The TDAQ System of the MEG Experiment

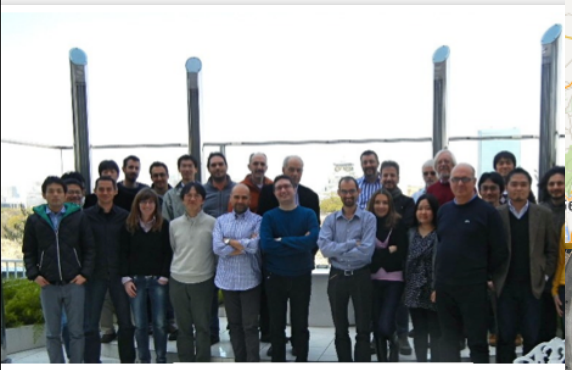
- The MEG experiment
- requirements for the TDAQ from physics
  - DAQ choice
  - requirements for the trigger
- An FPGA based trigger
  - algorithm implementation
- TDAQ efficiency considerations

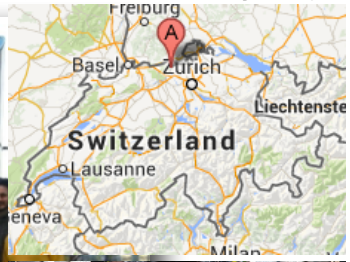
PAUL SCHERRER INSTITUT
PSI

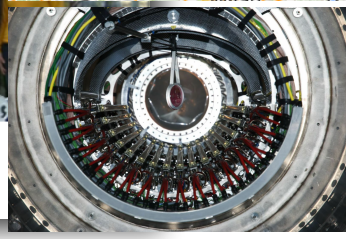
Luca Galli, PSI e INFN Pisa

INFN

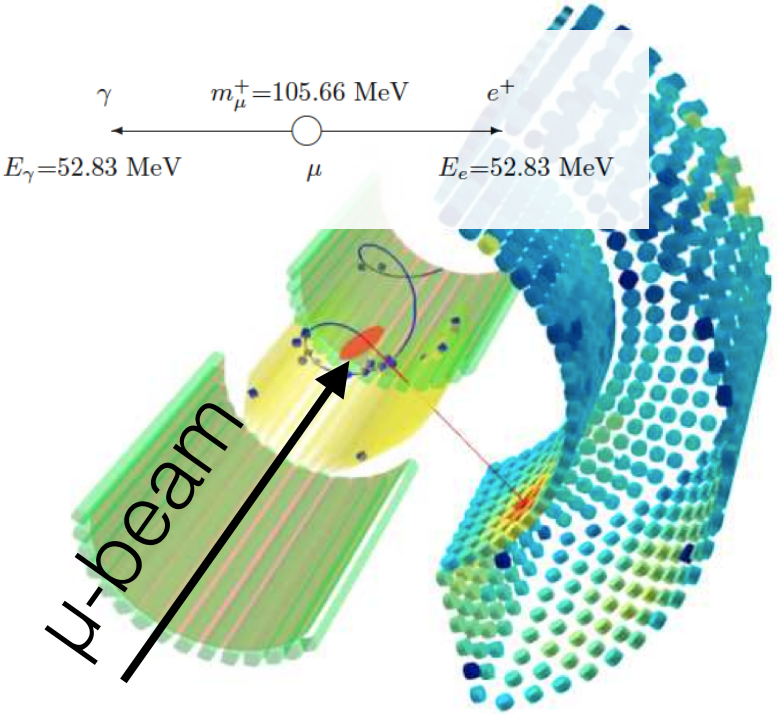
Very few words about the MEG experiment





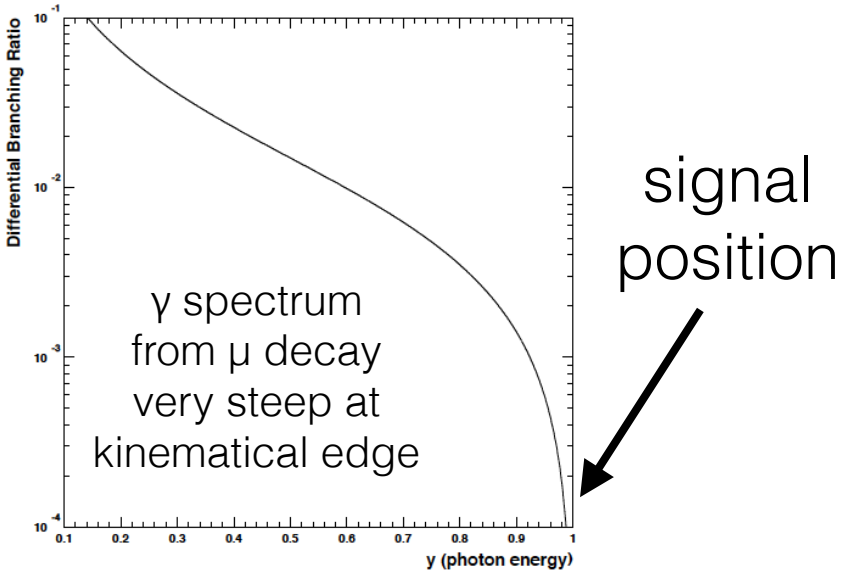


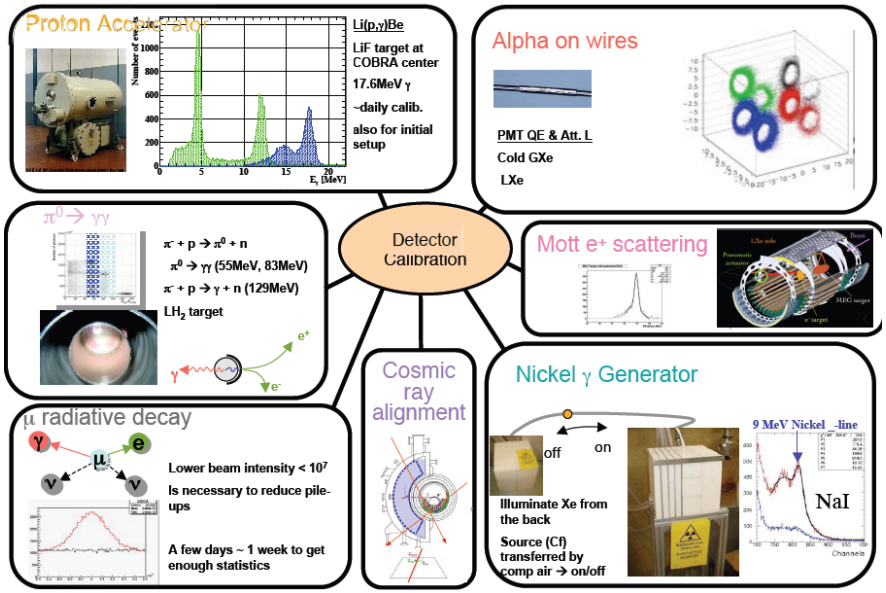
- "Tiny" collaboration
- Search μ->eγ with 5x10^(-13) sensitivity on the process @Paul Scherrer Institut, Villigen Switzerland
  - *prohibited in SM -> new physics?/! (SUSY?)*



- positive μ-beam stopped in a thin target (3x10^7 μ/sec)
  - *positron detector*
    - non-uniform magnetic field COBRA to bend positrons
    - tracking with drift chamber modules
    - timing with plastic scintillator slabs read by PMTs (Timing Counter)
  - *photon detector*
    - Liquid Xenon calorimeter read with PMTs

Requirements to the TDAQ

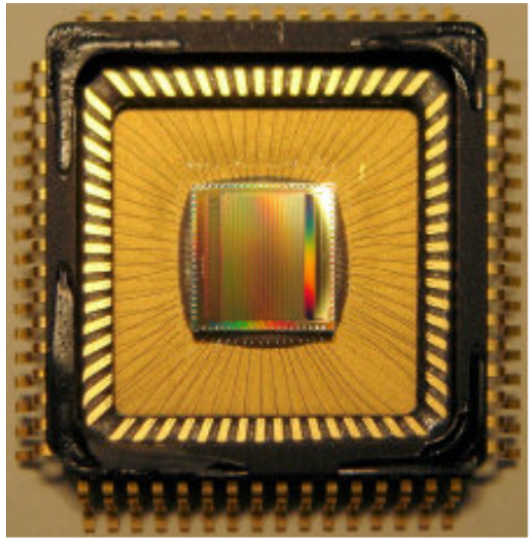


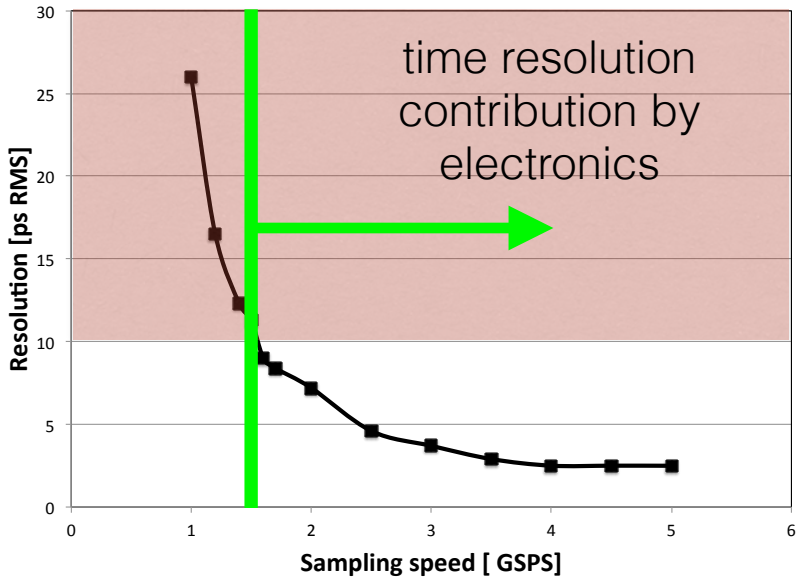


- A precision experiment demands for:
  - *statistics -> high beam intensity (and associated raw event rate and detector occupancy)*
    - electronics to resolve any possible pile-up in detectors
      - *use of waveform digitisers, no TDC and QDC*
  - *resolution -> background suppression*
    - development of detector with unprecedented resolutions at signal energy (52.8 MeV)
      - *high precision electronics for charge and time measurements*
  - *stability -> systematics under control*
    - the detector stability measured with a redundant set of calibration methods (evolving year by year)
      - *trigger system flexible to cope with any experimental requests*

