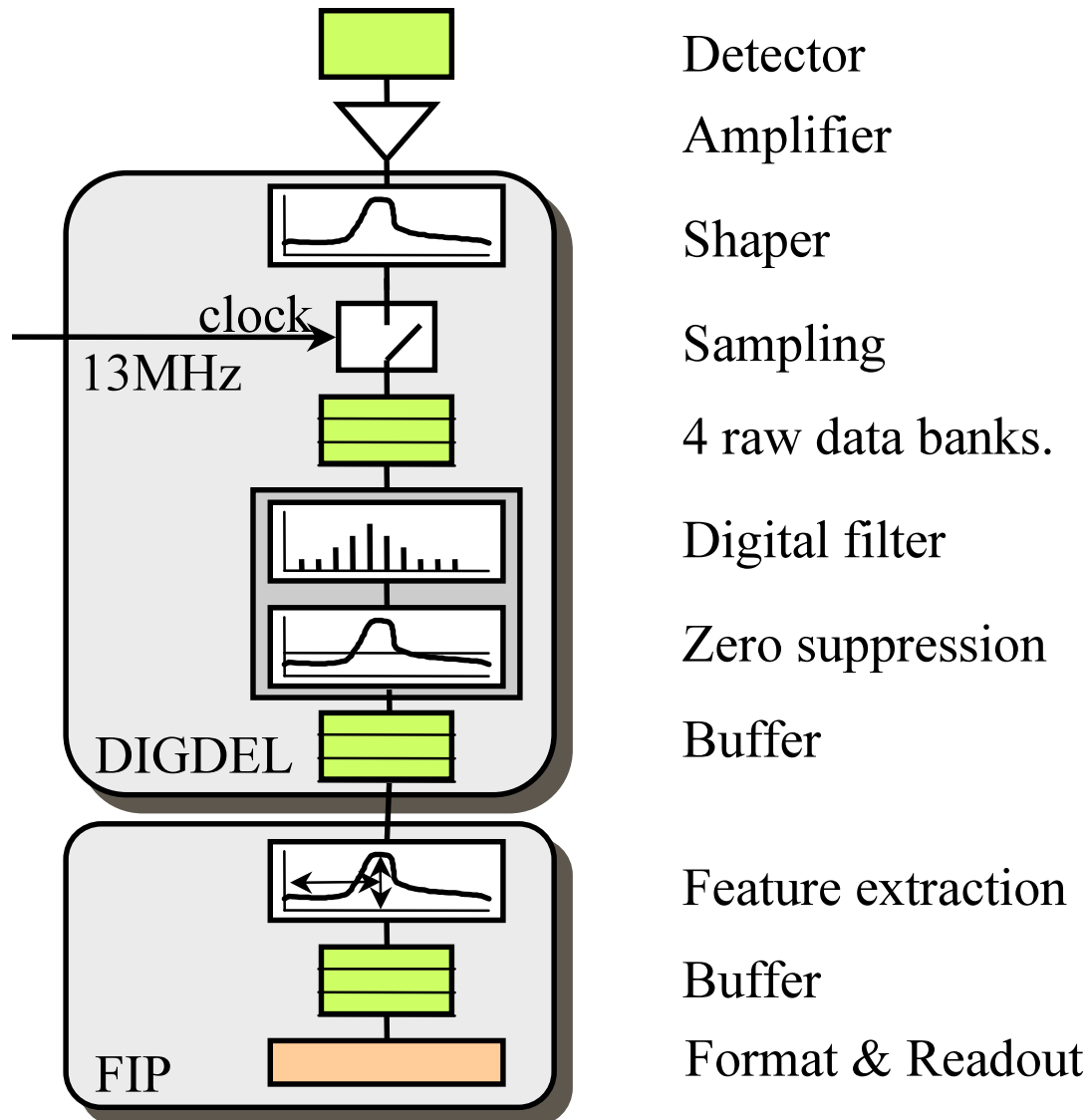


# Front-end structure





# Front-end example: DELPHI TPC



- Detector
- Amplifier
- Shaper
- Sampling
- 4 raw data banks.
- Digital filter
- Zero suppression
- Buffer
- Feature extraction
- Buffer
- Format & Readout



# F.E. at LHC (challenges):

## ☒ Small bunch interval

- Pipeline buffering (analog) to keep events until trigger decision ( $2\mu\text{s}$  - 100 col.)
- Very Fast A/D conversion
- Very precise timing distribution (order of sub-ns)

## ☒ Large Nr. of channels

- High Integration (Custom VLSI chips)

## ☒ Piled-up Events

- Digital Signal Processors (DSP) to sort them out

## ☒ Large amount of data

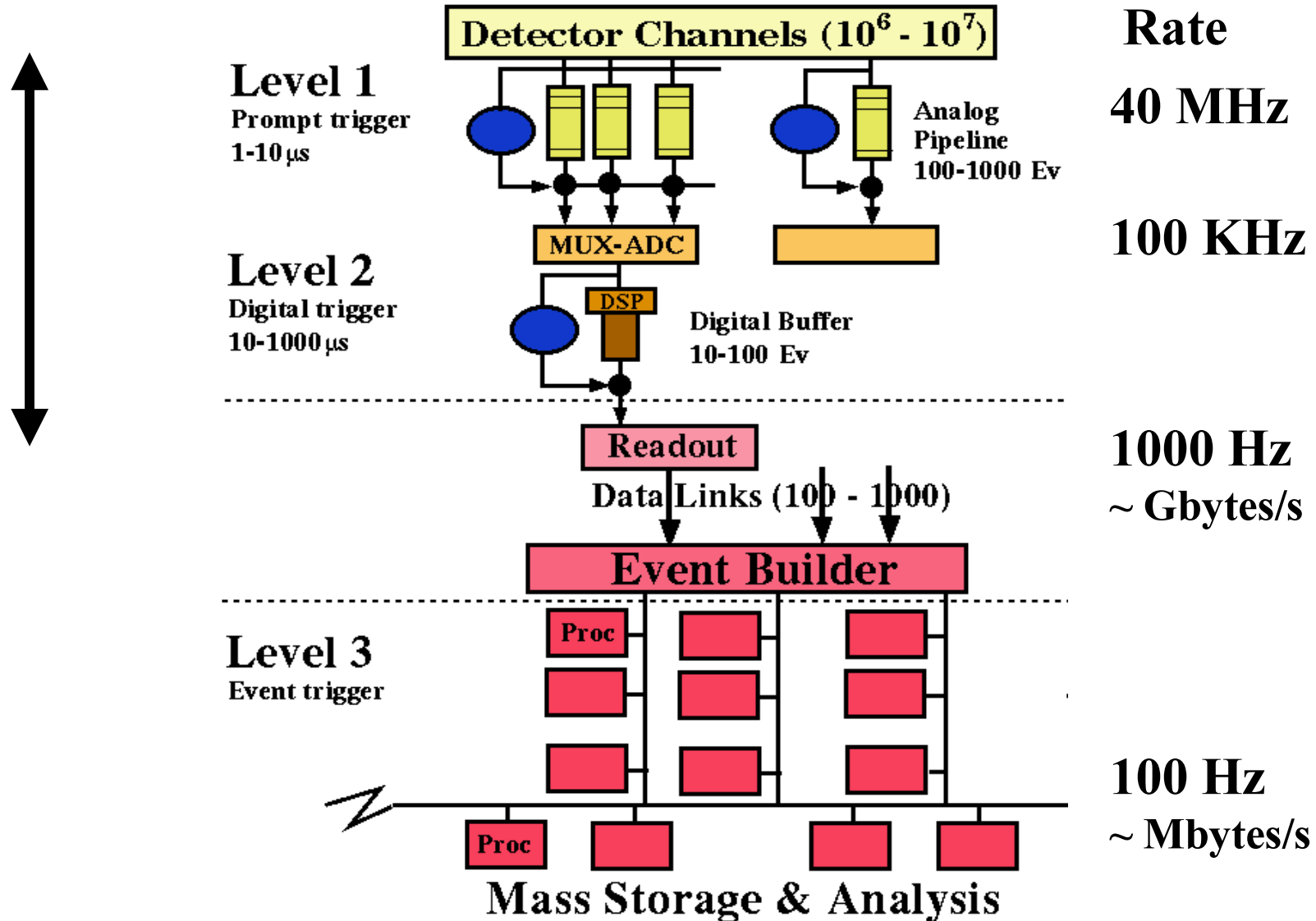
- Data Compressors (instead of zero skipping)

## ☒ Power Consumption

## ☒ Radiation Levels



# Front-end at LHC





# Trigger

Clara Gaspar CERN/EP-LBC

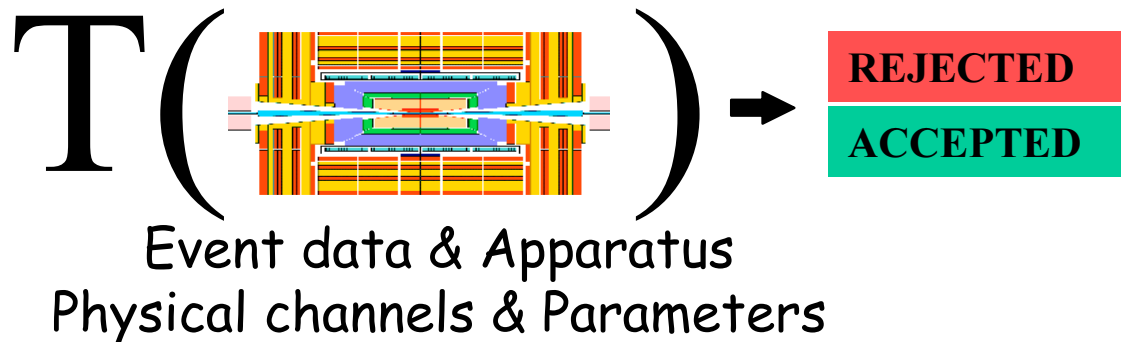




# Trigger System

The **trigger system** is the system which triggers the recording of the data in an experiment.

The trigger is a function of:



Since the detector data are not promptly available and the function is highly complex,  $T(\dots)$  is evaluated by successive approximations called:

**TRIGGER LEVELS**



# Trigger Goals

## ⌘ Leptonic Collisions ( $e^+e^-$ colliders)

⊠ **Cross Section** - probability of interaction  
(unit = barn =  $10^{-24}$  cm<sup>-2</sup>)

⊠  $\sigma_{\text{tot}} = 30 - 40$  nb ( $30-40 * 10^{-33}$  cm<sup>-2</sup>)

⊠ **Luminosity** - density of crossing beams  
(~ nb. of particles per section)

⊠  $L = 10^{31}$  cm<sup>-2</sup>s<sup>-1</sup> (LEP)

→ **Event rate ( $L * \sigma$ ) 0.1 - 1 events/s**

⊠  $\sigma_{\text{interesting}} \approx \sigma_{\text{tot}}$  (All interactions are interesting)

→ **No physics rejection needed**

⊠ But background rejection is crucial to perform precision measurements:

⊠ **Cosmics, Beam-gas, Off-momentum  $e^{\pm}$ , Detector noise**



# Trigger Goals (2)

## ⌘ Hadronic collisions (fixed target or pp)

### ⊡ Cross Section

⊗  $\sigma_{\text{tot}} = 30 - 40 \text{ mb} \quad (30-40 * 10^{-27} \text{ cm}^{-2})$

### ⊡ Luminosity

⊗  $L = 10^{30} \text{ cm}^{-2}\text{s}^{-1}$  (SPS),  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$  (fixed target),  
 $10^{34} \text{ cm}^2\text{s}^{-1}$  (LHC)

➔ **Event rate  $10^5 - 10^9$  events/s**

⊡  $\sigma_{\text{interesting}} \approx \text{nb} \rightarrow \text{pb}$

➔ **Rejection needed of  $10^6 \rightarrow 10^{13}$  (LHC)**



# Trigger Design (Physics Aims)

## ⌘ Leptonic interactions

⊞ High rejection of background

⊞ Criteria:

⊞ Very good efficiency for selected channel  $\approx 100\%$

⊞ Good rejection background. Depends on physics/background ratio, i.e. on machine conditions.  $\approx 10$

⊞ Good monitoring of efficiency  $\pm 0.1\%$

## ⌘ For High precision measurements:

⊞ "A 60% well known efficiency is much better than a 90% uncertain efficiency"



# Trigger Design (Physics Aims)

## ⌘ Hadronic Interactions

⊞ High rejection of physics events

⊞ Criteria:

⊞ Good efficiency for selected channel

50%

≥

⊞ Good rejection of uninteresting events

≈10<sup>6</sup>

⊞ Good monitoring of efficiency

±0.1%

⌘ Similar criteria, but in totally different ranges of values



# Trigger Design (2)

## ⌘ Simulation studies

- ⊞ It is essential to build into the experiment's simulation the trigger simulation.
- ⊞ For detector/machine background rejection
  - ⊗ Try to include detector noise simulation
  - ⊗ Include machine background (not easy).
  - ⊗ For example at LEP the trigger rate estimates were about 50-100 times too high.
- ⊞ For physics background rejection, the simulation must include the generation of these backgrounds



# Trigger Performance

## ⌘ Efficiency monitoring

⊞ Key point: REDUNDANCY

⊞ Each event should fire at least two independent sub-triggers (the global trigger is a logic OR of all sub-triggers)

⊞ Use the simulation (or common sense!) to evaluate for different sub-triggers how redundant they are.

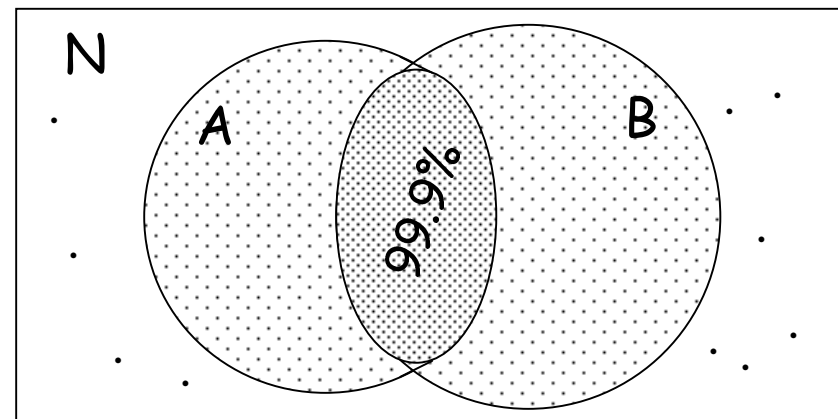
⊞ Use DATA (after reconstruction) to compute efficiency.