FE readout choice: the DRS chip

- The Domino Ring Sampler
  - *waveform digitiser developed at Paul Scherrer Institut*
  - *capacitor array to store the charge from detector signals, each capacitor is a "bin" in our waveform*
    - read out with external ADC
  - *sampling speed tuneable from 800 MSPS to 5 GSPS*
  - *memory depth = 1024 bins (from 12.8 µs to 200 ns)*
    - time resolution demands for sampling speed greater that 1.5 GSPS - 600 ns memory depth
      - *requirement on trigger latency!*



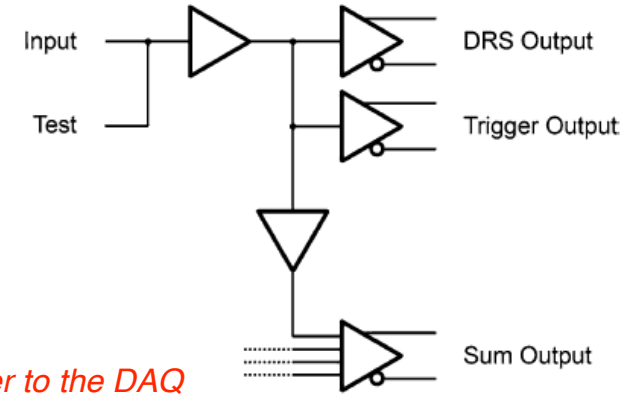


The DRS system

- DRS chips in VME boards
  - *32 channel per board*
- 5 crates
  - *<= 20 boards per crate*
    - 640 channels per crate
- Boards read-out with 2eVME protocol
  - *80 MB/s transfer speed*

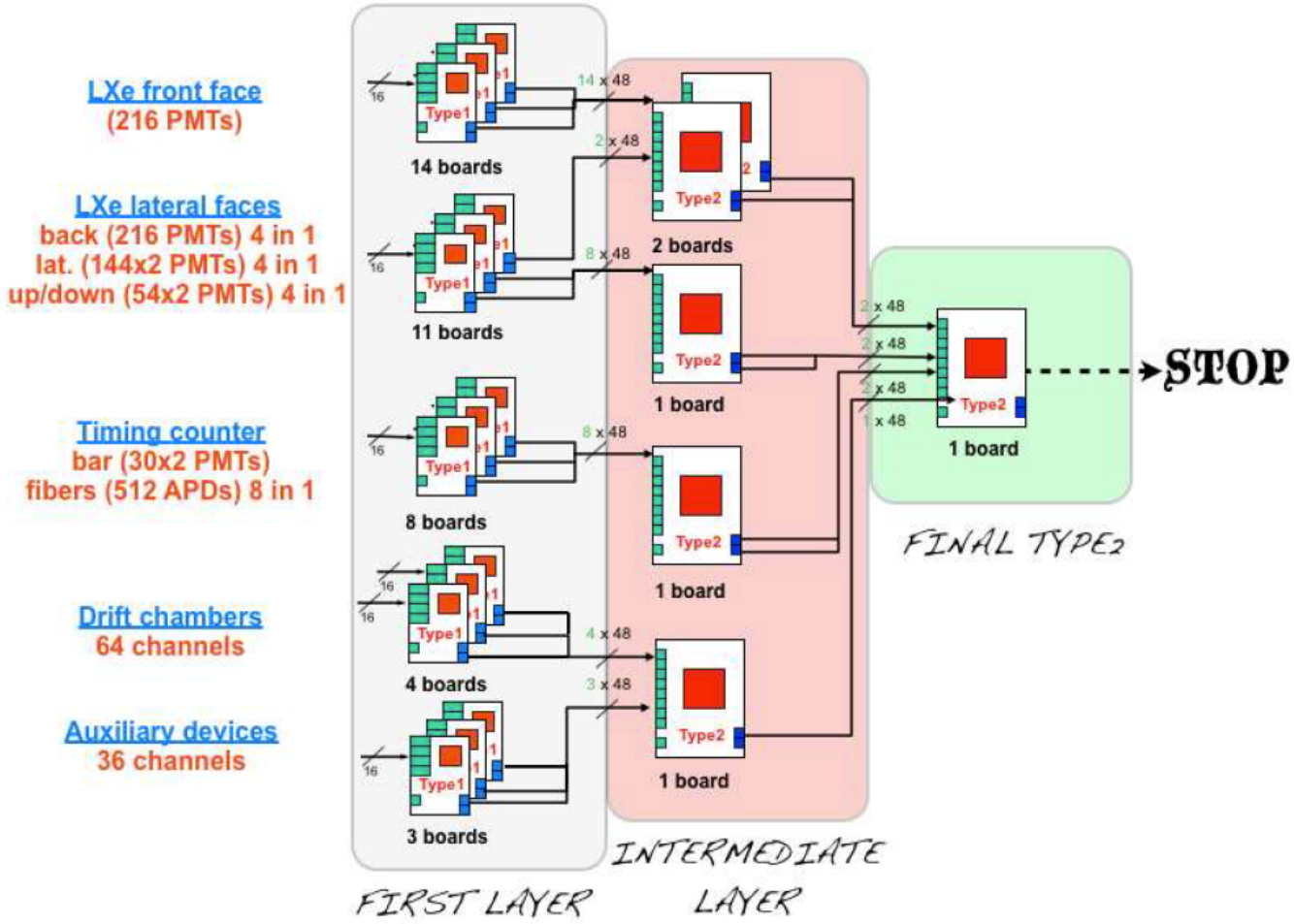
Trigger technology choice

- The trigger will be a separate system
  - *dedicated splitter system to drive signal to both DRS and trigger*
- Latency requirements
  - *DRS with 1.6 GSPS*
    - memory depth of about 600ns
  - *~50 ns are required before the pulse in DRS waveforms for offline processing*
  - *~50 ns is a conservative estimate for the trigger signal distribution from the trigger to the DAQ*
  - *the decision must be taken as fast as 500 ns after the event occurrence*
    - at trigger level only fast detectors are used: LXe calorimeter and Timing Counter (plastic scintillator detector)
- Flexibility requirements
  - *the system have to cope with any possible experimental needs*
    - …and there are really a lot of!
- FPGA based trigger
  - *Xilinx Virtex-II pro FPGA (bought in 2004)*
    - algorithm execution frequency to be 100 MHz
  - *VME boards*
    - the trigger data stored in the data stream for online algorithm calibration and monitoring
  - *Multi layer system*
    - transmission with LVDS serialiser-deserialiser, 4.8Gbit/s per connection



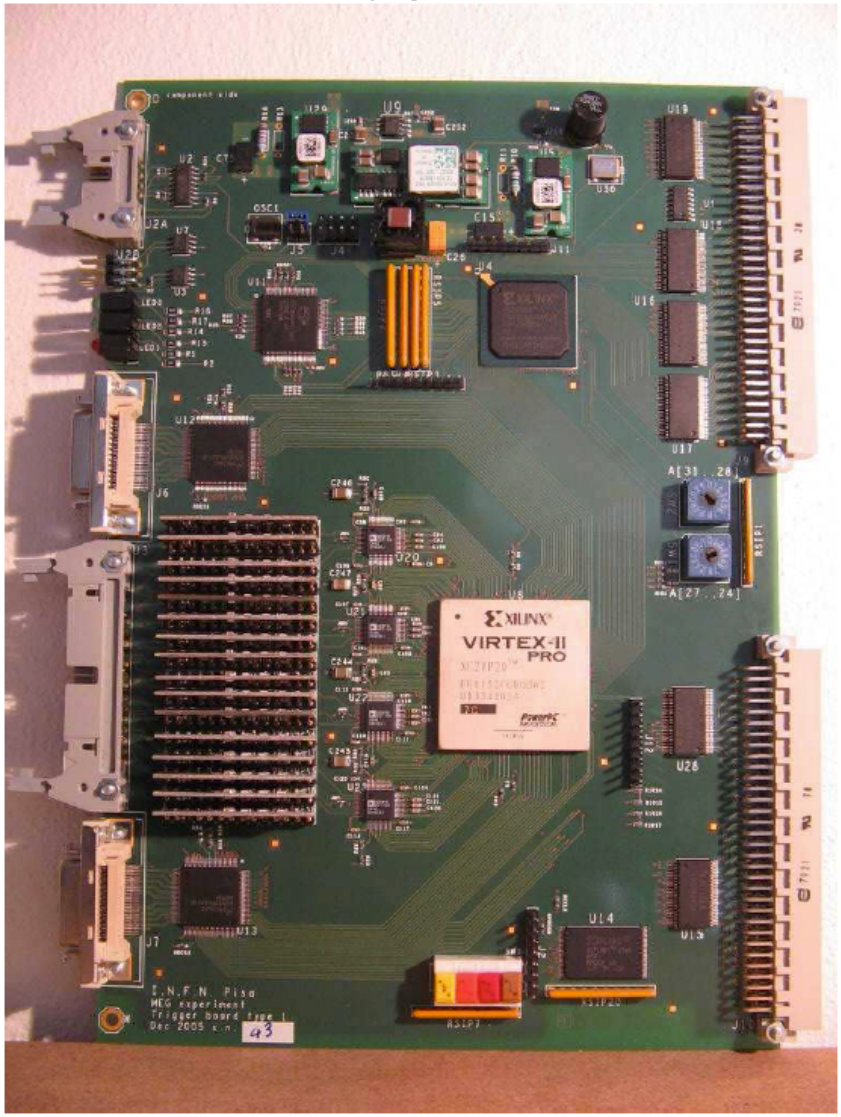
Overview of the trigger system

- Three layers system
  - *Type1 board*
    - 6U VME board
    - receiving detector data
    - equipped with FADC 100MHz, 10 bit
  - *Type2 board*
    - data processing
    - the master issues the trigger signal