## ⌘ Ex: Electrons Efficiency

A = TPC (tracking)

B = Calorimeter

$$\epsilon_A = \frac{N(A \cap B)}{N(B)}$$





# Trigger Levels

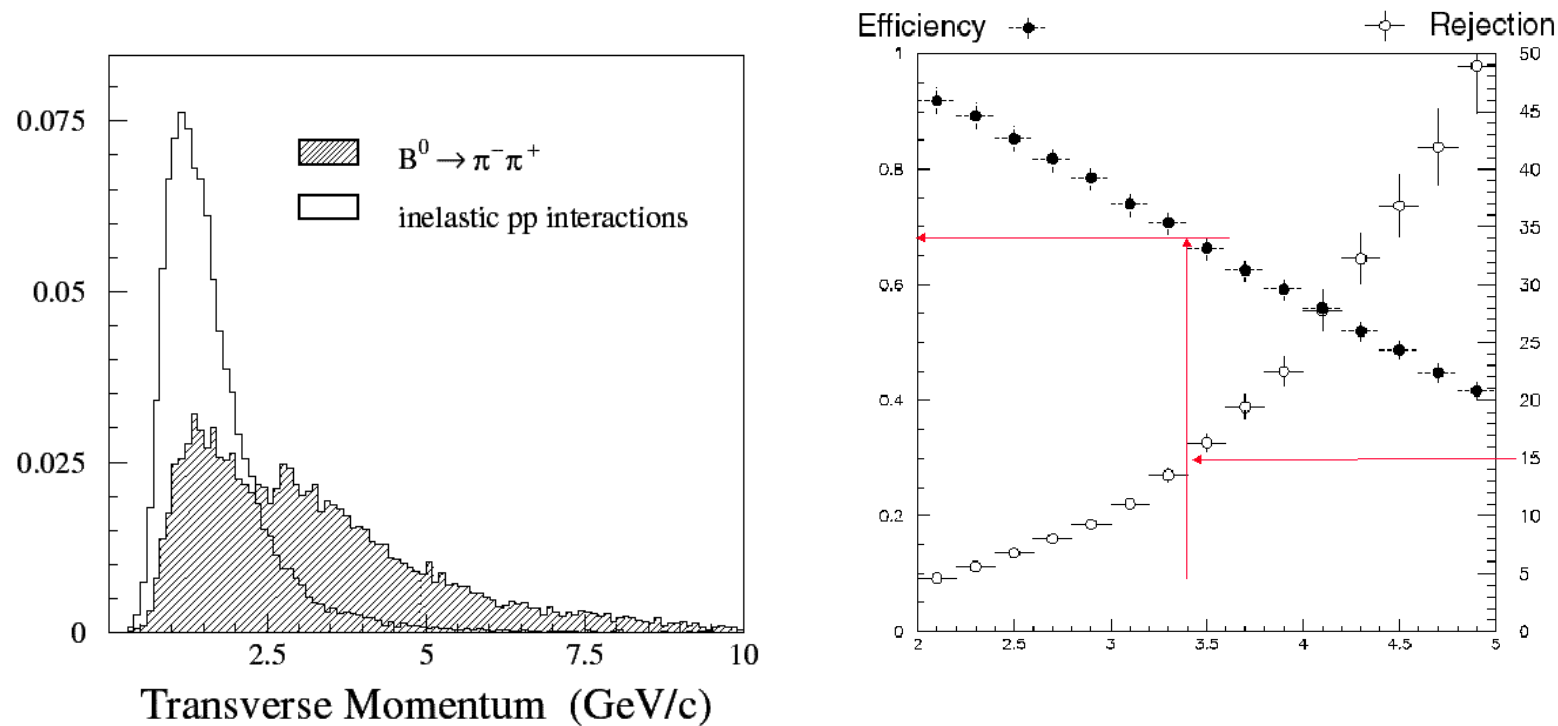
- ⌘ Since the detector data is not promptly available and the trigger function is highly complex, it is evaluated by **successive approximations**.
- ⌘ We need to optimize the amount of data needed and the time it takes to provide a decision.
- ⌘ Trigger levels
  - ☒ Hardware trigger: *Fast* trigger which uses crude data from few detectors and has normally a limited time budget and is usually implemented using hardwired logic.  
⇒ **Level-1** sometimes **Level-2**
  - ☒ Software triggers: Several trigger levels which refines the crude decisions of the hardware trigger by using more detailed data and more complex algorithms. It is usually implemented using processors running a program.  
⇒ **Level-2, Level-3, Level-4, ...**





# Example: B meson trigger in LHCb

⌘ Discriminating variable: Transverse momentum ( $P_T$ )



⌘ Efficiency versus background rejection



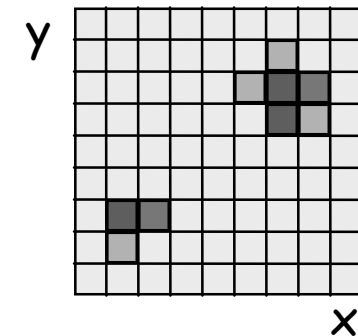
# LHCb LO Calorimeter Trigger

## ⌘ Select the particles with the highest $P_T$

- ⊡ For the Level 0 decision, we need only the particle with the highest  $P_T$
- ⊡ Check if above threshold

## ⌘ Identify hot spots

- ⊡ Detect a high energy in a 'small' surface
- ⊡ Use a square of  $2 \times 2$  cells area
  - ⊗  $8 \times 8$  cm<sup>2</sup> in the central region of ECAL
  - ⊗ more than  $50 \times 50$  cm<sup>2</sup> in the outer region of HCAL

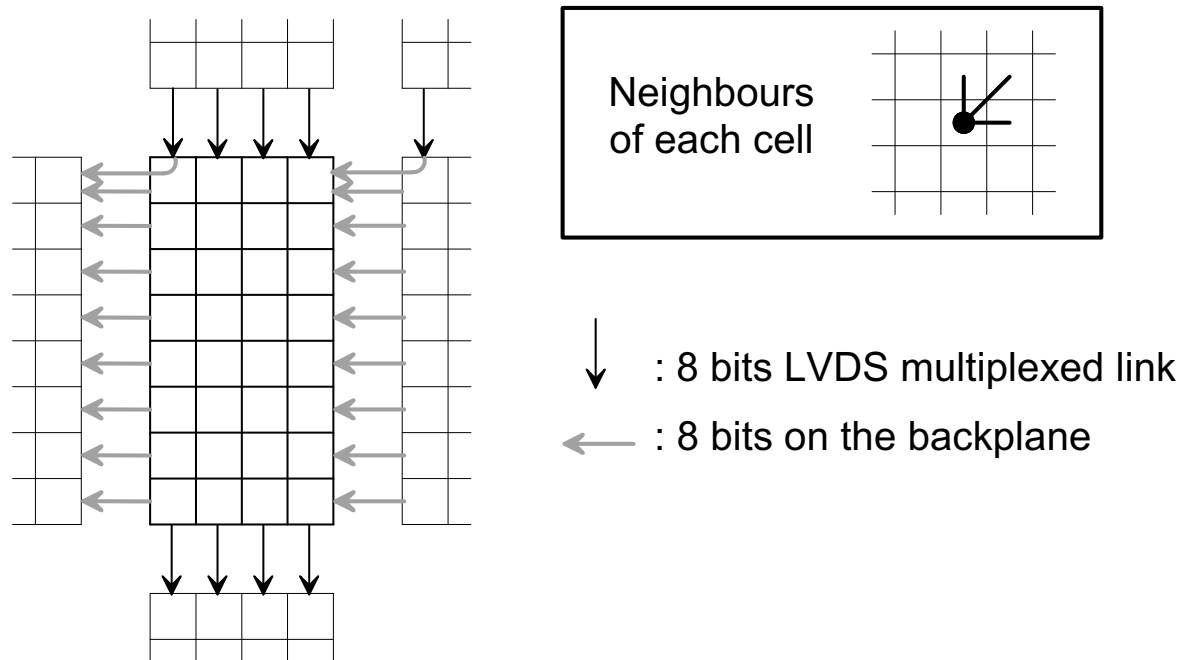




# LHCb L0 Calorimeter Trigger (2)

## ⌘ Build the 2x2 sums

- ⊞ Work inside a 32 channels (8x4) front-end card
  - ⊞ To obtain the 32 2x2 sums, one needs to get the 8 + 1 + 4 neighbours
  - ⊞ Via the backplane (bus) or dedicated point-to-point cables

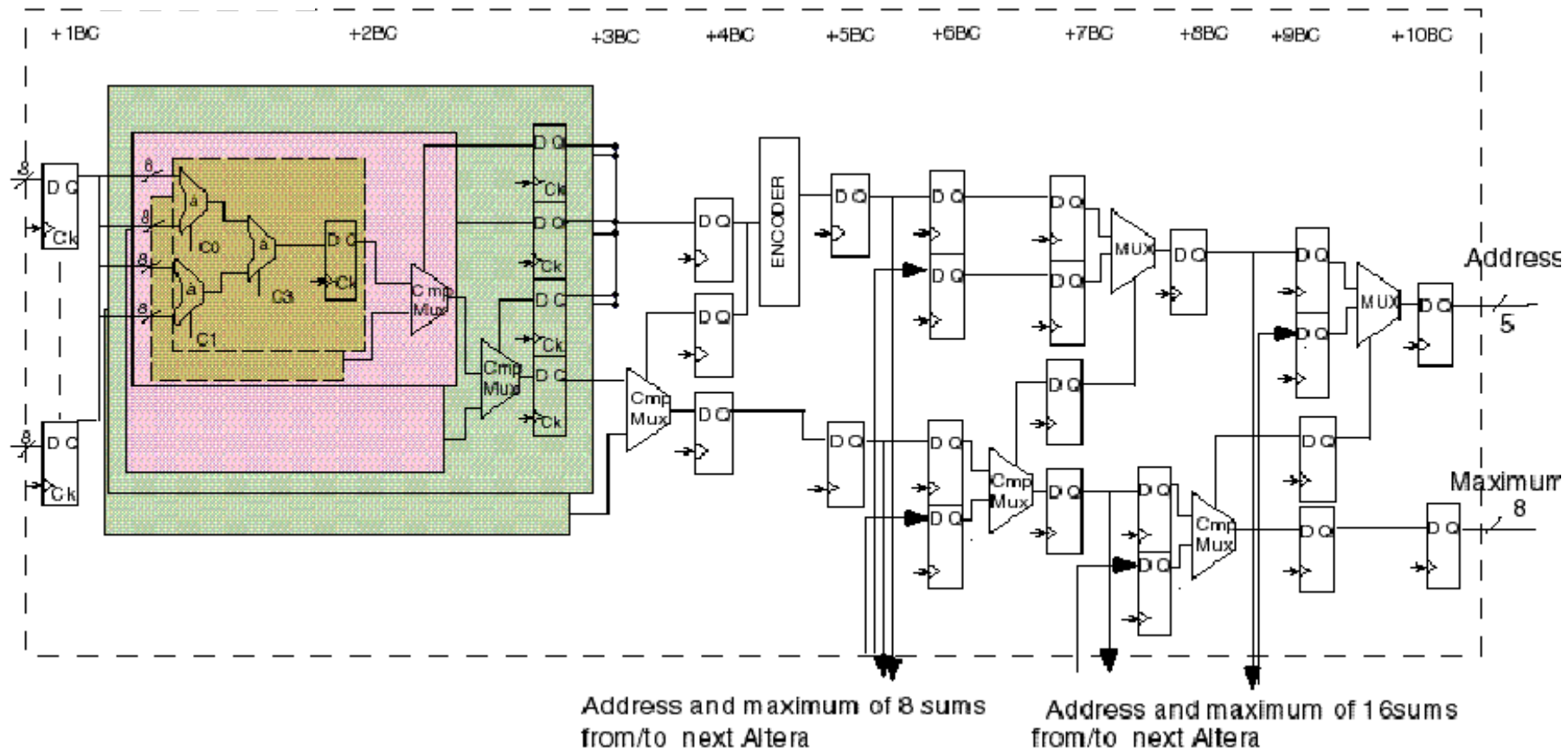
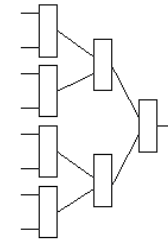




# LHCb LO Calorimeter Trigger (3)

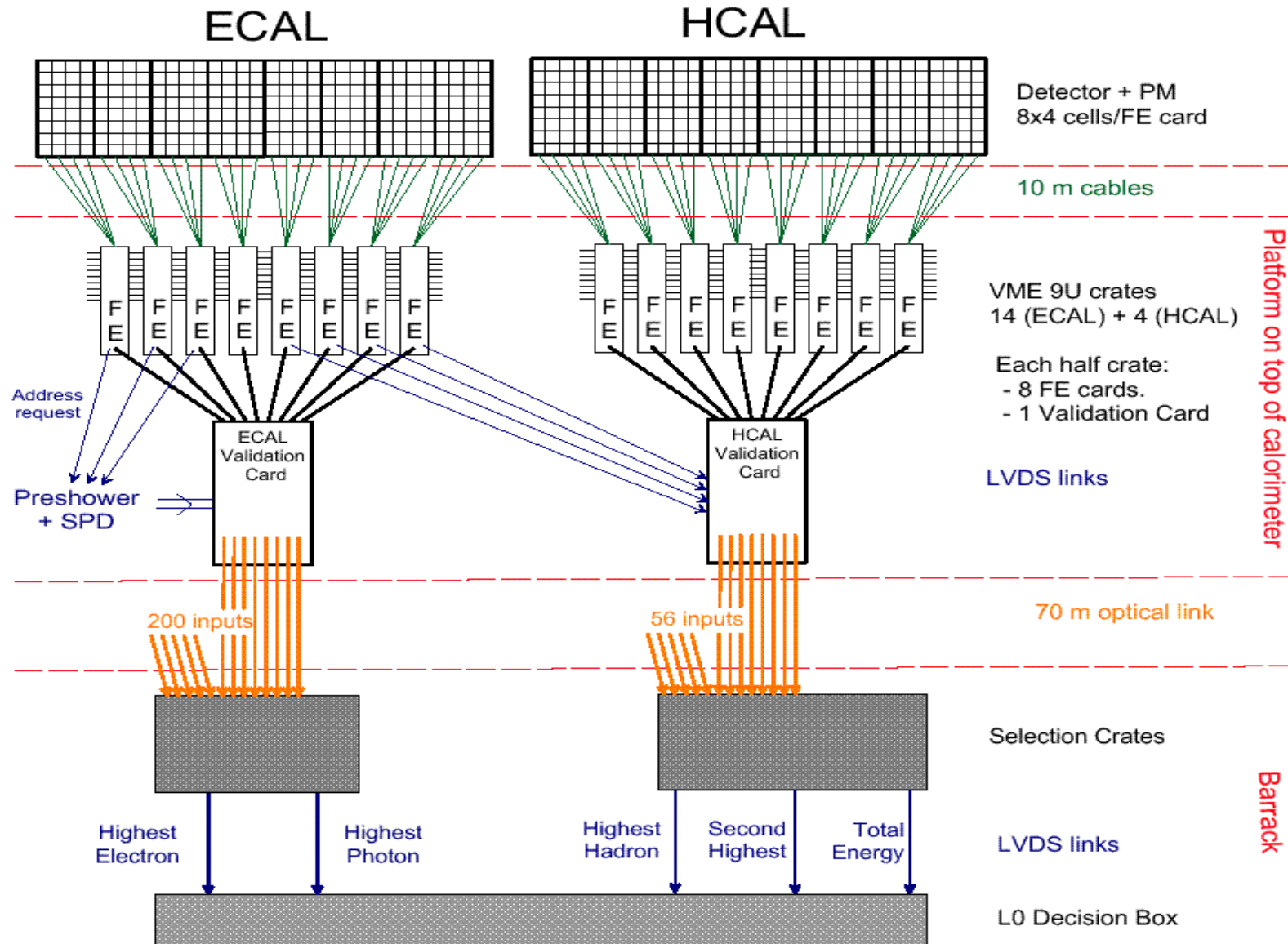
⌘ Select the local maximum in the card

- ▣ Simple comparison of the summed  $E_T$ .
- ▣ Currently implemented in 4 ALTERA FPGA's





# LHCb LO Calorimeter Trigger (4)





# Central Decision Logic

## ⌘ Look Up Tables

- ☒ Use N Boolean informations to make a single decision: YES / NO
- ☒ Use a RAM of  $2^N$  bits
- ☒ Example: N=3

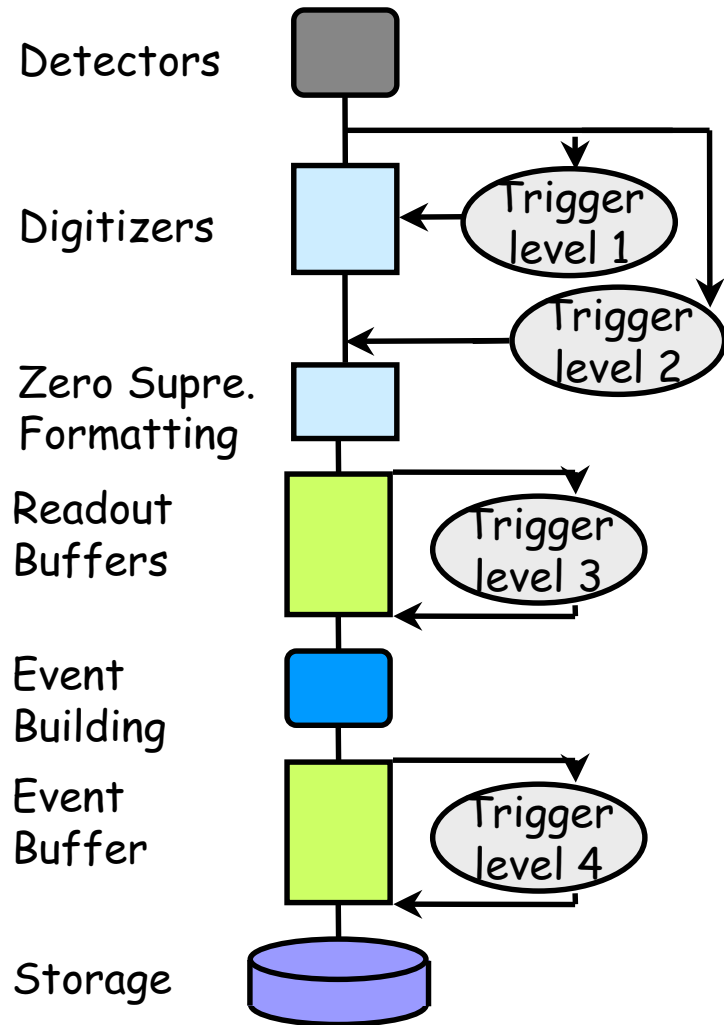
	EM	$\mu$	TR	RAM	
RAM Address	0	0	0		No track, no EM
	0	0	1		Track, no $\mu$
	0	1	0		$\mu$ , track inefficient
	0	1	1		$\mu$ , track
	1	0	0		EM, no track
	1	0	1		EM, track, no $\mu$
	1	1	0		EM, $\mu$ , track ineffic.
	1	1	1		EM, $\mu$ , track

## ⌘ To Trigger On:

- ☒ "Single Photons"  
→ 0,0,0,0,1,0,0,0
- ☒ "At least one  $\mu$ "  
→ 0,0,1,1,0,0,1,1



# Trigger Levels in DELPHI (LEP)



## ⌘ Level-1 (3 $\mu$ s) (hardware proc.)

### ⌘ Single detector Information:

- ⌘ Energy: calorimeter (EM and Had.)
- ⌘ Tracks: counting, coincidences
- ⌘ Muons: Had. Cal. and Muon Chambers
- ⌘ Luminosity: analog sums

## ⌘ Level-2 (36 $\mu$ s) (hardware proc.)

### ⌘ Detector Coincidences:

- ⌘ Accept only tracks coming from the IP.
- ⌘ Beam-gas rejection. Uses the TPC

## ⌘ Level-3 ( $\approx$ ms) (In parallel for each sub-detector: OS9 processors)

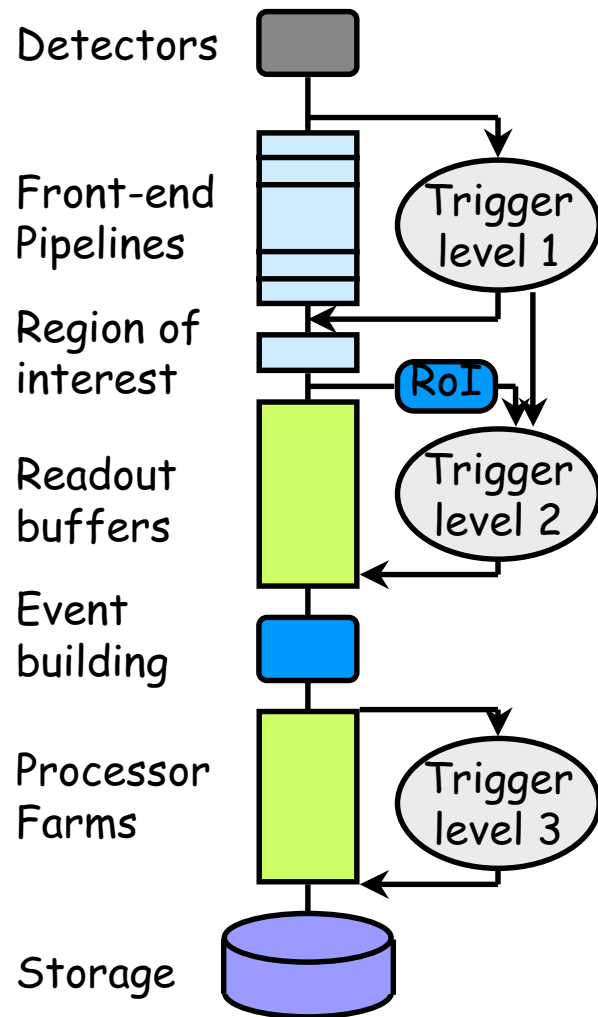
### ⌘ Verifies L2 triggers with digitized data

## ⌘ Level-4 ( $\approx$ ms) (a small farm: 3 alpha CPU)

- ⌘ Reconstructs the event using all data
- ⌘ Rejects Empty Events
- ⌘ Tagging of interesting physics channels



# Trigger Levels in ATLAS (LHC)



⌘ Level-1 ( $3.5 \mu\text{s}$ ) (custom processors)

- ⊞ Energy clusters in calorimeters
- ⊞ Muon trigger: tracking coincidence matrix.

⌘ Level-2 ( $100 \mu\text{s}$ ) (specialized processors)

- ⊞ Few Regions Of Interest relevant to trigger decisions
- ⊞ Selected information (ROI) by routers and switches
- ⊞ Feature extractors (DSP or specialized)
- ⊞ Staged local and global processors

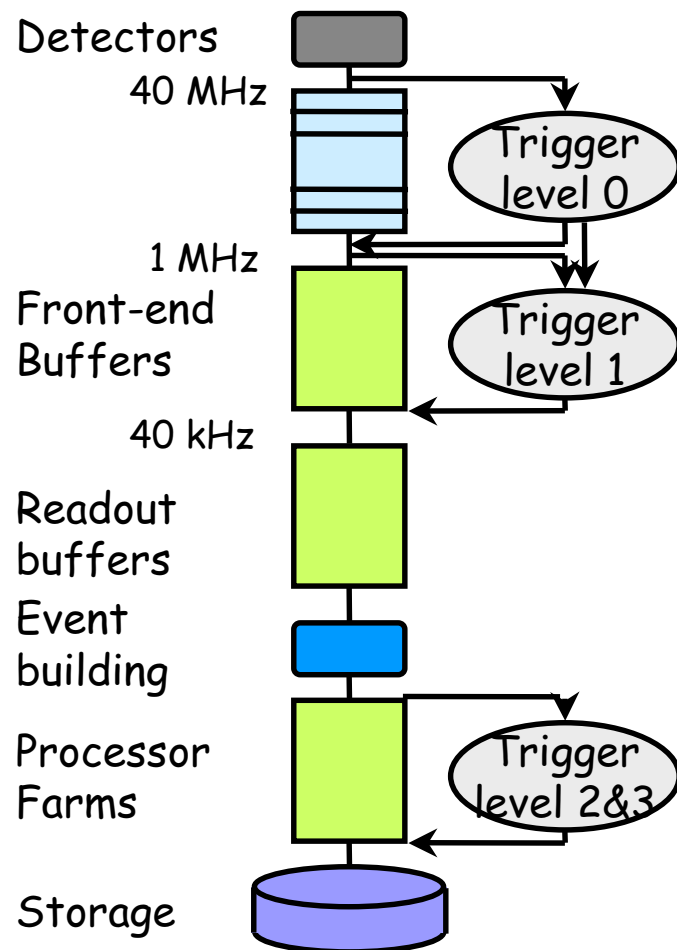
⌘ Level-3 ( $\approx\text{ms}$ ) (commercial processors)

- ⊞ Reconstructs the event using all data
- ⊞ Selection of interesting physics channels





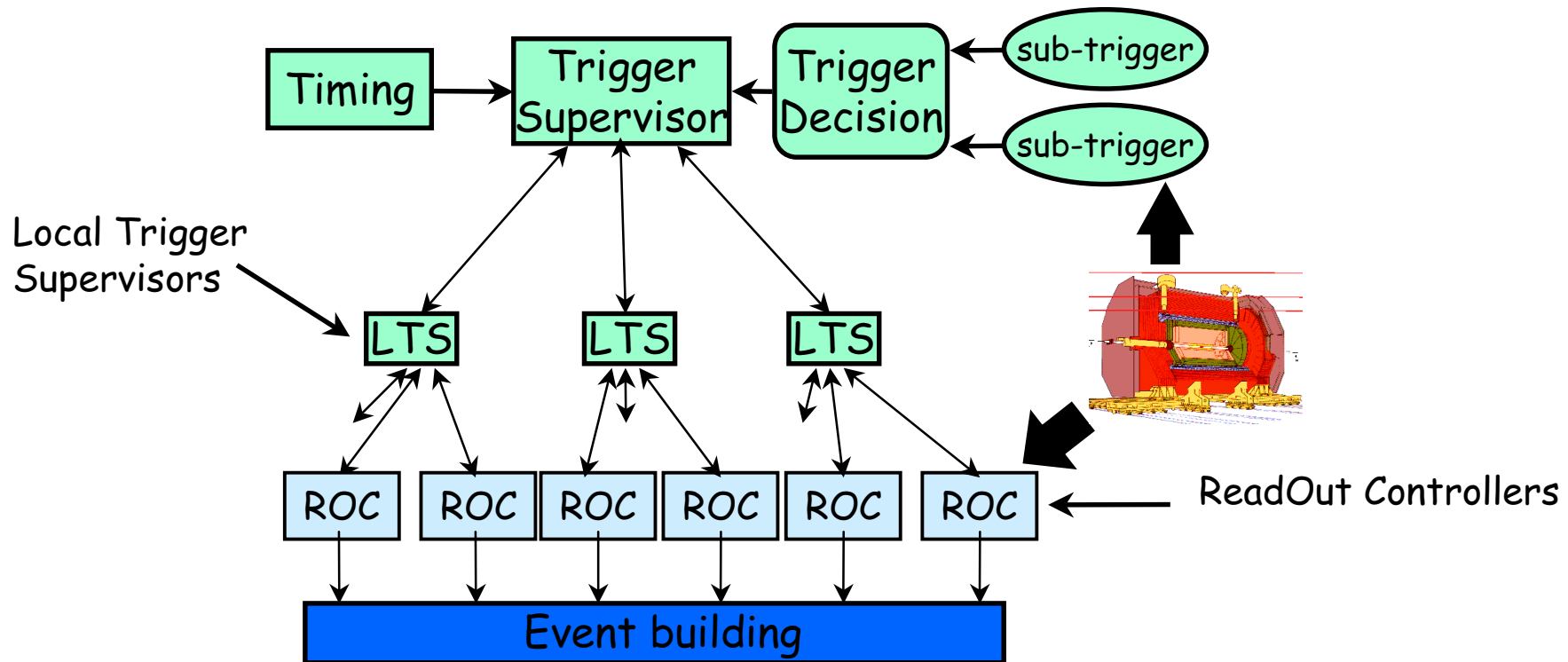
# Trigger Levels in LHCb (LHC)



- ⌘ Level-0 (4  $\mu$ s) (custom processors)
  - ⊞ High  $p_T$  for electrons, muons, hadrons
  - ⊞ Pile-up veto.
- ⌘ Level-1 (1000  $\mu$ s) (specialized procs)
  - ⊞ Vertex topology (primary & secondary vertices)
  - ⊞ Tracking (connecting calorimeter clusters with tracks)
- ⌘ Level-2 ( $\approx$ ms) (commercial processors)
  - ⊞ Refinement of the Level-1. Background rejection.
- ⌘ Level-3 ( $\approx$ ms) (commercial processors)
  - ⊞ Event reconstruction. Select physics channels.



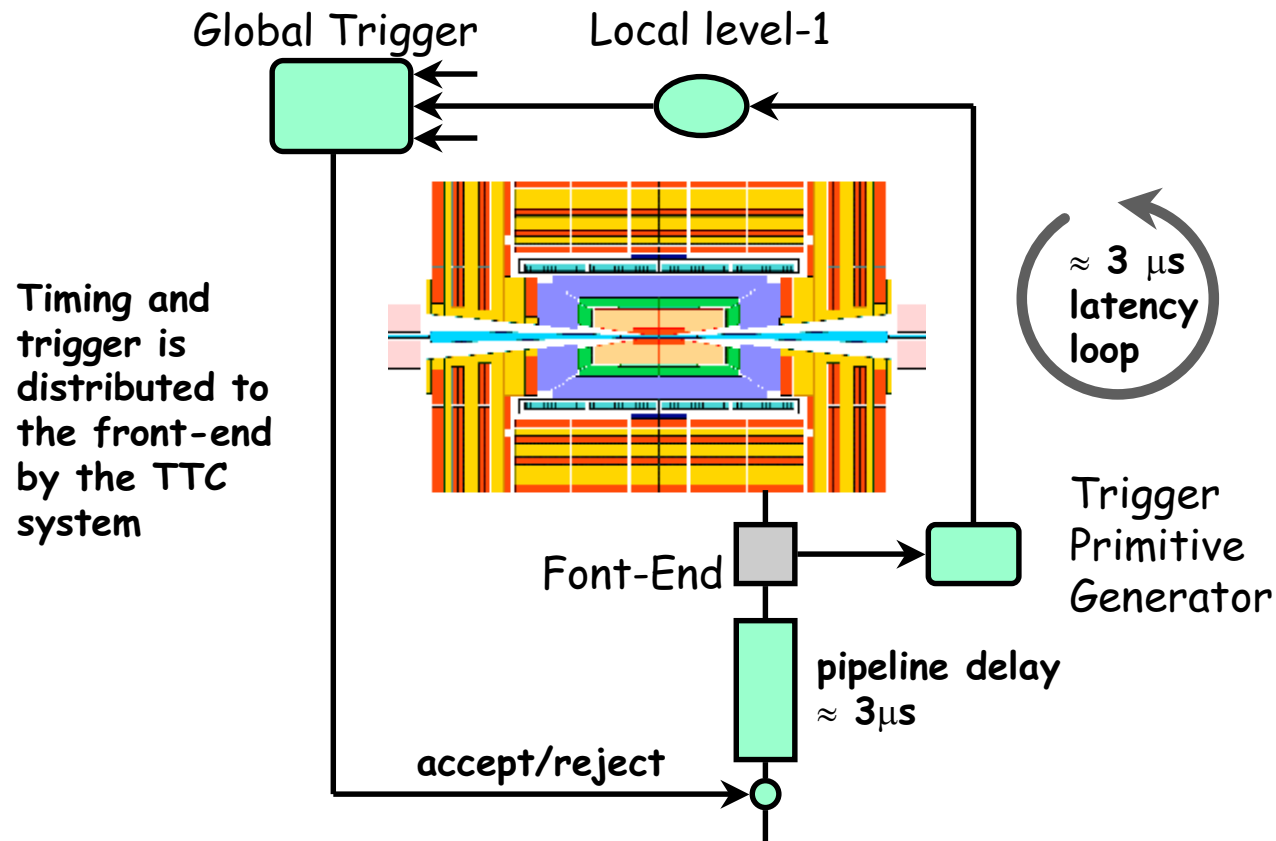
# LEP: Trigger Distribution



- ⌘ Trigger decision logic: look up tables
- ⌘ Trigger protocol (Busy, L1yes, Abort, etc.)
- ⌘ Trigger identifier can be delivered to each ROC.
- ⌘ Programmable TS and LTS to distribute trigger signals and collect them from subsets of ROCs.