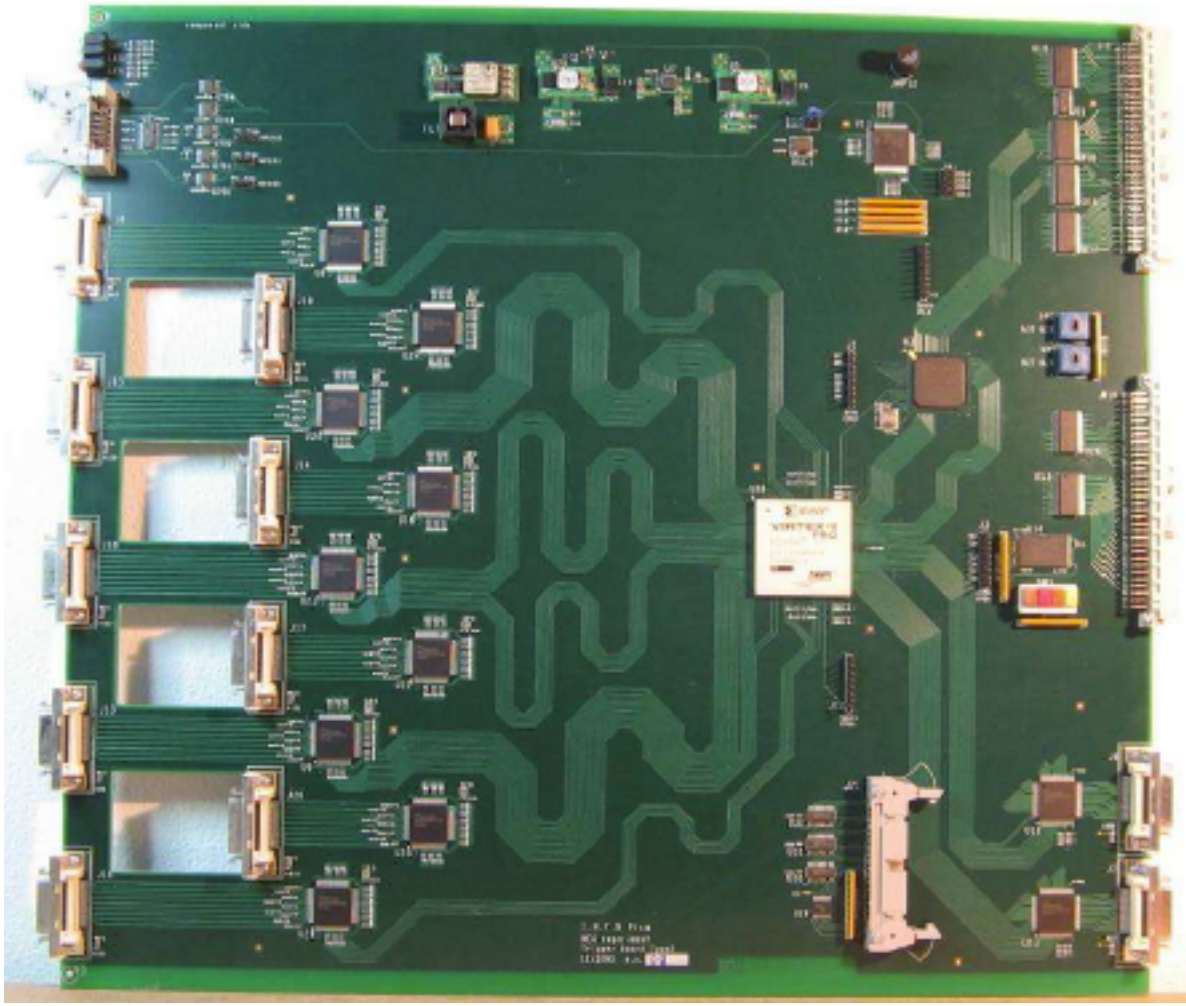


Some pictures… trigger boards

Type1



Type2

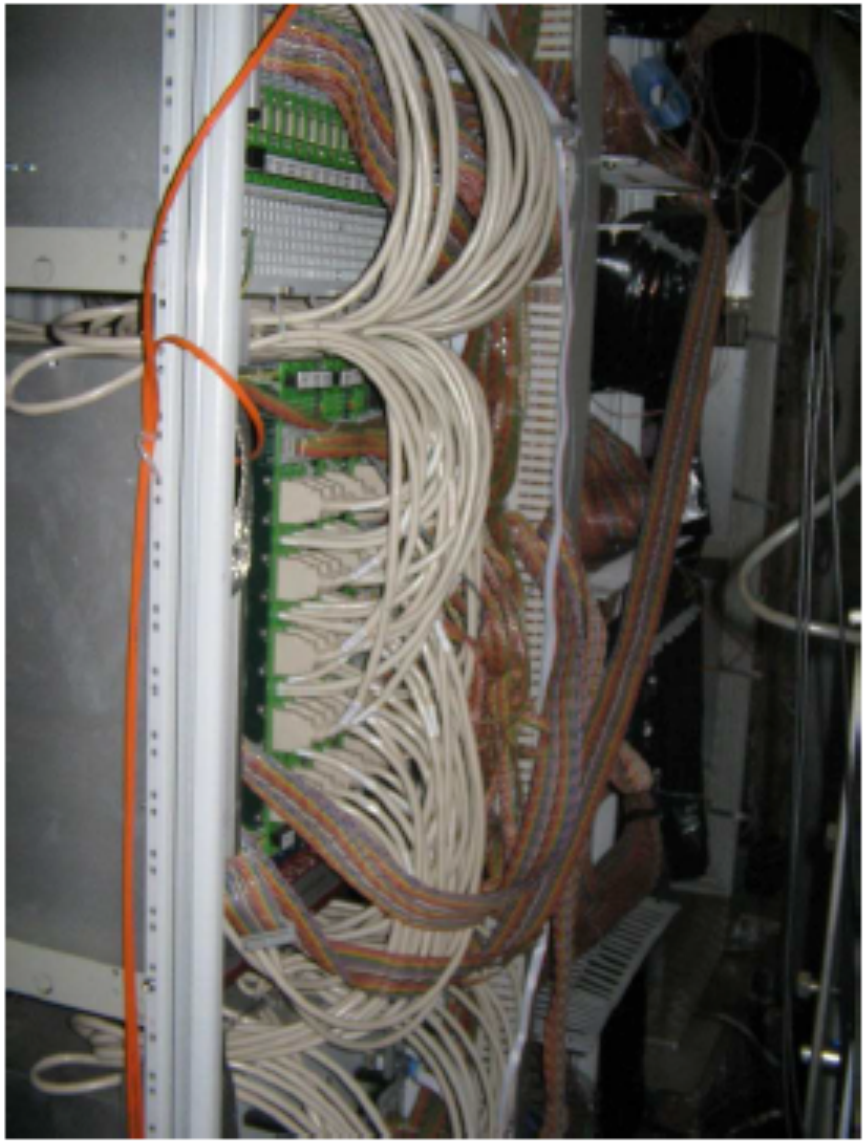


A picture of the trigger system

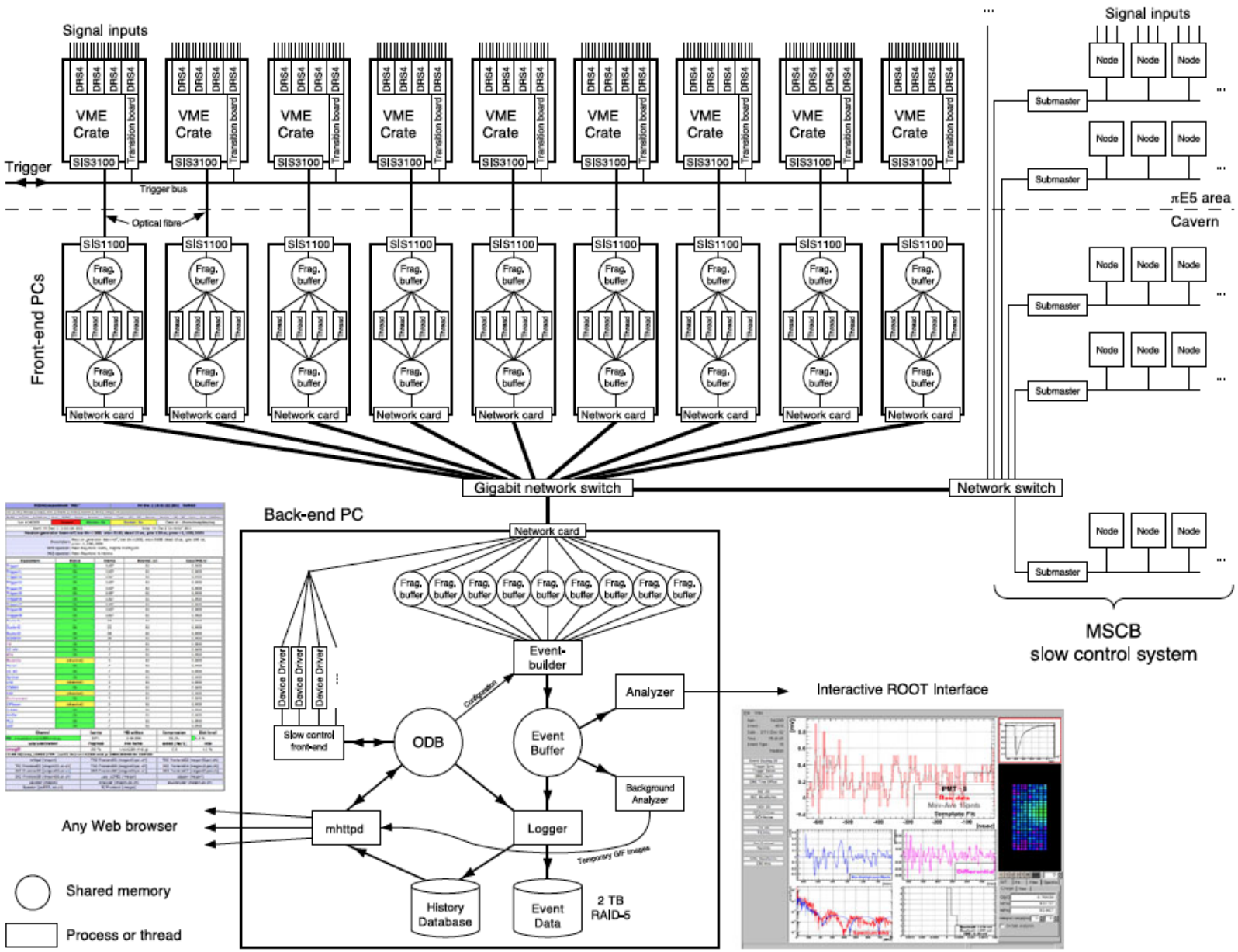
Type1 6U crate

Type2 9U crate

Type1 6U crate

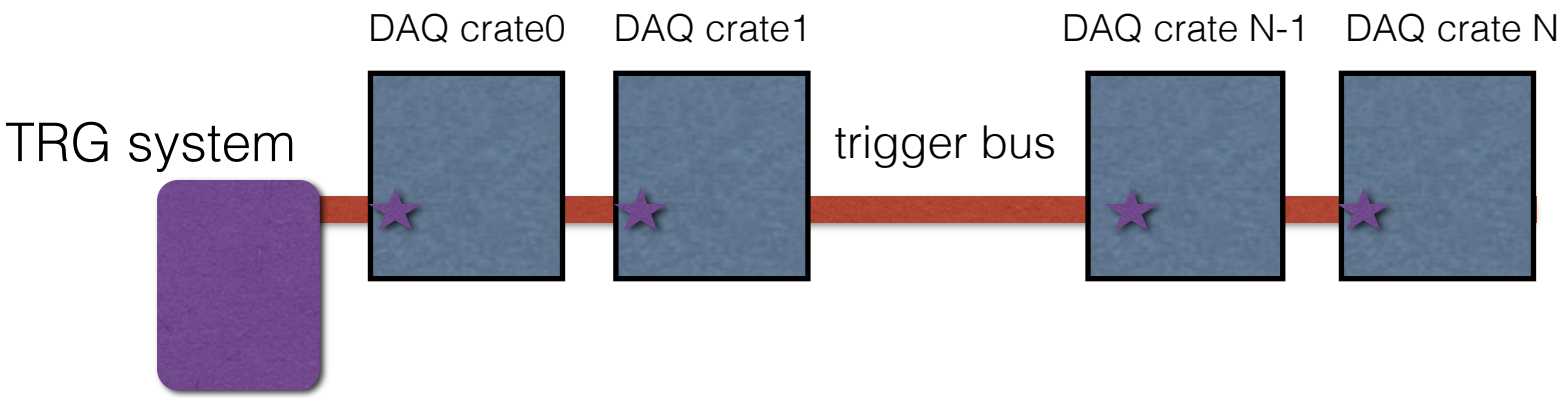


Data read-out system



Event building

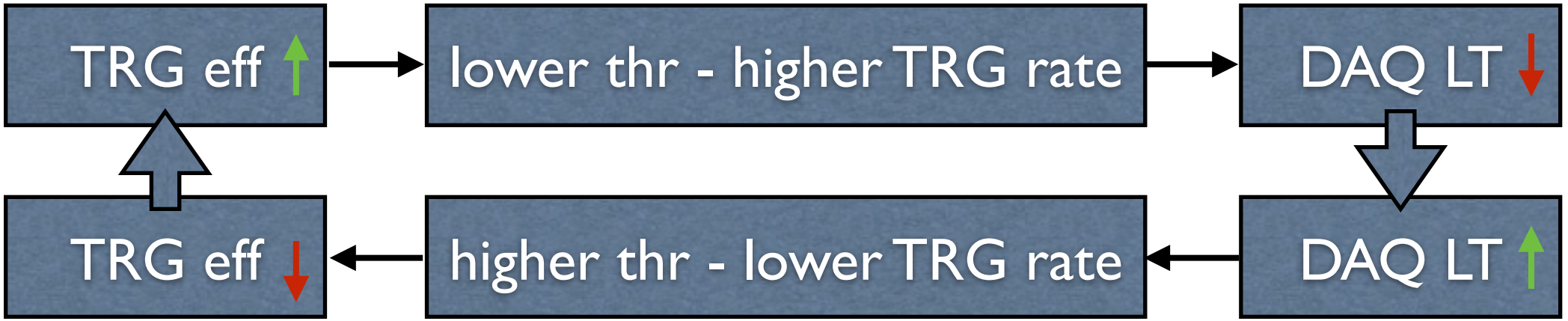
- The raw data are scattered over several VME crate
  - *each crate is read-out after any trigger by dedicated CPU that create a fragment*
  - *all the fragments sent to the main CPU*
    - what happens if at a certain point a fragment is loss??
      - *fragment belonging different events are mixed*
      - *data useless!!*



- The solution is the trigger bus: from the trigger to the DAQ crates
  - *event number + event code (trigger type)*
  - *connected via the VME transition cards*

Some consideration about DAQ efficiency

- DAQ efficiency is defined as
  - *DAQLiveTime x TriggerEfficiency*
- This quantity has to be as large as possible in any TDAQ system



- Compromise…
  - *milestones! let's start from dead time…*

Read-out dead time

- As a first attempt we used a single buffer read-out
  - *the system is in dead time during the read-out of a crate through VME*