# LHC: Trigger communication loop

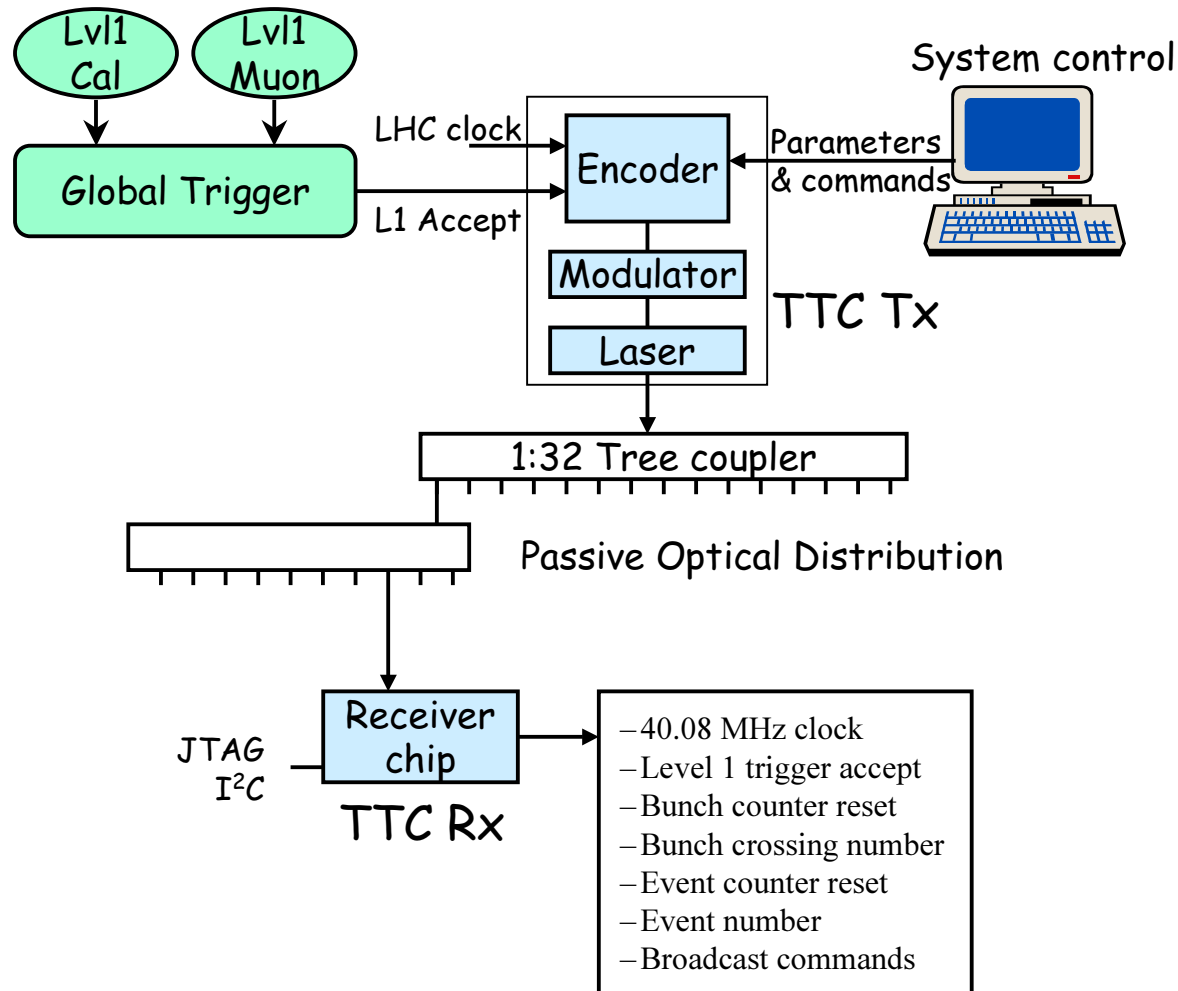


- ⌘ 40 MHz synchronous digital system
- ⌘ Synchronization at the exit of the pipeline non trivial.  
⇒ Timing calibration



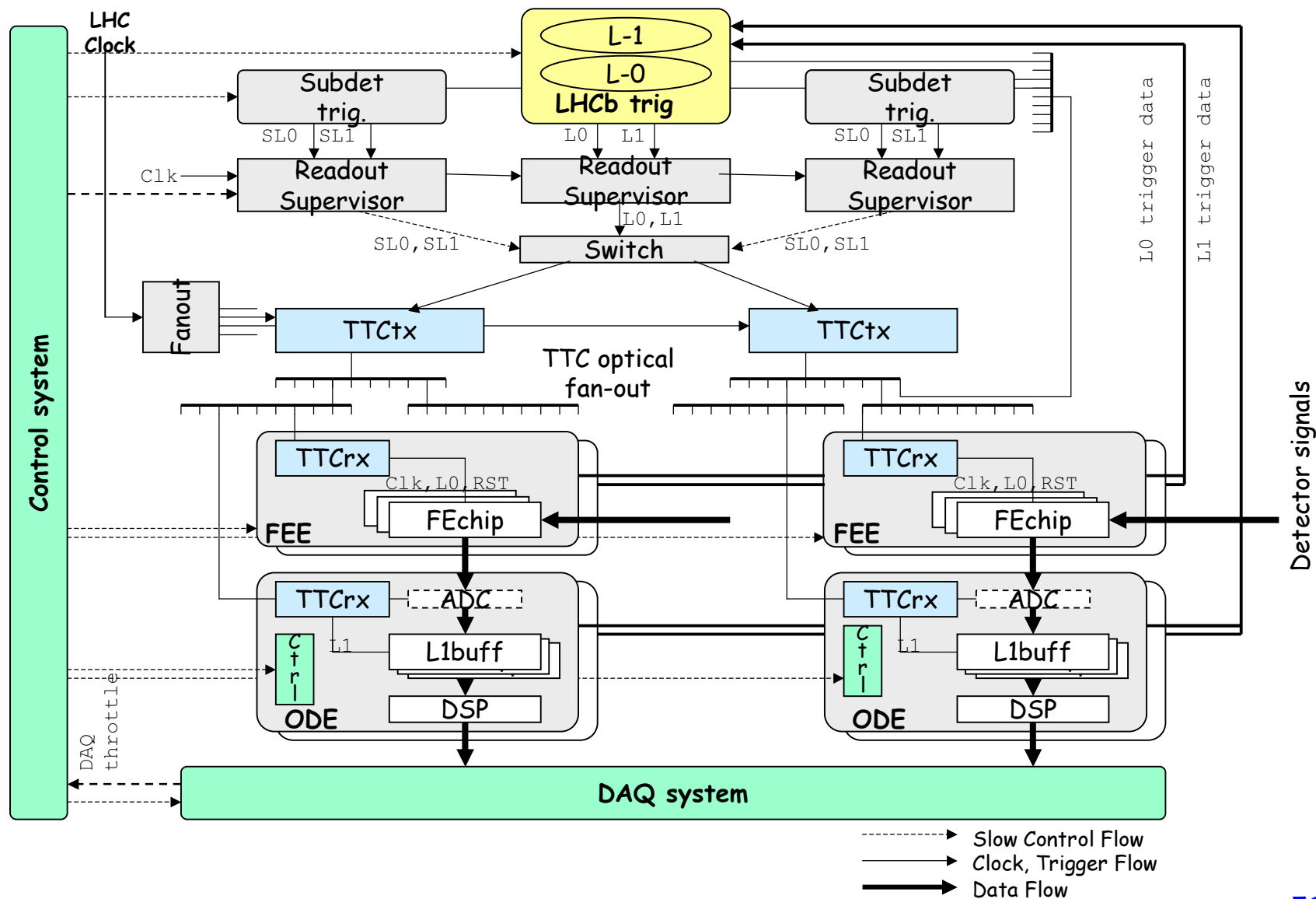
# LHC: Timing & Trigger distribution

## ⌘ The TTC system





# LHCb: Timing & Fast Control





# Data Acquisition

Clara Gaspar CERN/EP-LBC

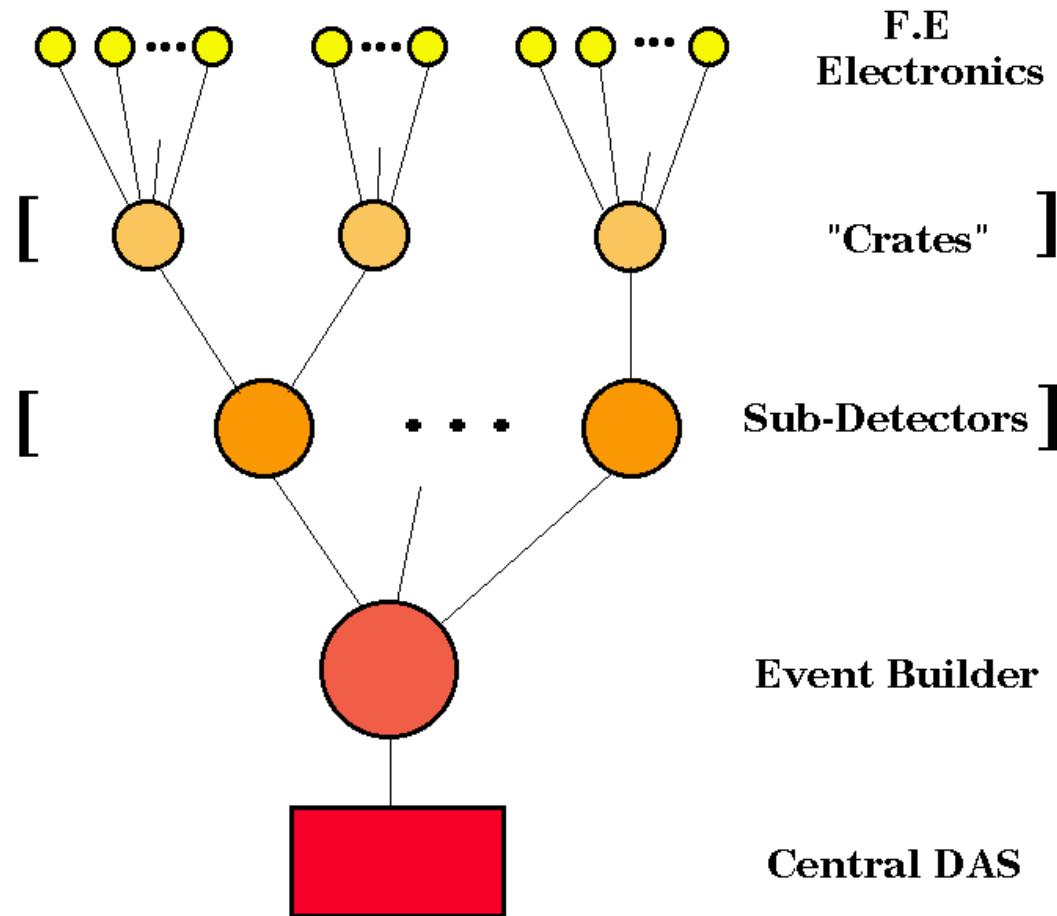


# DAQ Readout

- ⌘ Reading data from FE buffers to form a full "event" on tape:
  - ☒ (Sub-event building)
  - ☒ Event Building
  - ☒ (Processing)
  - ☒ Storage
- ⌘ Asynchronous with beam crossing
- ⌘ HW (and SW) can be common to all sub-detectors



# LEP Readout Architecture







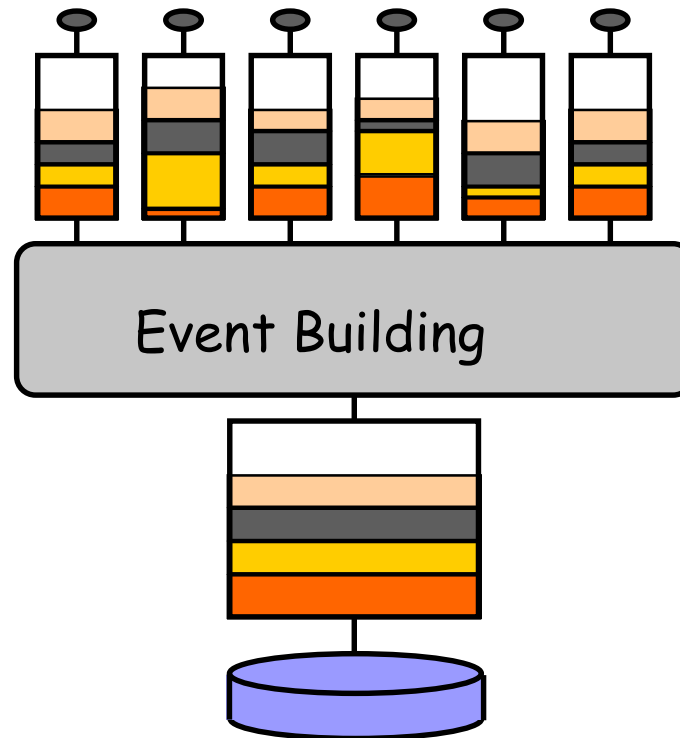
# Event Building

Data sources

Event Fragments

Full Events

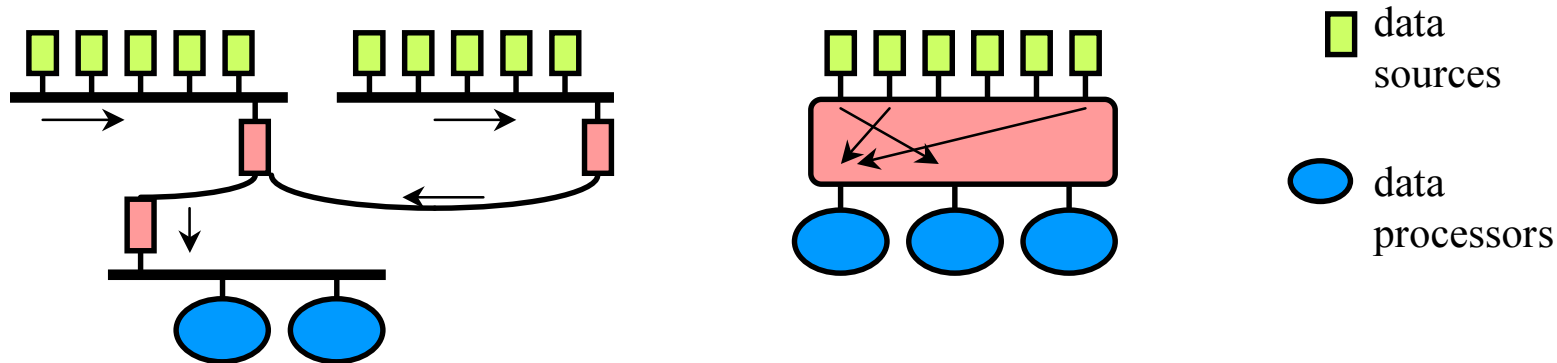
Data storage





# Readout Networks

- ⌘ Since the early 70's there have been a need for a standard for building big readout systems with many hundred thousands of electronics channels.
- ⌘ Basic components needed:
  - ⊠ FE boards (digitizers, etc), Readout controllers, Crates, Crate interconnects
- ⌘ With these components you can build networks using buses or switches.