  - *the dead time is dictated by the heaviest crate*

$$t_d = \underset{\#\ \text{mezzanines/crate}}{40} \times \Big(\underset{\#\ \text{DMA setup time}}{0.125\,\text{ms}} + \underset{\#\ \text{data transfer time}}{40960\,\text{B}/83000\,\text{MB/ms}} \Big) = 24.7\,\text{ms}$$

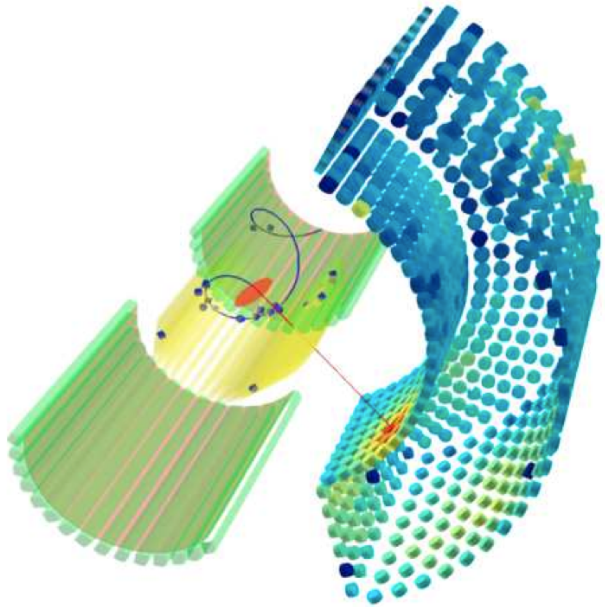
- the associated DAQ live time is given by the probability to have 0 event during the read-out (trigger rate = 7 Hz, Poissonian event distribution)

$$LT = \exp(-R_{trg} \cdot t_d) \simeq 84\%$$

- Requirement for the trigger rate
  - *which trigger algorithms can we use?*
  - *what about the rejection power?*

Considerations about trigger algorithms

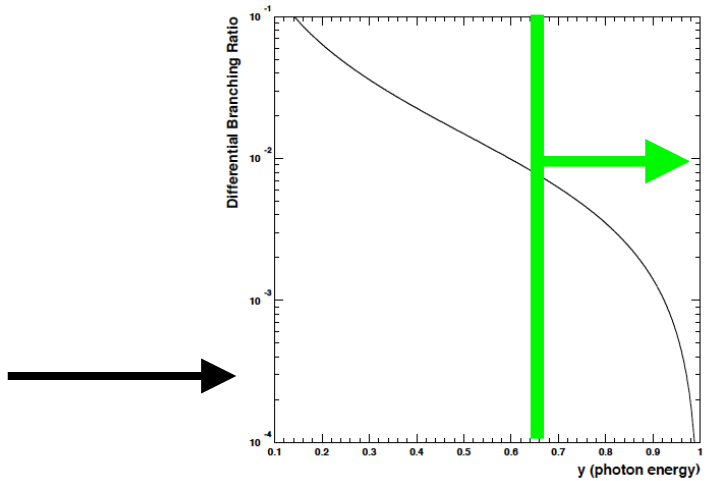
- Goal: DAQ rate of about 7 Hz
- Detector to be used:
  - *LXe calorimeter (photon energy, time and direction)*
  - *Timing Counter (positron time and hit position)*
- Synchronous processing with FPGA
  - *algorithms to use 1 CLK cycle per operation (if possible)*
    - registers, adders, subtracters, comparators for simple operations
    - Look Up Tables for more complex operations, for example multiplications



This choice is driven by physics and your detector!