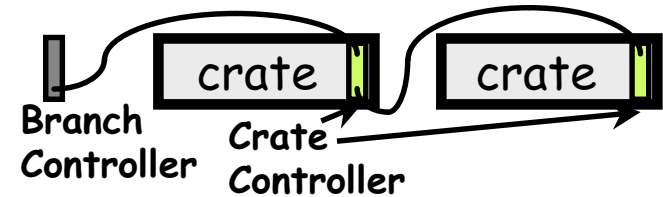




# Buses used at LEP

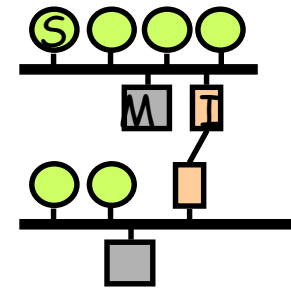
## ⌘ Camac

- Very robust system still used by small/medium size experiments
- Large variety of front-end modules
- Low readout speed (0.5-2 Mbytes/s)



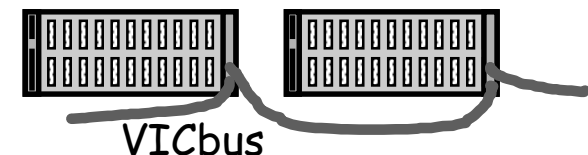
## ⌘ Fastbus

- Fast Data Transfers (Crate: 40Mbyte/s, cable: 4 Mbyte/s)
- Large Board Surface ( 50-100 electronic channels)
- Few commercial modules (mostly interconnects)



## ⌘ VME

- Large availability of modules
  - CPU boards (68k, PowerPC, Intel), Memories, I/O interfaces (Ethernet, Disk, ...), Interfaces to other buses (Camac, Fastbus,...), Front-end boards.
- Small Board Size and no standard crate interconnection
- But VICBus provides Crate Interconnects
- 9U Crates provide large board space





# Choice of the bus standard

⌘ LEP experiments had to choose a standard bus.

▣ How to choose:

- ▣ There is no truth
- ▣ There are fashions and preferences.

▣ Hybrid solutions are possible.

▣ Choices taken:

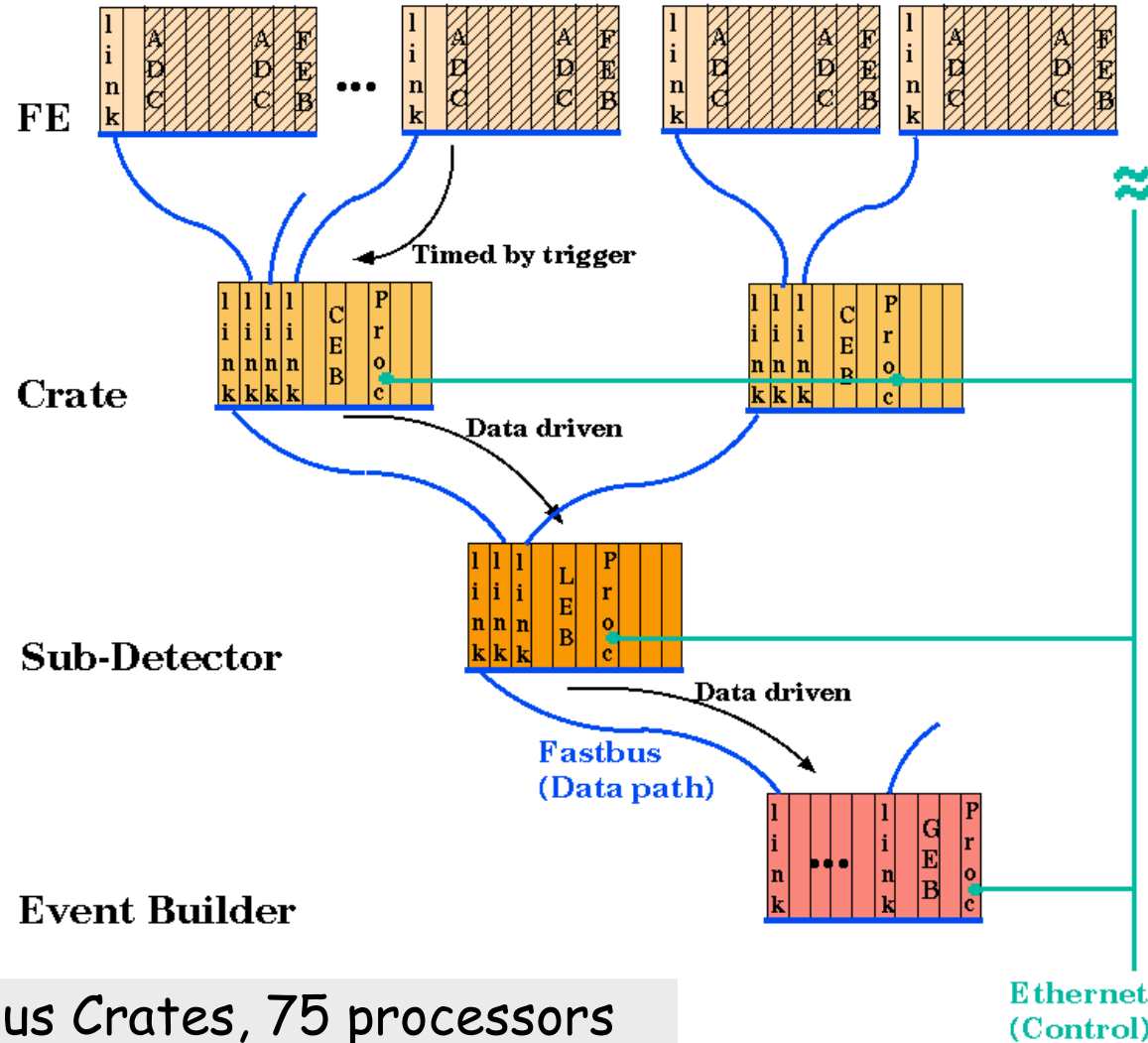
- ▣ OPAL      Camac, Fastbus, VME for Front-end, VME for Readout
- ▣ ALEPH     Fastbus for Front-end, VME for Readout
- ▣ L3         Fastbus
- ▣ DELPHI    Fastbus + few Camac for Front-end

⌘ For LHC, standard buses will not be used to readout the data (performance limitations). VME may be used for configuration and control.



# Readout in Delphi

⌘ FE data



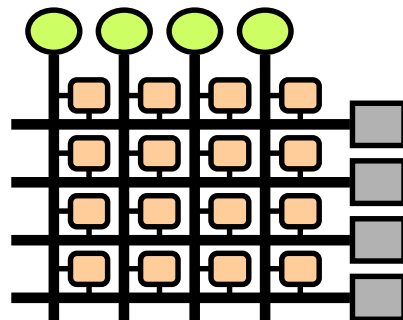
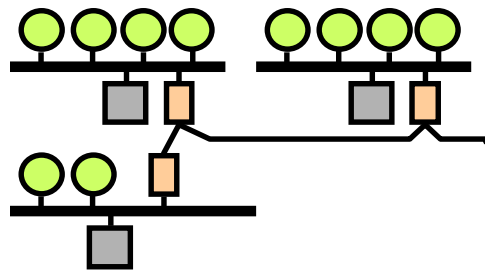
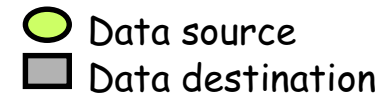
⌘ Full Event

Event Builder

⌘ In Total: 200 Fastbus Crates, 75 processors



# Event builder techniques



## ⌘ Time-shared Bus

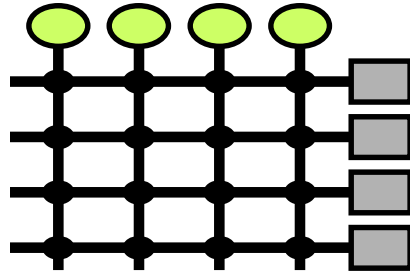
- ⊞ Most common at LEP (VME, Fastbus)
- ⊞ Bi-directional
- ⊞ Limited to the maximum throughput
- ⊞ Staged event building by independent buses in parallel (trees). No real gain, but reduces overhead.

## ⌘ Dual-port memory

- ⊞ Event fragments are written in parallel and read sequentially by the destination processor.
- ⊞ Easy to implement. Commercially available. ex. D0, OPAL (VME/VSB)



# Event builder techniques(2)



## ⌘ Cross bar switch

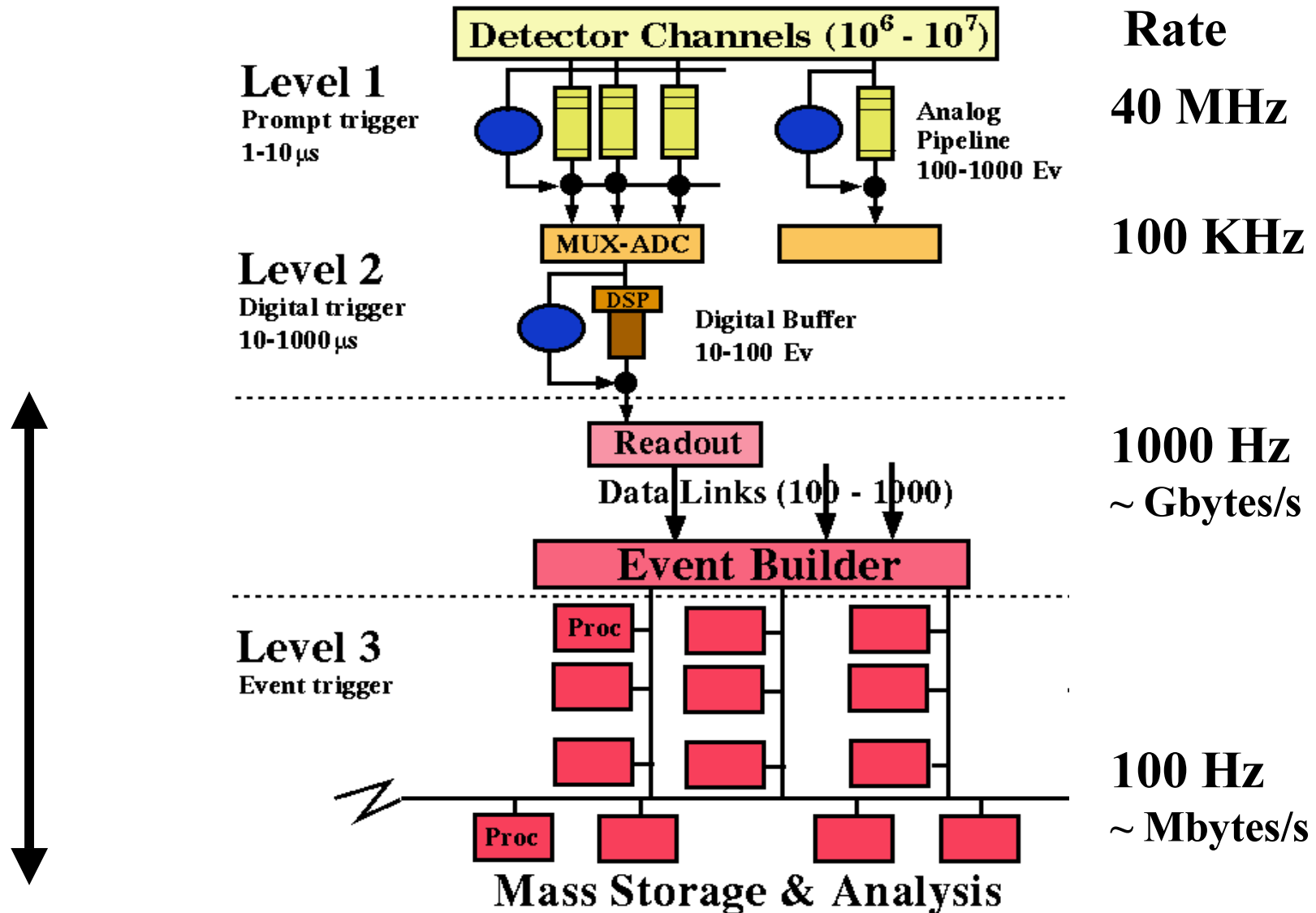
- ⊞ Complete, non blocking interconnection all inputs/all outputs.
- ⊞ Ideal bandwidth efficiency.
- ⊞  $N^2$  crosspoints.
- ⊞ Control of the path routing:
  - ⊗ External control (barrel shifter).
  - ⊗ Auto-routing (by data). Data frame protocols.

## ⌘ Switches vs. Buses

- ⊞ Total bandwidth of a Bus is shared among all the processors. Adding more processors degrades the performance of the others. In general, **Buses do not scale** very well.
- ⊞ With switches,  $N$  simultaneous transfers can co-exists. Adding more processors does not degrade performance (bigger switch). **Switches are scalable.**



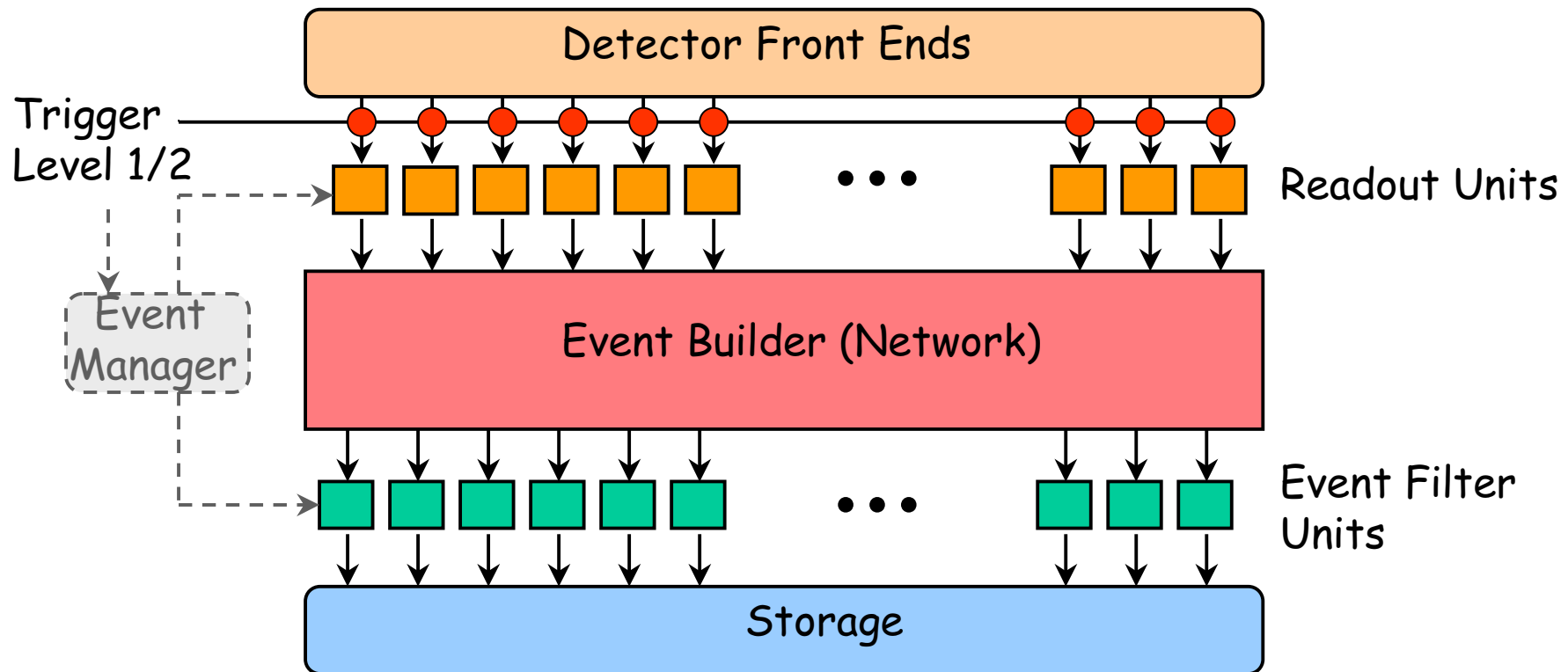
# LHC Readout







# LHC Readout Architecture





# Event Building to a CPU farm

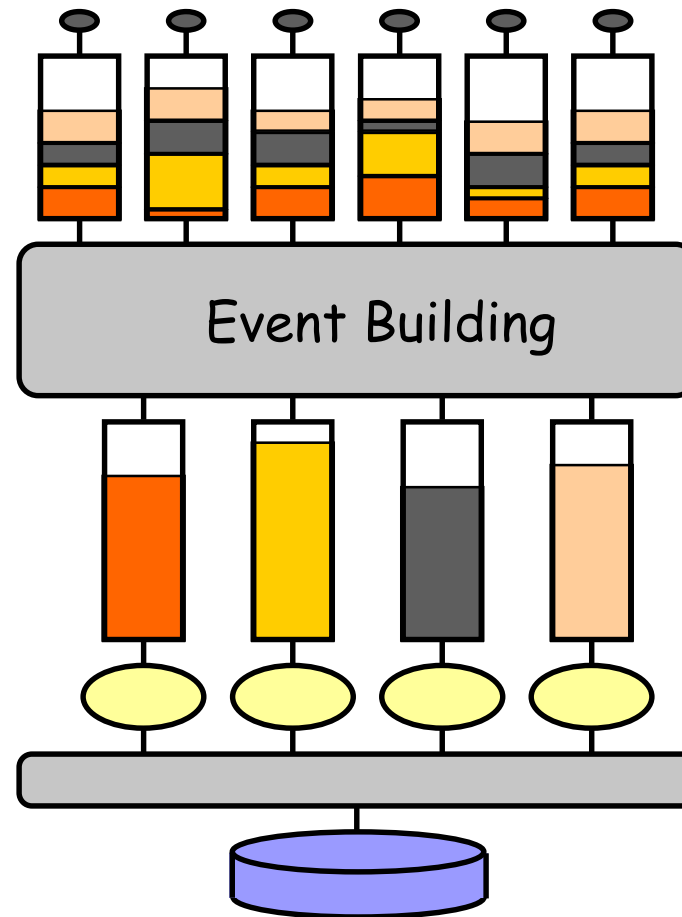
Data sources

Event Fragments

Full Events

Event filter CPUs

Data storage





# LHC Event Building Technologies

## ⌘ Industrial Digital Switching Technology

⊡ From Telecommunications and Computing Industries

⊡ Some Possibilities:

⊗ ATM (speeds up to 9.6 Gb/s)

- 53 byte cells, expensive

⊗ Fiber Channel (1 Gb/s)

- Connection oriented -> more overhead

⊗ Myrinet (2.5 Gb/s)

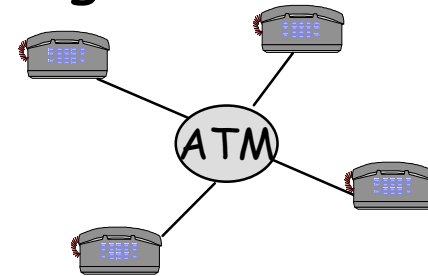
- Unlimited message size, cheap switches
- No buffer in the switch -> less scalable

⊗ SCI - Scalable Coherent Interface (4 Gb/s)

- Memory mapped IO, 64 byte messages

⊗ Gigabit Ethernet (specification for 10 Gb/s)

- Cheaper, ethernet protocol





# Event Building protocol

## ⌘ Push protocol

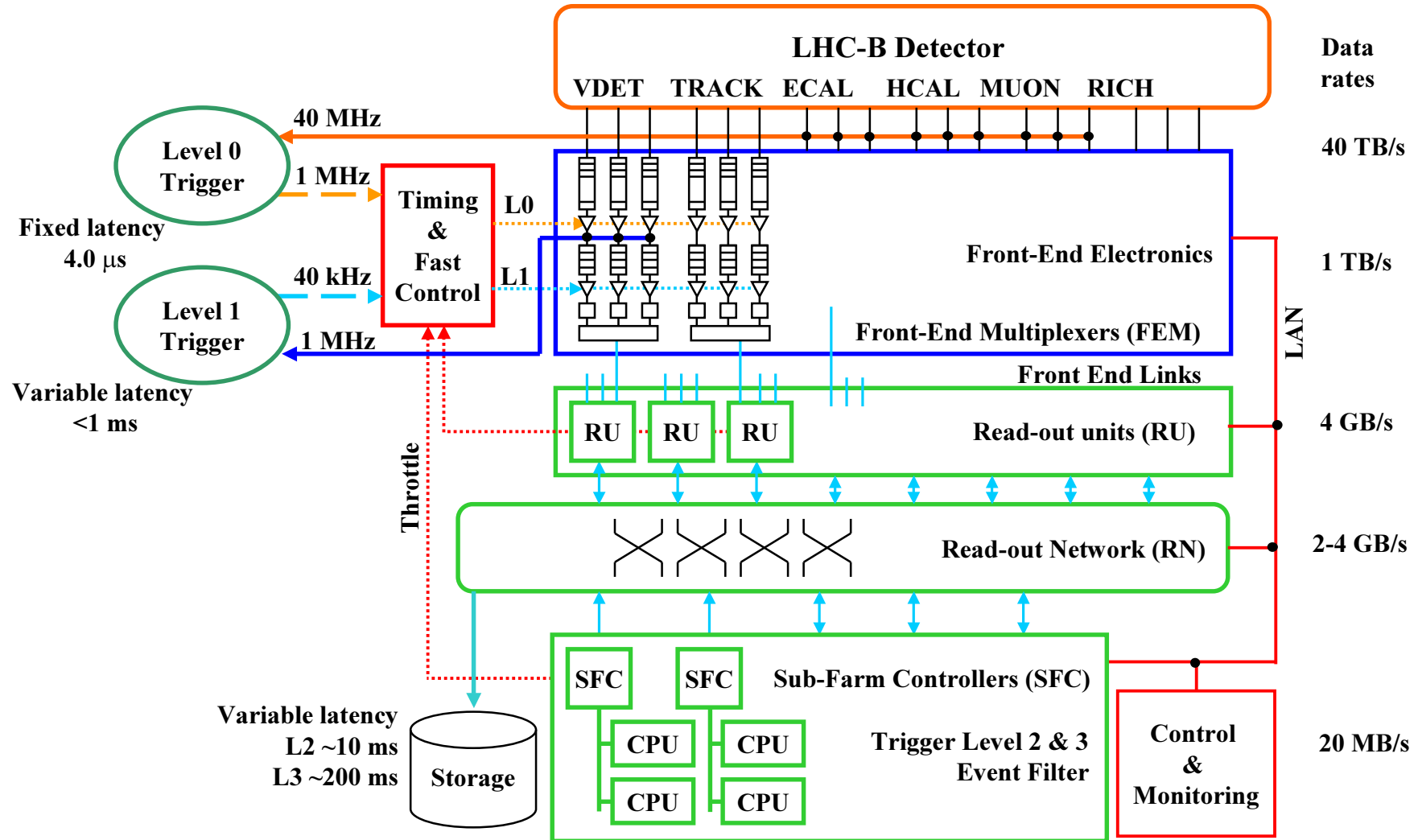
- ⊞ The data is pushed to the destinations by the sources.
- ⊞ The source needs to know the destination address.
- ⊞ It is assumed that there is sufficient data buffer at the destination.
- ⊞ There is no possibility of re-transmitting a fragment of event.
- ⊞ The protocol is simple.

## ⌘ Pull protocol

- ⊞ The data in the sources is pulled from the destinations.
- ⊞ Only buses can implement a pure pull protocol.
- ⊞ Sources need in any case indicate to the destinations when data is ready (interrupt??).
- ⊞ Destinations can re-read the event fragment.
- ⊞ Destinations need to indicate when the transfer is finished to free memory in the source.
- ⊞ The protocol is heavier.

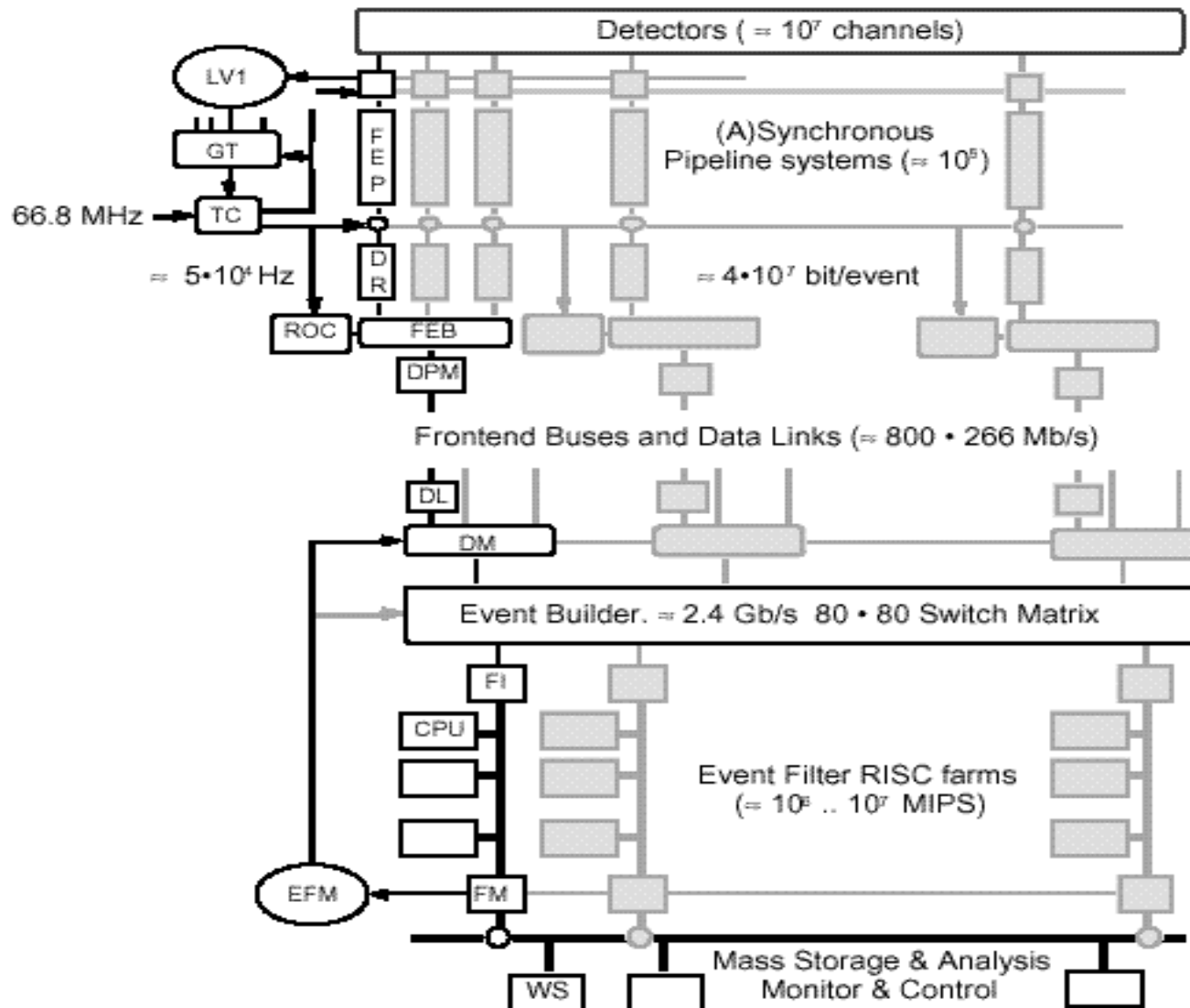


# LHCb DAQ (Push Protocol)





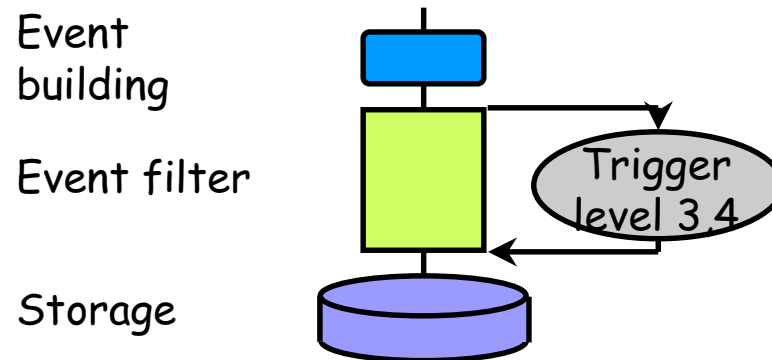
# CMS DAQ (Pull Protocol)





# Event Filters

## ⌘ Higher level triggers (3, 4, ...)



- ⌘ LHC experiments can not afford to write all acquired data into mass-storage. -> Only useful events should be written to the storage
- ⌘ The event filter function selects events that will be used in the data analysis. Selected physics channels.
- ⌘ The algorithms are usually high level (dealing with physics quantities) therefore implying a full or partial event reconstruction.

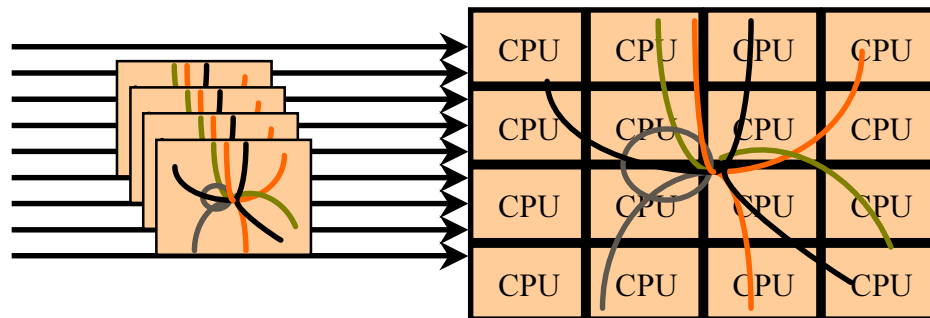


# Parallel versus Farm processing

⌘ A single CPU can not provide the required processing power (especially in LHC)

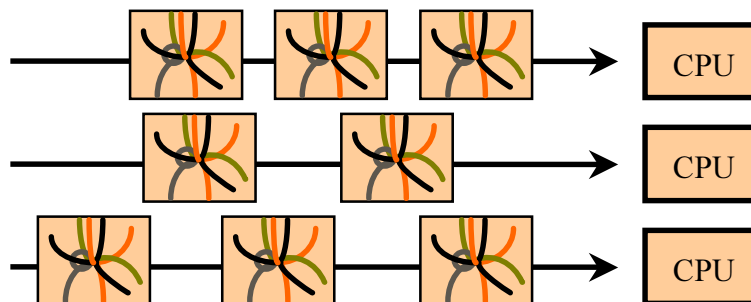
**Parallel**

Massive parallel processing



⌘ Low latency, complex, expensive.