- Final choice
  - *photon energy discrimination*
  - *photon-positron timing*
  - *photon-positron space correlation*

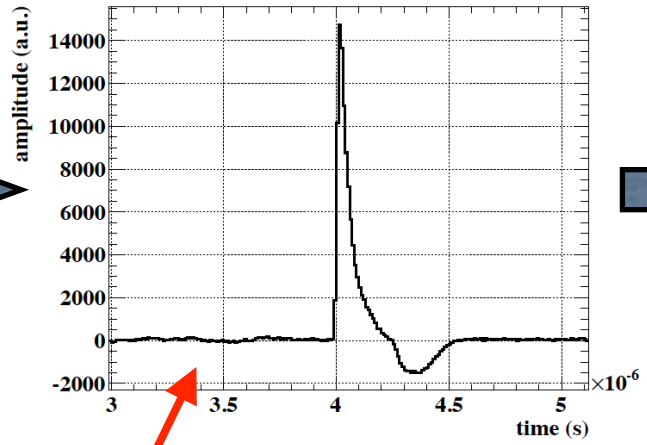
- The discrimination on photon energy plays the lead role
  - *take care of online energy reconstruction*

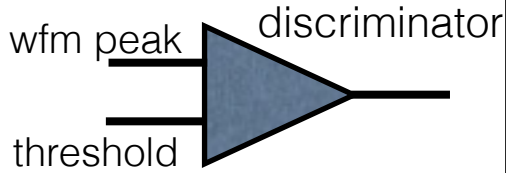


Online energy reconstruction

- Synchronous sum of the LXe signals
  - *the peak of the obtained waveform is the online energy estimate*

$$A_{tot}(t) = \sum_{i=0}^{N_{CHA}-1} f_i A_i(t)$$





- Refined algorithm
  - *online pedestal subtraction*
    - the baseline value may differ from channel to channel (remember QDC in lab4)
      - *refer the waveform to a common value before the sum stage*
  - *channel calibration with Look Up Table*
    - for example in case of PhotoMulTipliers gain and Quantum Efficiency calibration

Implementation of online pedestal subtraction

pipeline with shift registers

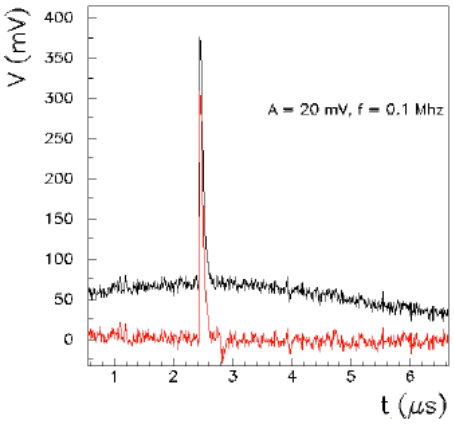
Algorithms with most schematic language

VME protocol + some algorithm with verilog

If the sample is larger than threshold it is not included in the pipeline

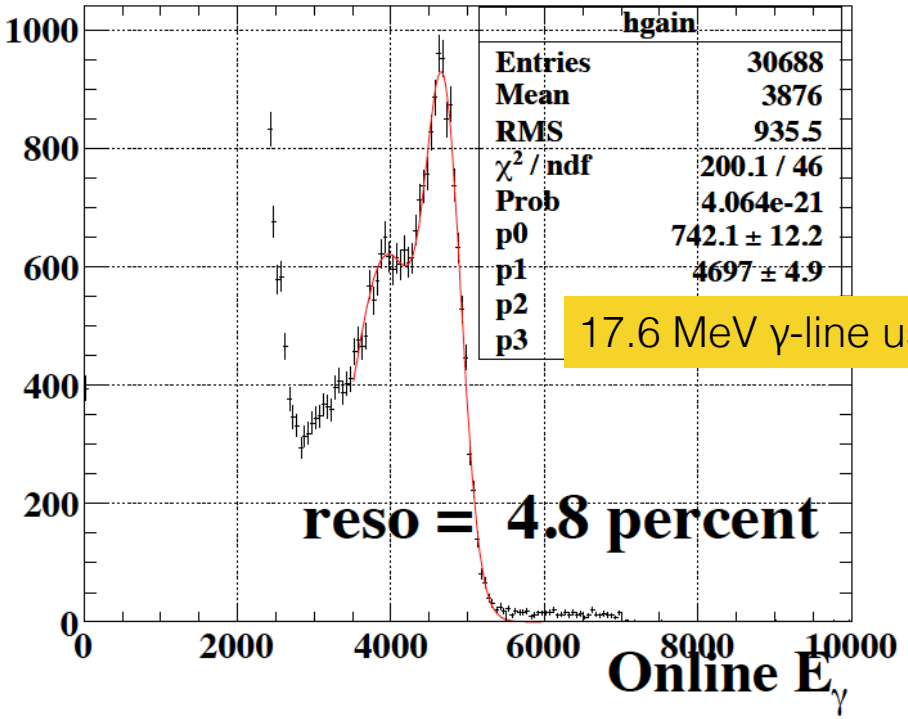
mean value from last 4 samples in pipeline as pedestal estimator

pedestal subtraction in 1 CLK cycle!!

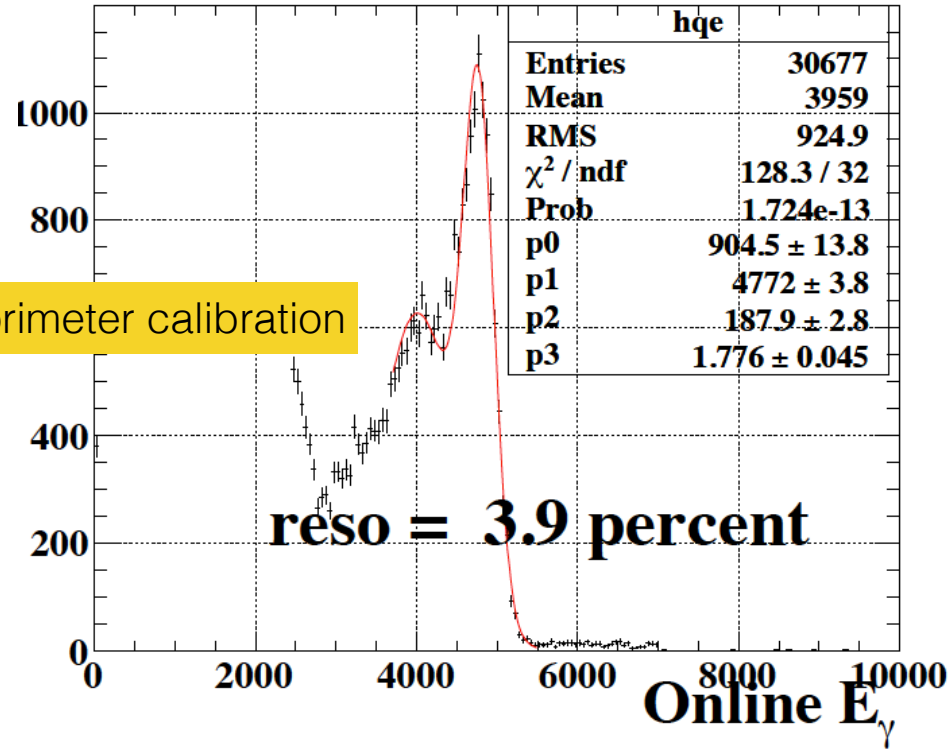


Effect of calibration!

- By applying a refined calibration reconstruction improves!
  - *better reconstruction -> higher threshold with same efficiency on signal -> lower trigger rate -> higher DAQ Live Time*
    - improvement of the DAQ efficiency



17.6 MeV γ-line used for calorimeter calibration

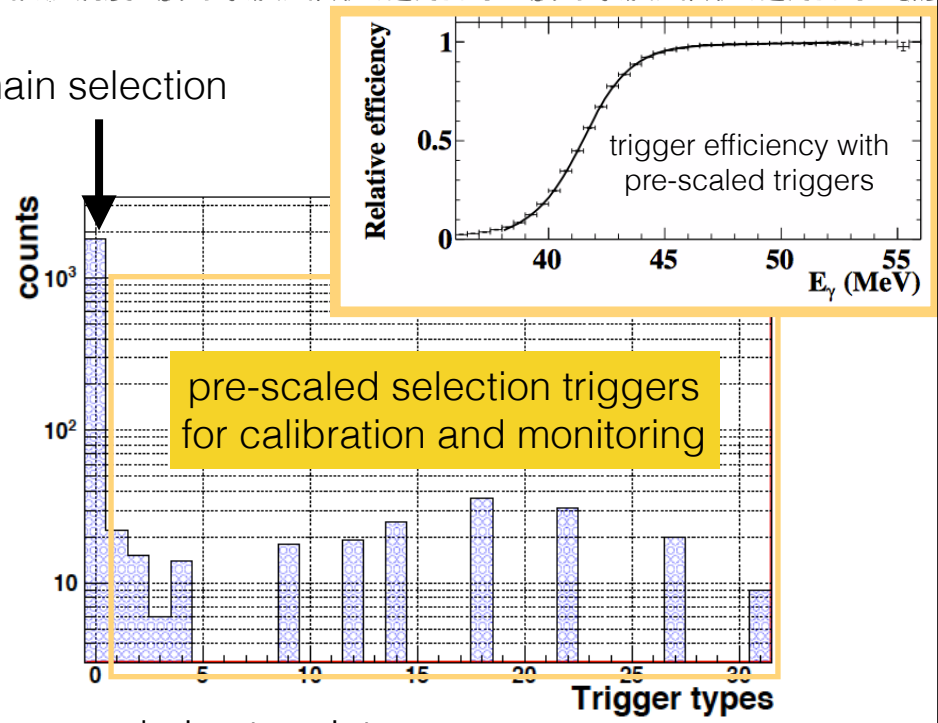


- It is important to get the best from the system with the available techniques!

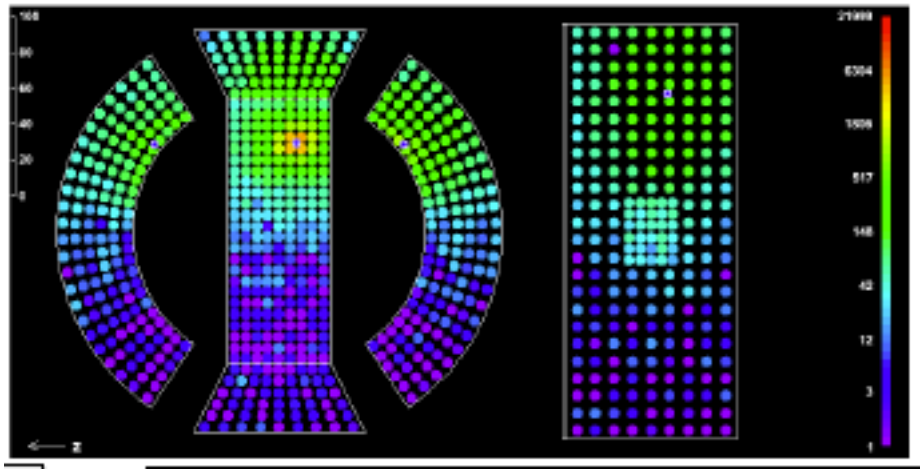
Run configuration and monitoring

- The trigger system provides the DAQ with 32 different selection algorithms
  - *the trigger type are order with a priority*
  - *pre-scaling factors*
    - programmable fraction of minimum bias selection in the data stream
      - *trigger efficiency studies*
      - *detector calibration and monitoring*
      - *physics analysis, normalisation evaluation*
- This flexibility is crucial to take under control the detector
  - *the run configuration is store in a database*
    - access to the shift crew
    - few default run configuration available
    - is it written together for data to be used in analysis
- GUI to check data quality online
  - *events "photographs"*
    - not so precise, sensitive to major problems
- Automatic analysis right after the acquisition to check further the data quality
  - *histograms for detector debugging*
    - dead channels, electronic noise…

main selection

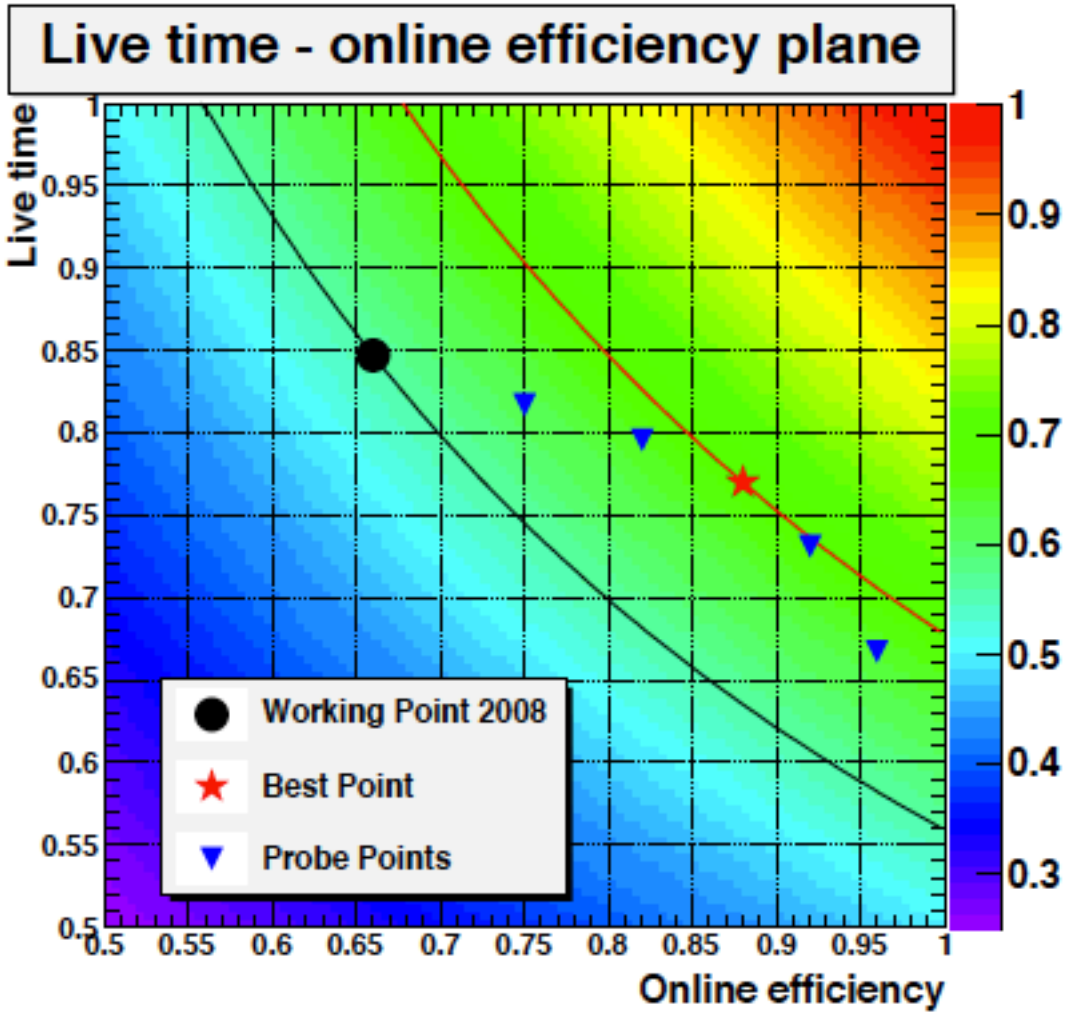


calorimeter picture



DAQ efficiency: measurement and optimisation

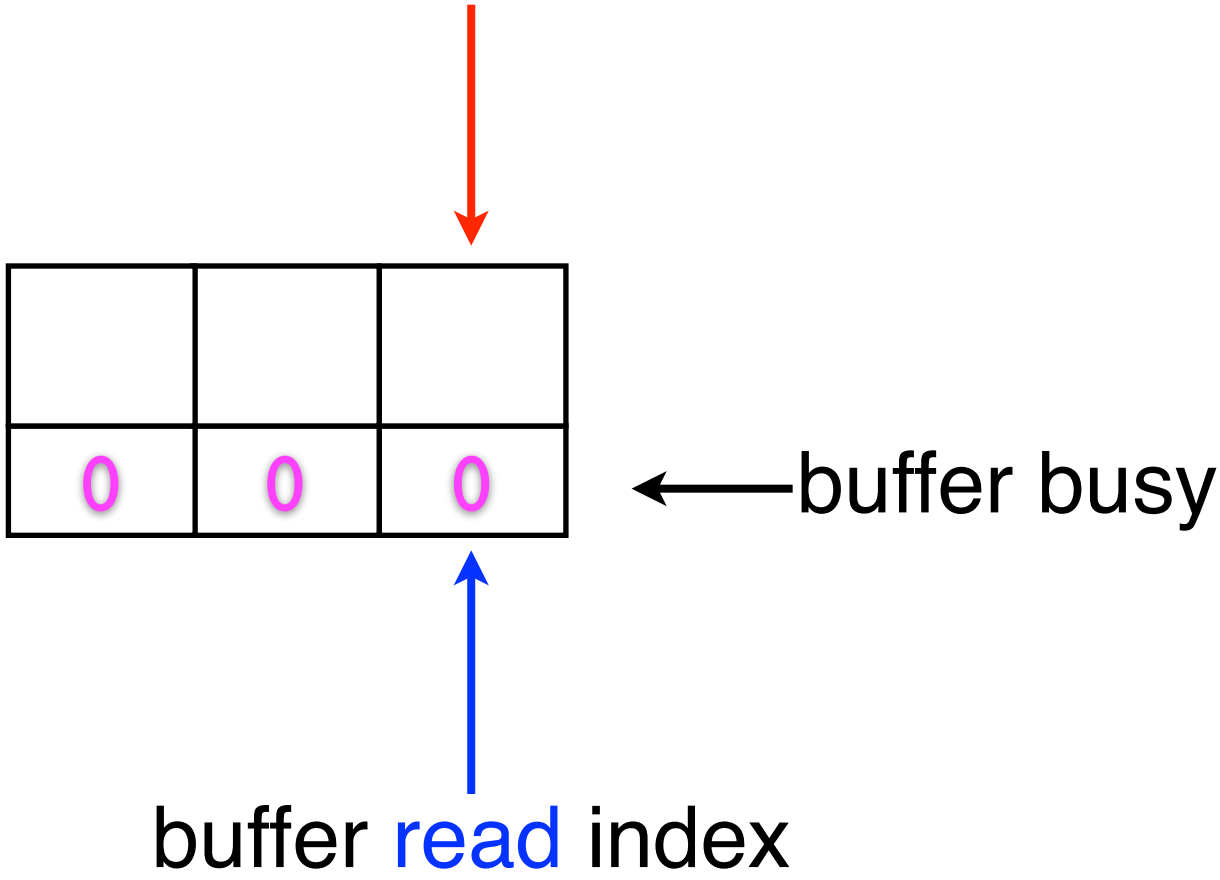
- First run in 2008: DAQ efficiency = 55%
  - *"preliminary" trigger system configuration*
    - first version of online reconstruction algorithm
    - calibration not optimised yet
- The DAQ (trigger) efficiency measured at the end of data taking
  - *the selection has to be trained with data*
  - *the higher Live Time is NOT the experimental working point*
    - but the Live Time is measured online…
- How to find the best working point?
  - *trigger "simulation" with real data*
    - modify selection and predict the trigger rate and efficiency with no algorithm improvement
  - *improve the selection algorithms*
    - for example improve calibrations!
- By algorithm refinement we reached 75% DAQ efficiency



Is it possible to improve further?