**Farm**



⌘ Larger latency, simple, scalable.  
⌘ Exploits the fact that each event is independent of the next one.  
⌘ Use of commercially (commodity items) available processors





# Data Storage

## ⌘ Data logging needs:

### ▣ LEP experiments:

- ▣ Storage bandwidth  $< 1$  MB/s
- ▣ Data is stored on disk and then copied to magnetic tape.
- ▣ Data set: 500 GB/year

### ▣ LHC experiments:

- ▣ Storage bandwidth:  $\gg 100$  MB/s (general purpose)
- ▣ Expected data set:  $\gg 100$  TB/year

## ⌘ Hierarchical Storage System (HSS)

- ▣ File system where files are migrated to the mass storage system (tape robot) automatically when not in use and retrieved automatically when needed.
- ▣ Cache system at the level of files.



# Data Storage (2)

## ⌘ How the data is organized in the storage

- ⊞ The LEP experiments organized the data in events and banks within the events written sequentially.
- ⊞ Studying the data access patterns on the analysis programs (which data are more often used and which are hardly ever accessed) should give us an idea on how to store the data.
- ⊞ Data is written once and read many times. Therefore the optimization should go into the read access.

## ⌘ Organizing the event data store as a huge distributed database

- ⊞ We could profit from the database technology to have fast access to specific data. Sophisticated data queries to do event selection.
- ⊞ Problems: How to populate the database at the required throughput? Database backup? Schema evolution? ...



# Configuration, Control and Monitoring

Clara Gaspar CERN/EP-LBC



# System Performance

$$\text{Total Efficiency} = \frac{\text{\# of interesting events on tape}}{\text{\# of interesting events produced}}$$

$$= \text{Trigger efficiency} * \text{Dadtime} * \text{DAQ efficiency} * \text{Operation efficiency}$$

DAQ system not running

Detector problems: background, data quality

weighted by the Luminosity

- ⌘ All the factors have equal importance. Therefore all of them need to be optimized.
- ⌘ The performance needs to be monitored in order to detect problems and point where there is a need for improvement.



# Trigger & DAQ Control

## ⌘ Run Control

### ☒ Configuration

- ☒ Type of RUN, loading of parameters, enabling/disabling parts of the experiment

### ☒ Partitioning

- ☒ Ability to run parts of the experiment in stand-alone mode simultaneously

### ☒ Error Reporting & Recovery

### ☒ Monitoring

### ☒ User Interfacing



# Experiment Control (ECS)

⌘ In charge of the Control and Monitoring of:

▣ Data Acquisition and Trigger (Run Control)

▣ FE Electronics, Event building, EFF, etc.

▣ Detector Control (Slow Control)

▣ Gas, HV, LV, temperatures, ...

▣ Experimental Infrastructures

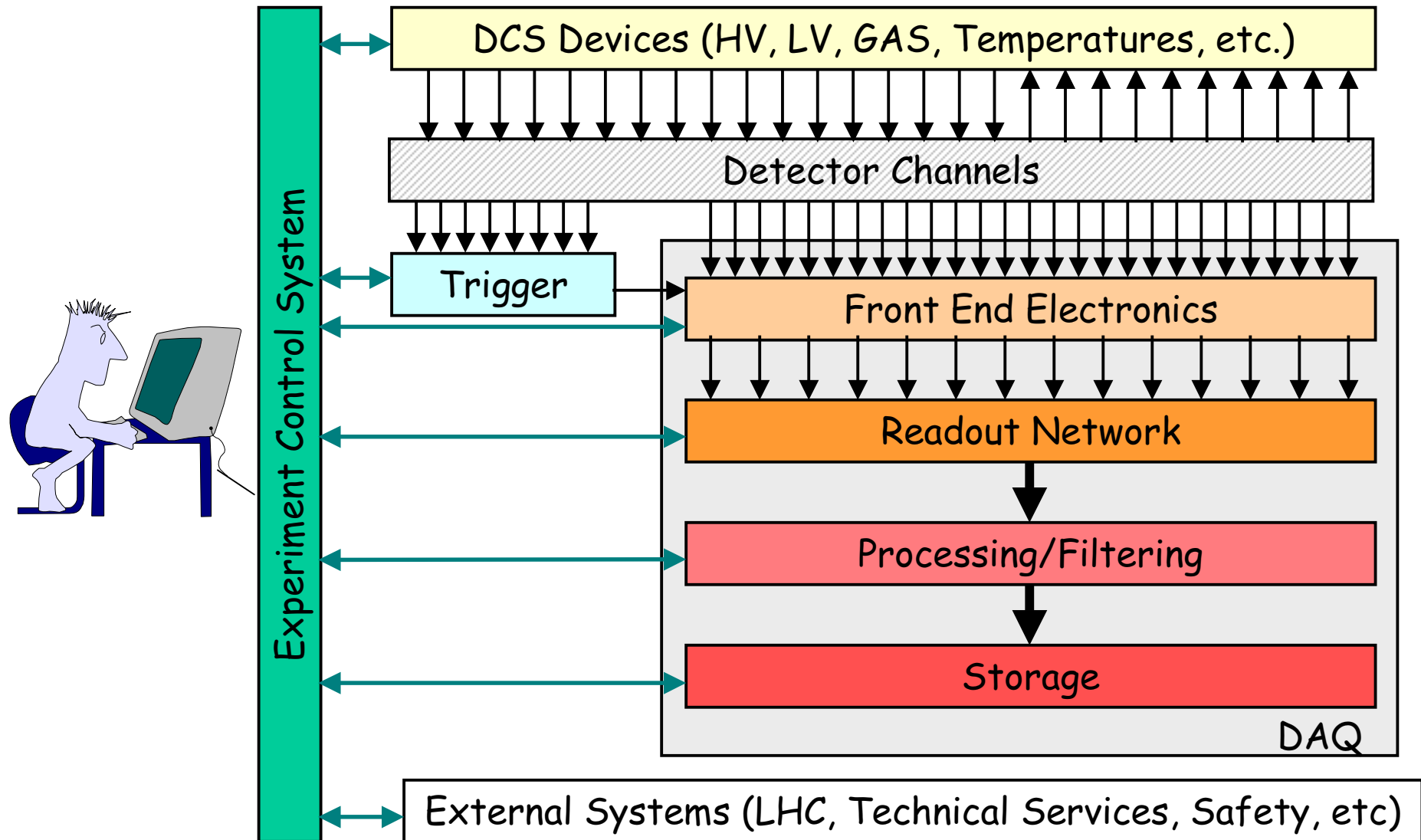
▣ Cooling, ventilation, electricity distribution, ...

▣ Interaction with the outside world

▣ Magnet, accelerator system, safety system, etc.



# ECS Scope





# ECS Requirements

## ⊞ Integrate the different activities

- ⊞ Such that rules can be defined (ex: Stop DAQ when SC in Error)

## ⊞ Allow Stand-alone control of sub-systems

- ⊞ For independent development and concurrent usage.

## ⊞ Automation

- ⊞ Avoids human mistakes and speeds up standard procedures

## ⊞ Easy to operate

- ⊞ Two to three operators (non-experts) should be able to run the experiment.

## ⊞ Scalable & Flexible

- ⊞ Allow for the integration of new detectors

## ⊞ Maintainable

- ⊞ Experiments run for many years





# Experiment Control

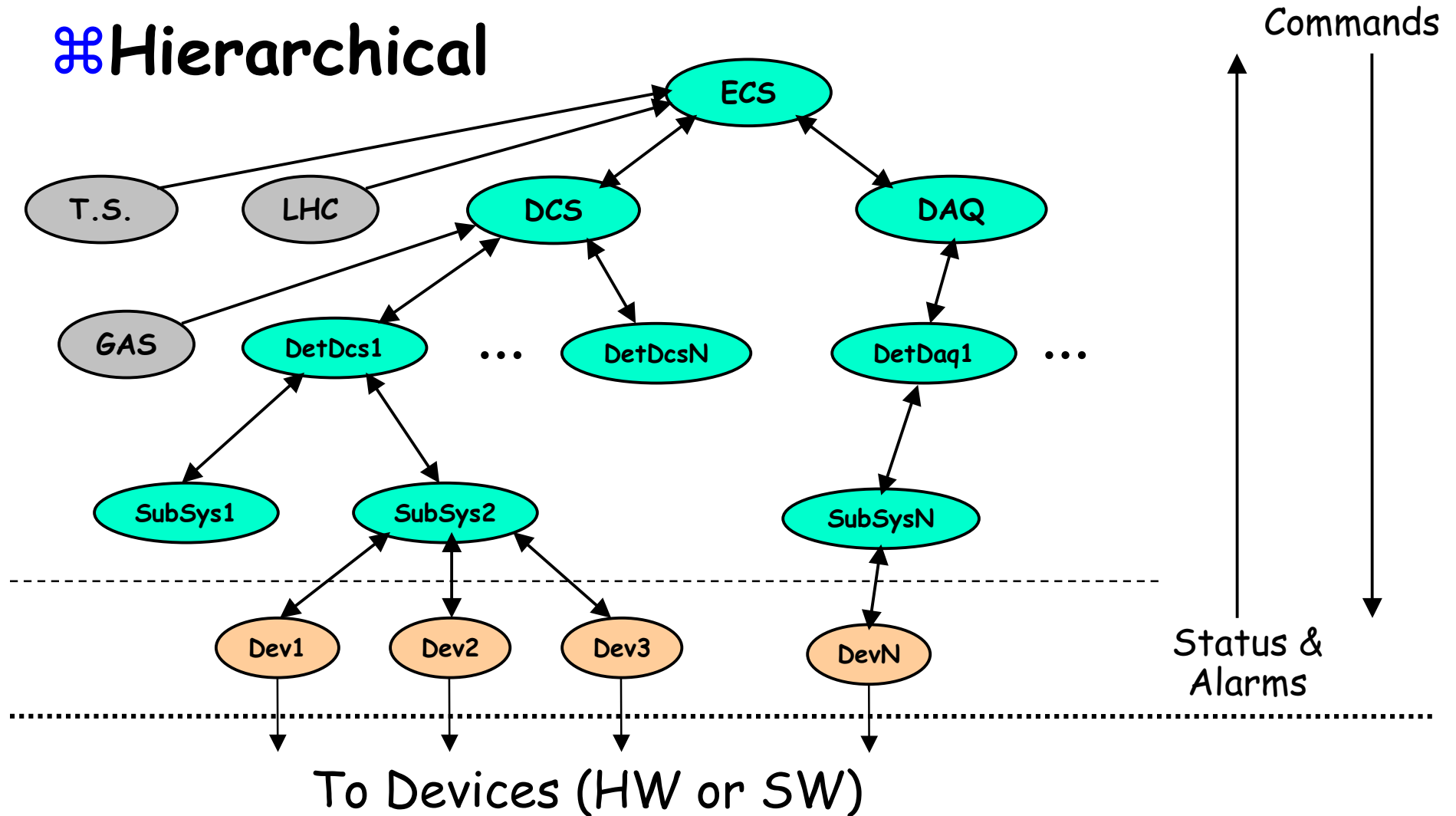
## ⌘ Keyword: Homogeneity

- ☒ A Common Approach in the design and implementation of all parts of the system:
  - ☒ Facilitates inter-domain integration
  - ☒ Makes it easier to use:
    - Standard features throughout the system (ex: partitioning rules)
    - Uniform Look and Feel
  - ☒ Allows an easier upgrade and maintenance
  - ☒ Needs less manpower



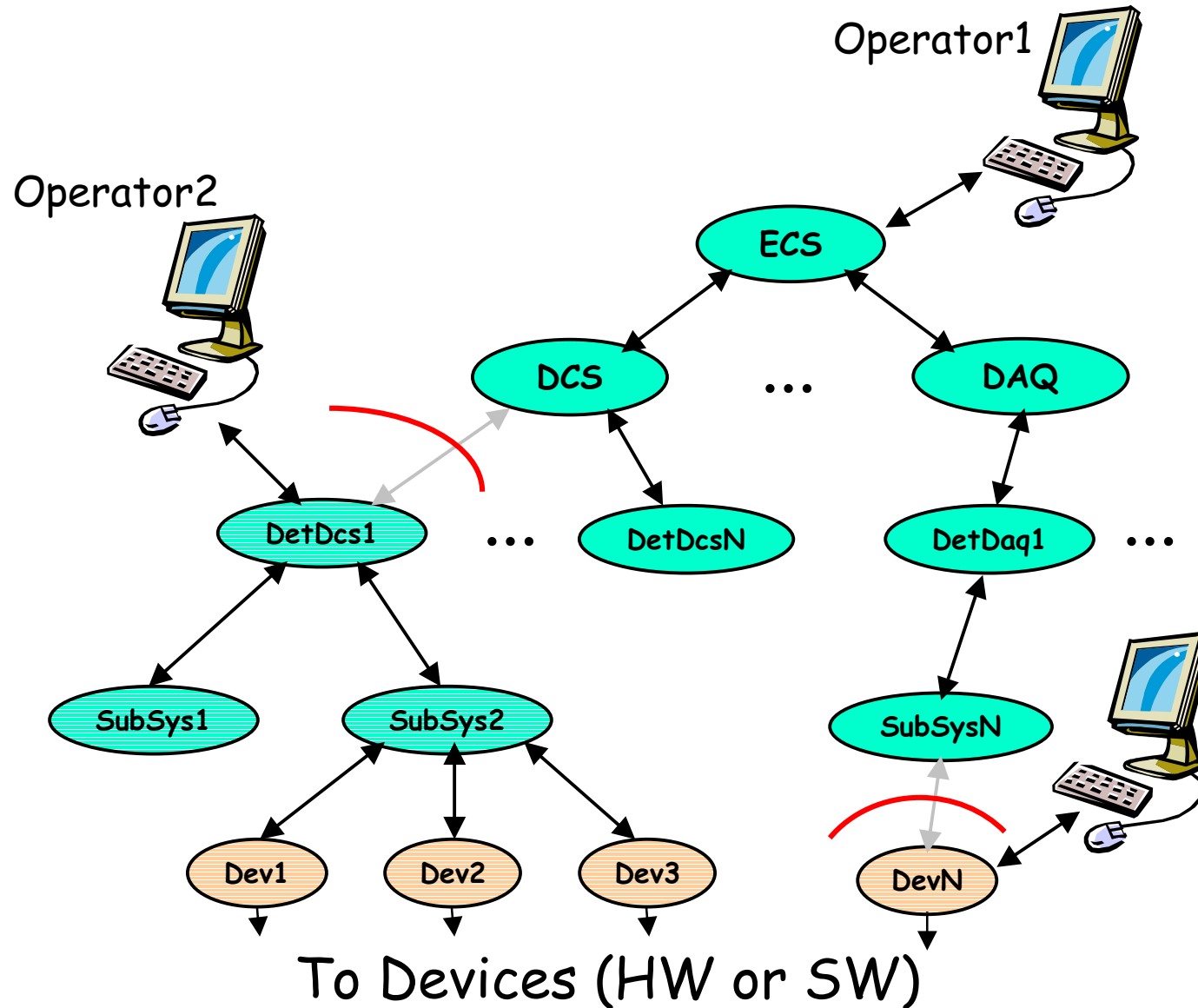
# Control System Architecture

## ⌘ Hierarchical





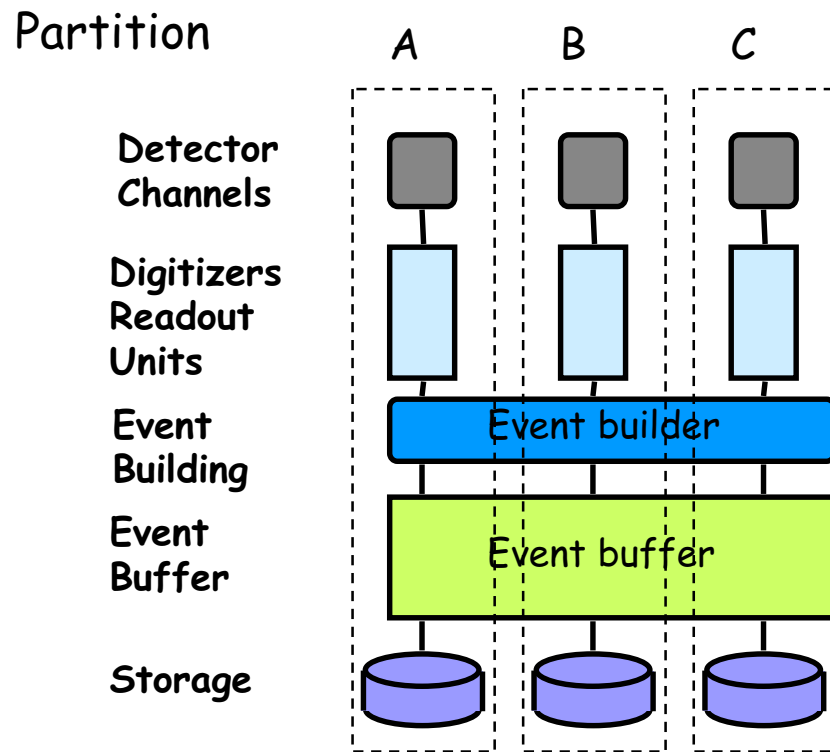
# Partitioning





# Partitioning (2)

⌘ Partitioning the DAQ imposes strong constraints in the software and hardware. Some resources are shareable and some not



- ⌘ A trigger source (local trigger or trigger supervisor)
- ⌘ A set of readout units (data sources) attached to the detector parts which we want to use.
- ⌘ Some bandwidth on the event building.
- ⌘ Processing resources.
- ⌘ Data storage.