- Many possibilities
  - *reduce dead time read out*
    - zero-suppression of FPGA to neglect the read-out of “empty” channels
      - *this is dangerous with an homogeneous calorimeter…*
    - Use 2eVME D64 read-out (160 MB/s instead of 80 MB/s)
      - *you have to foreseen it at design phase… :(*
        - unfortunately you usually miss something in your original design
  - *refined selection algorithms*
    - current latency of the order of 500 ns, DRS running at 1.6GSPS
      - *a more complex algorithm would lead to a larger latency… :(*
  - *Multiple-buffer read-out!*
    - data stored in circular memories on VME boards
      - *during a buffer read-out the next buffer is filled (if free…)*
      - *busy released immediately!*
        - this can be implanted in FPGA in any time… if you have enough resources!

Multiple buffer(0)

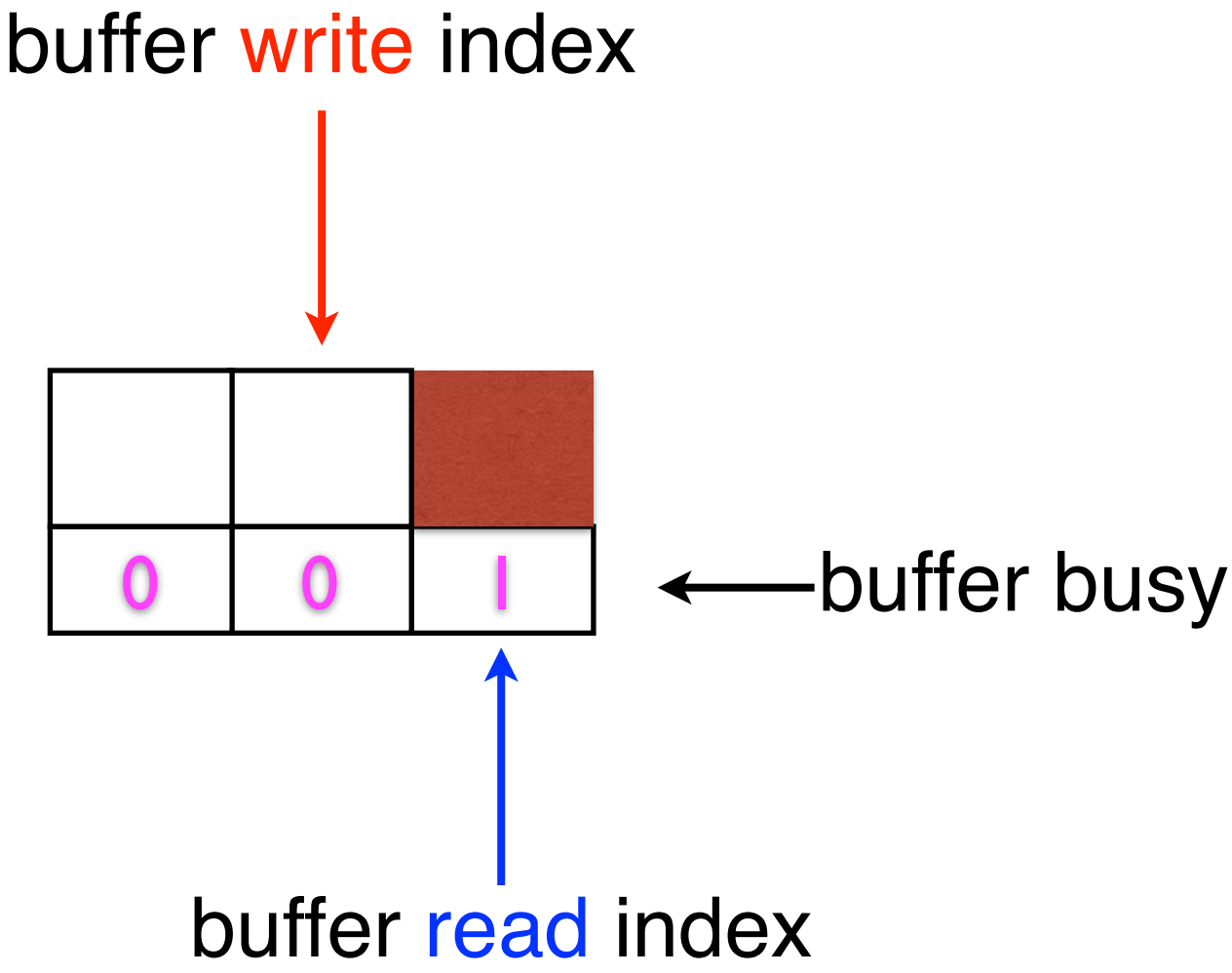


Situation at the begin of a run

all buffers free

all BUSY at 0

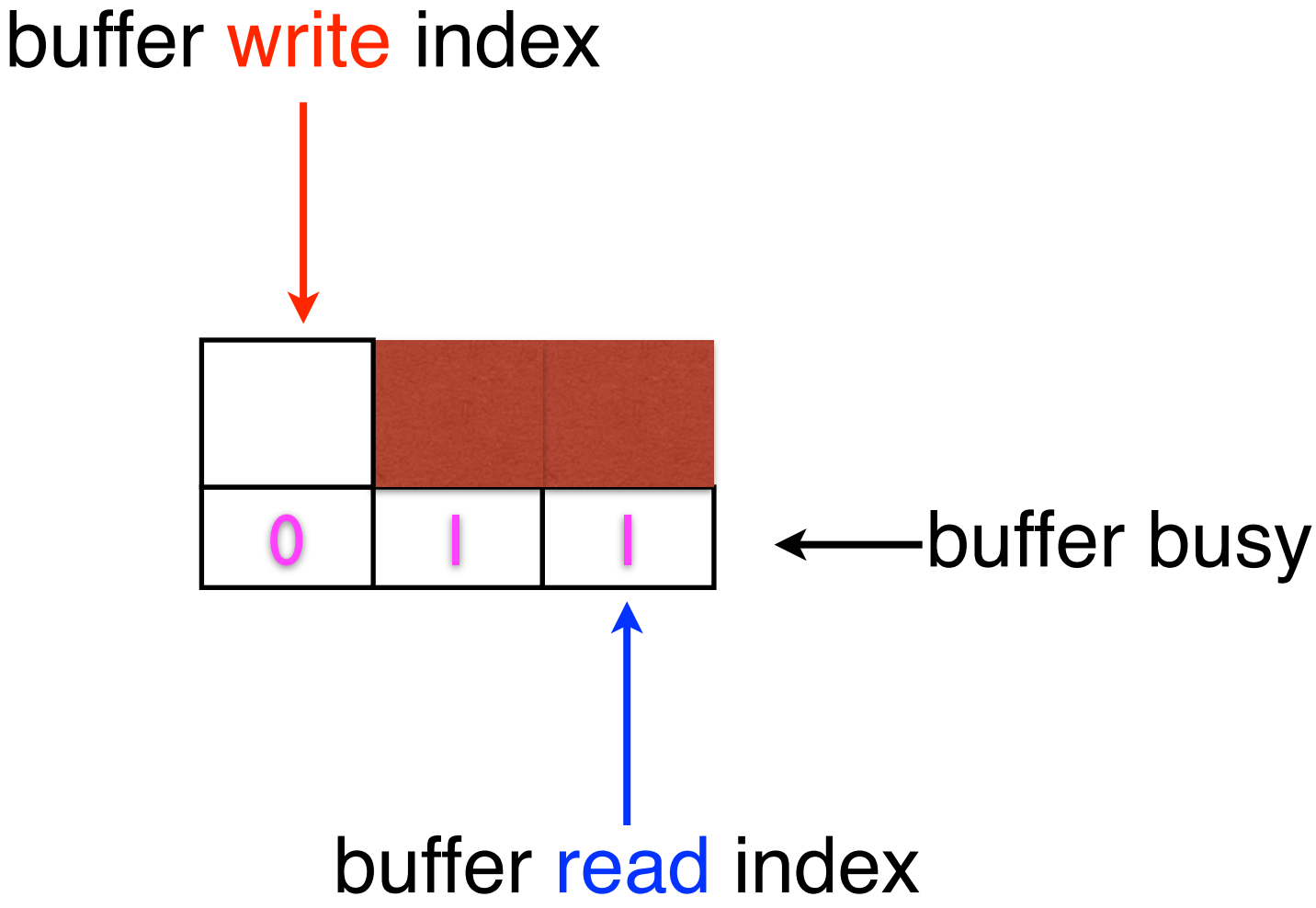
Multiple buffer(1)



Situation after the first event

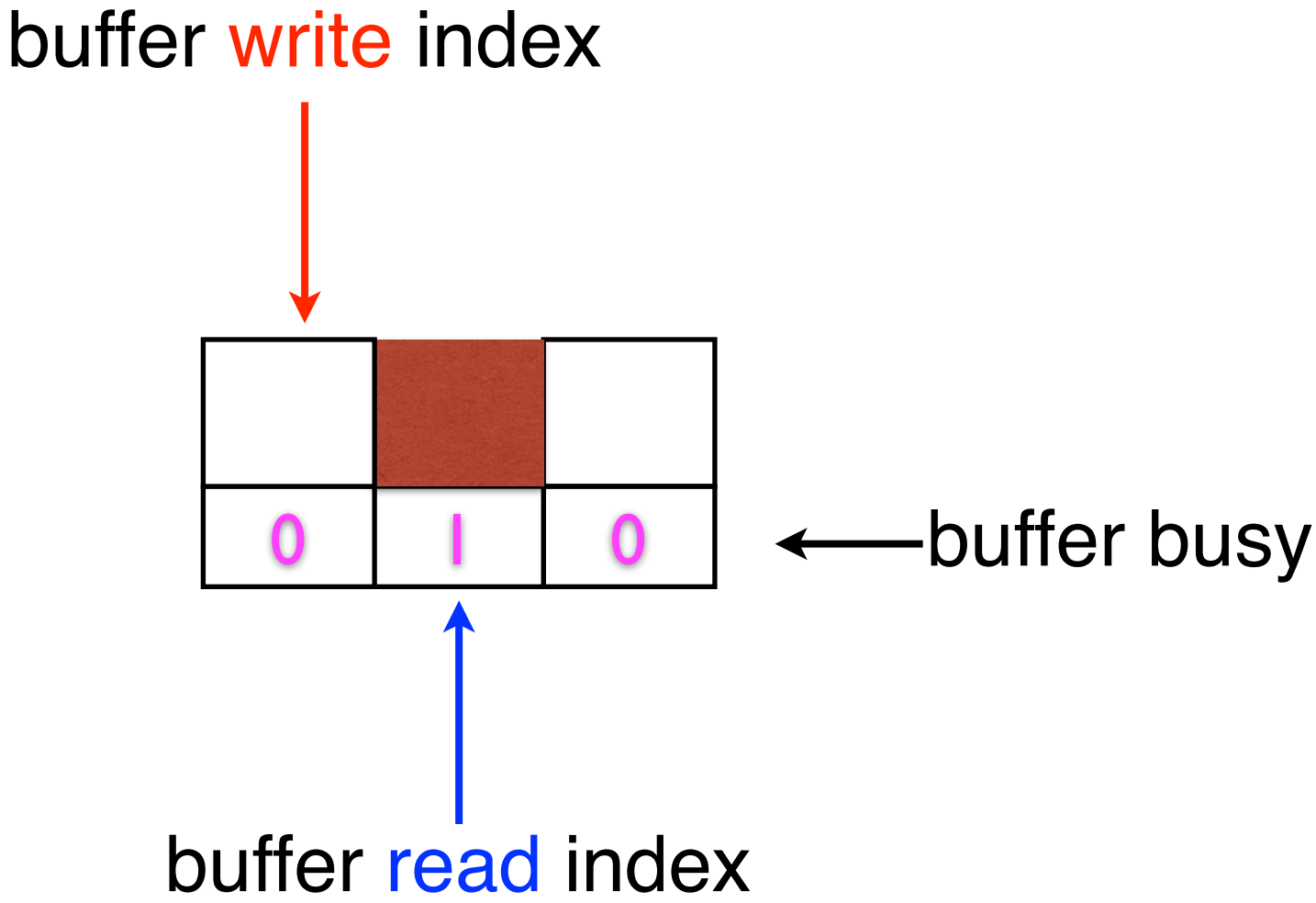
1 buffer full filling the next one

Multiple buffer(2)



Another event during the first buffer read-out

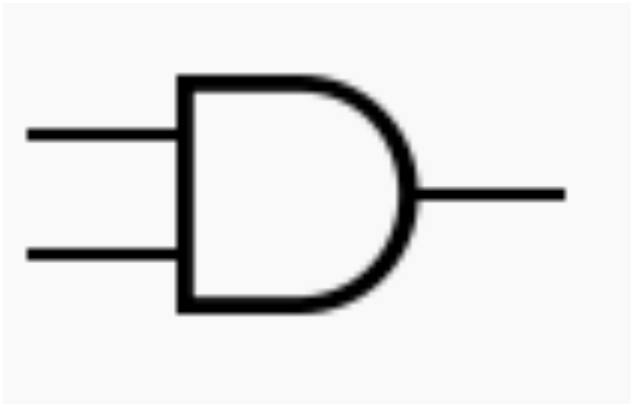
Multiple buffer(3)



The first buffer read-out finished

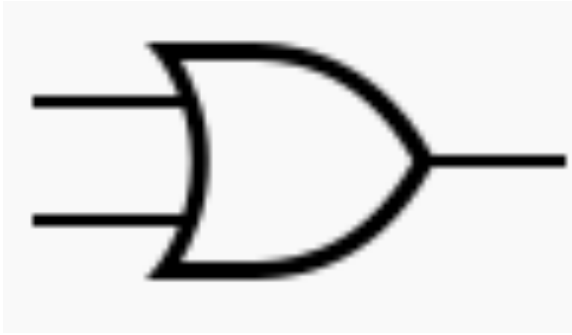
Busy handling

BUSY0
BUSY1
BUSY2



ALL RAM full --> DAQ in stop

BUSY0
BUSY1
BUSY2



NOT EMPTY RAM --> read out

- Read-out when one of the buffer busy is 1
- The system is busy when all the 3 buffers are full

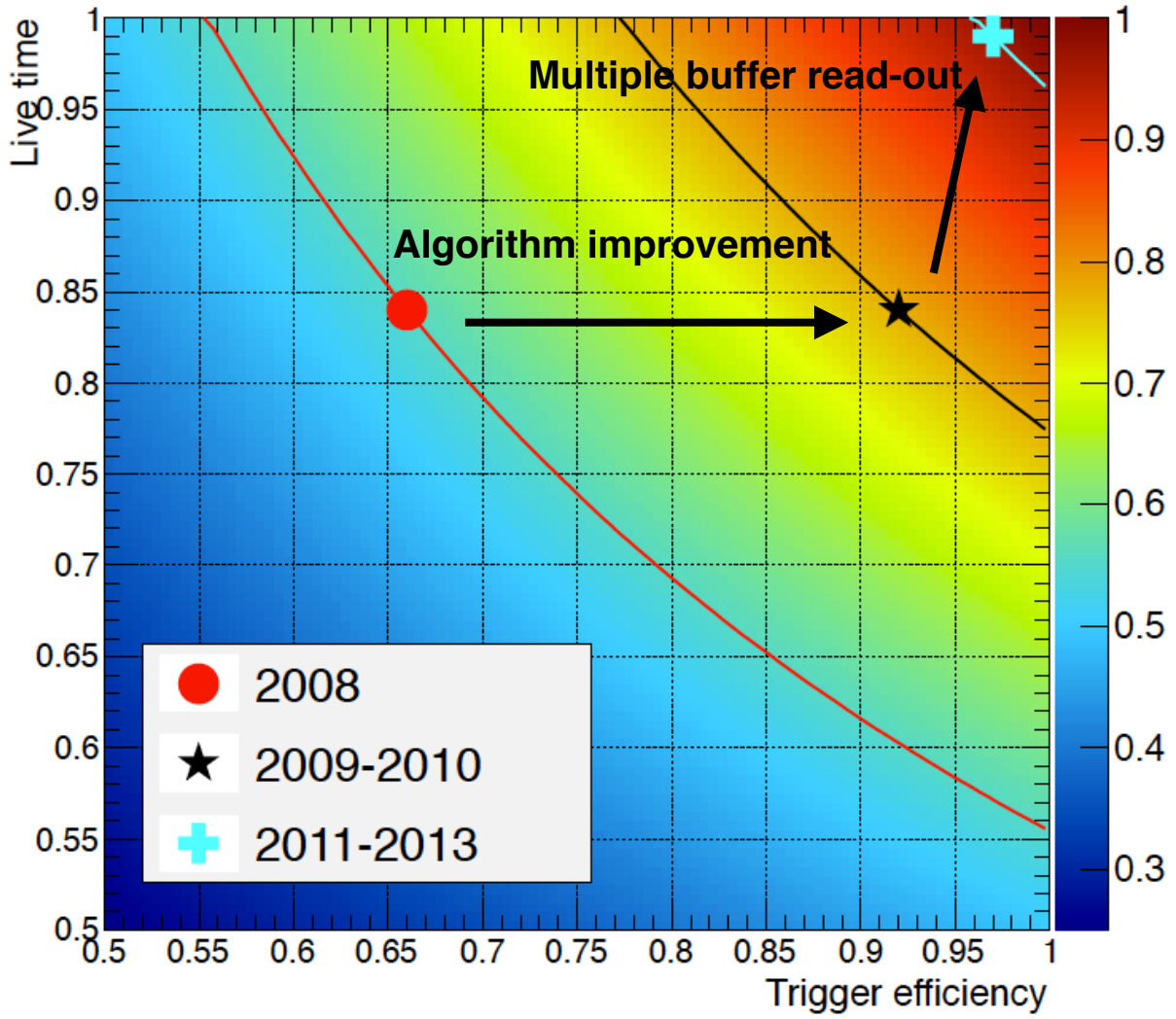
And DAQ Live Time?

- The system is busy when all the 3 buffers are filled during while you are reading
  - *the read out time is unchanged*
    - ~25ms
  - *the LiveTime fraction can be evaluated*

$$LT = e^{-R_{trg} \cdot t_d} \cdot [1 + R_{trg} \cdot t_d + (R_{trg} \cdot t_d)^2 / 2!]$$

- it is close to 99% even with a trigger rate close to 14 Hz
  - *there is room to relax the trigger condition and improve also the trigger efficiency!*

Final DAQ efficiency



Conclusions

- A TDAQ system to be designed on top of an experiment
  - *the experimental needs drive the choice of the technology*
- Even when the technology is decided a lot of compromise, in this case
  - *Waveform digitiser with lat least 1.6 GSPS sampling speed*
    - background rejection requires the best timing possible
    - trigger latency to be least than 500 ns
  - *trigger based FPGA*
    - system flexibility, for example detector calibration
      - *calibration procedures may changes during the run… and you do want to be the reason why a new calibration procedure will not be used ;)*
- Once your TDAQ is built you have to get the best performance
  - *use all the accessible handles*
    - this is just an example but can help