# System Configuration

⌘ All the components of the system need to be configured before they can perform their function.

- ☒ Detector channels: thresholds, calibration constants need to be downloaded.
- ☒ Processing elements: programs and parameters.
- ☒ Readout elements: destination and source addresses. Topology configuration.
- ☒ Trigger elements: Programs, thresholds, parameters.

⌘ Configuration needs to be performed in a given sequence.



# System Configuration (2)

## ⌘ Databases

- ☒ The data to configure the hardware and software is retrieved from a **database system**. No data should be hardwired on the code (addresses, names, parameters, etc.).

## ⌘ Data Driven Code

- ☒ Generic software should be used wherever possible (it is the data that changes)
- ☒ In Delphi all sub-detectors run the same DAQ & DAQ control software



# System Configuration (3)

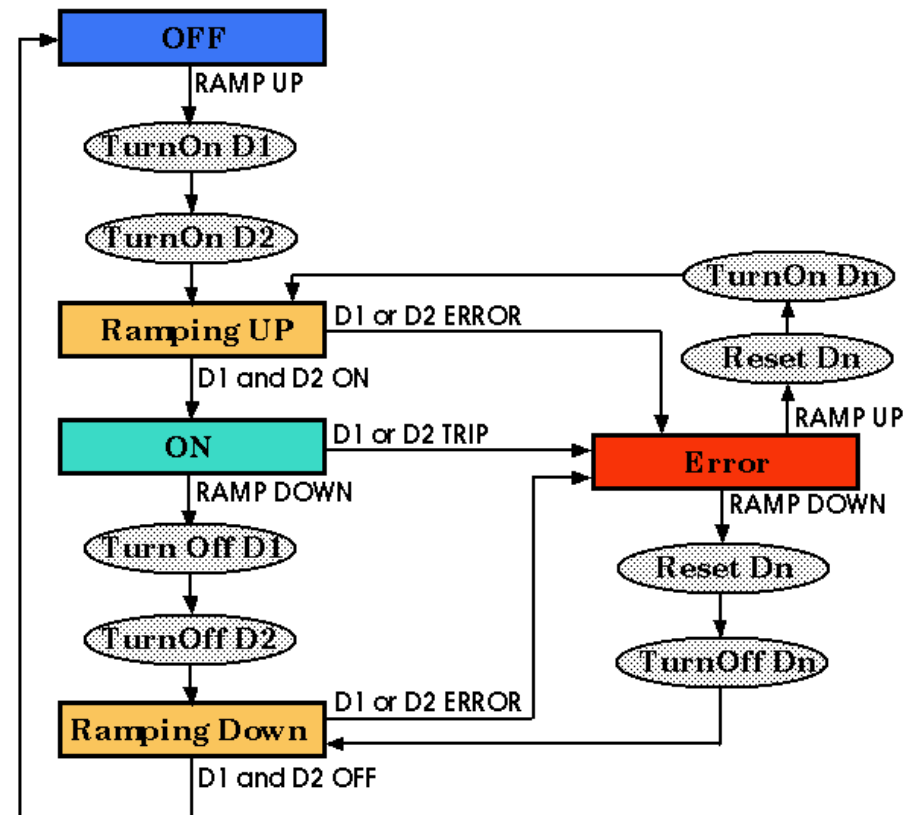
## ⌘ Finite State Machine methodology

⊞ Intuitive way of modeling behaviour, provide:

- ⊞ Sequencing
- ⊞ Synchronization

⊞ Some FSM tools allow:

- ⊞ Parallelism
- ⊞ Hierarchical control
- ⊞ Distribution
- ⊞ Rule based behaviour





# Automation

## ⌘ What can be automated

- ☑ Standard Procedures:

  - ☒ Start of fill, End of fill

- ☑ Detection and Recovery from (known) error situations

## ⌘ How

- ☑ Expert System (Aleph)

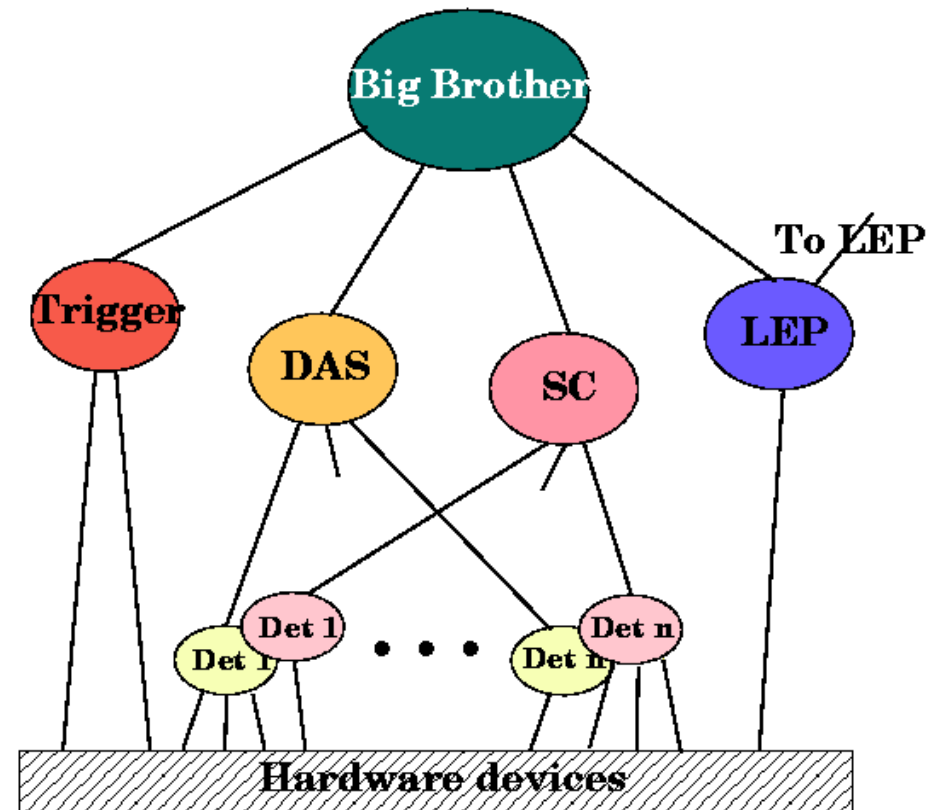
- ☑ Finite State Machines (Delphi)





# Delphi Control System

- ⌘ Hierarchical
- ⌘ Fully Automated
- ⌘ Homogeneous
  - ⊞ One single control mechanism
  - ⊞ One single communication system
  - ⊞ One user interface tool





# Monitoring

## ⌘ Two types of Monitoring

### ▣ Monitor Experiment's Behaviour

- ▣ Automation tools whenever possible
- ▣ Good User Interface

### ▣ Monitor the quality of the data

- ▣ Automatic histogram production and analysis
- ▣ User Interfaced histogram analysis
- ▣ Event displays (raw data)



# Run Control U.I.

Delphi DAS Status

TRIGGER DAS DATAFLOW LEP BACKGROUND FASTBUS SC RUN REL. SLOW CONTROL

Inst. T2 Rate: 3.36 Hz  
Inst. Livetime: 98.27 %

RUN\_IN\_PROGRESS

OK

PHYSICS  
Luminosity: 34.87

LOW  
Bg St: 0.87

OK

READY

READY  
DISALLOW\_CHANGE

### DAS

Run Status

View **RUN IN PROGRESS** 😊

Run Number : 111817

Run Time : 01:34:42

Triggers : 17059

File Number : 7

File Triggers : 1190

T3 Rate: 2.85

File Status (%): 27.3

Update Time: 27-JUN-2000 19:57:33

Big Brother: DATA TAKING  
Fill 7576 103.4 GeV

Auto Pilot: ON

Option Status: PHYSICS RUN  
T4 ON  
LOGGING

### TRIGGER

Trigger Control: READY

Trigger Mode: PHYSICS

BCO/RF Type: LEP

Nr. of Bunches: 4

Inst. T2 rate: 3.36 (Hz)

Inst. live time: 98.3 (%)

Run T2 rate: 4.18 (Hz)

Fill live time: 97.1 (%)

### Detector Status

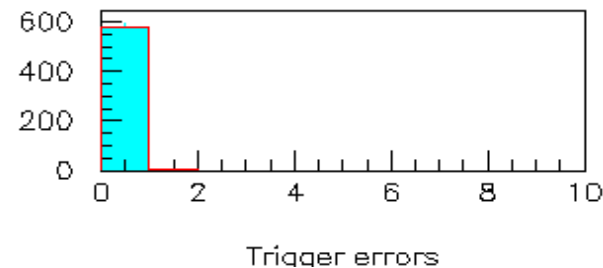
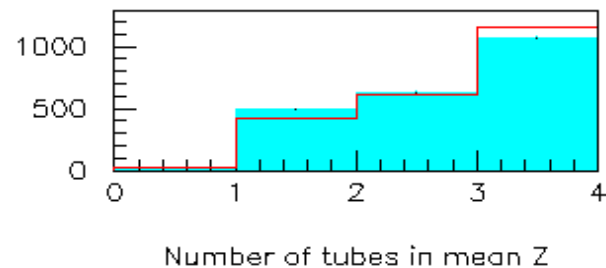
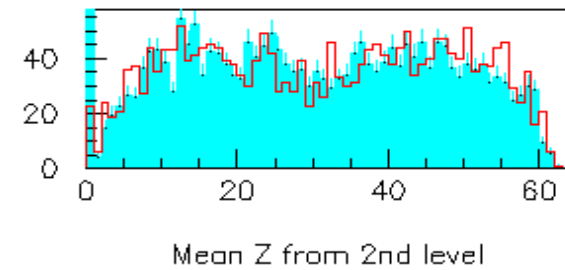
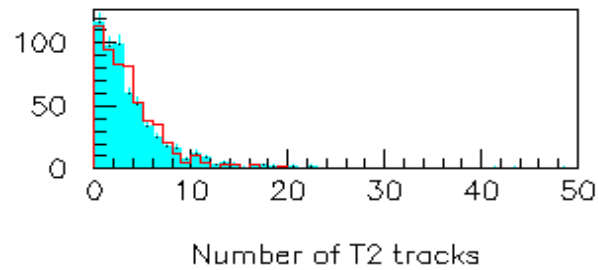
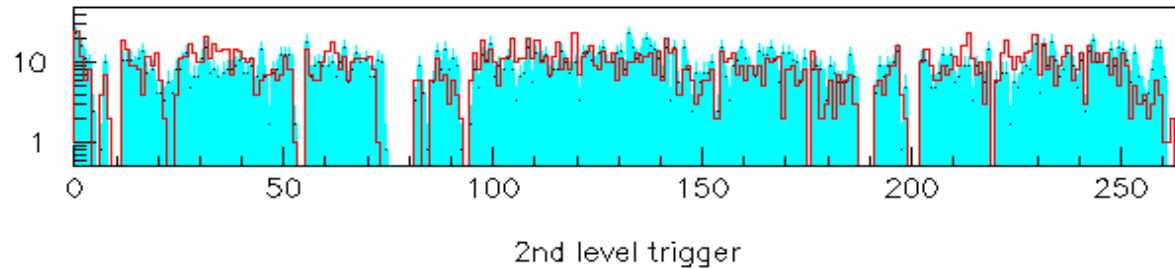
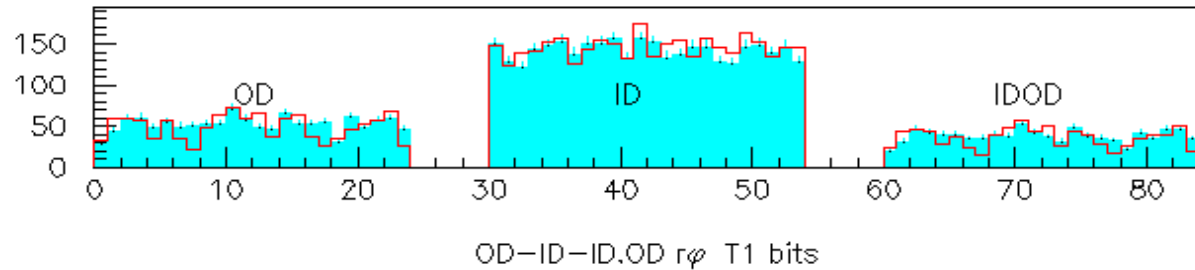
TP	VD	ID	TPC0	TPC1	STC	OD	FCH	EMF	HAC
HPC0	HPC1	MUON	MUS	RIB0	RIB1	RIF0	RIF1	TOF	VSAT

### History

```
--- 27-JUN-2000 18:27:34.31: Executing Command: START_NEW_RUN /GES executing BEGIN
--- 27-JUN-2000 18:27:50.59: Executing Command: GO /GES executing CONTINUE
```



# Histogram Presenter





# LHC Control Systems

## ⌘ Based on Commercial SCADA Systems (Supervisory Control and Data Acquisition)

### ☒ Commonly used for:

- ☒ Industrial Automation
- ☒ Control of Power Plants, etc.

### ☒ Providing:

- ☒ Configuration Database and Tools
- ☒ Run-time and Archiving of Monitoring Data including display and trending Tools.
- ☒ Alarm definition and reporting tools
- ☒ User Interface design tools



# Concluding Remarks

- ⌘ Trigger and Data Acquisition systems are becoming increasingly complex as the scale of the experiments increases. Fortunately the advances being made and expected in the technology are just about sufficient for our requirements.
- ⌘ Requirements of telecommunications and computing in general have strongly contributed to the development of standard technologies and mass production by industry.
  - ⊞ Hardware: Flash ADC, Analog memory, PC, Helical scan recording, Data compression, Image processing, Cheap MIPS, ...
  - ⊞ Software: Distributed computing, Integration technology, Software development environment, ...
- ⌘ With all these off-the-shelf components and technologies we can architect a big fraction of the new DAQ systems for the LHC experiments. Customization will still be needed in the front-end.
- ⌘ It is essential that we keep up-to-date with the progress being made by industry.