Introduction to trigger systems

Manfred Jeitler
Institute of High Energy Physics (HEPHY)
of the Austrian Academy of Sciences

Level-1 Trigger of the CMS experiment
LHC, CERN

Trigger


3361150
ASA
1000
500
250
125
60
30
15
8
4
2
1
B
50
OPTIMA 1535
sensor

HEPHY
Institute of High Energy Physics

trig•ger | ˈtrigər |

noun

a small device that releases a spring or catch and so sets off a mechanism, esp. in order to fire a gun: *he pulled the trigger of the shotgun.*

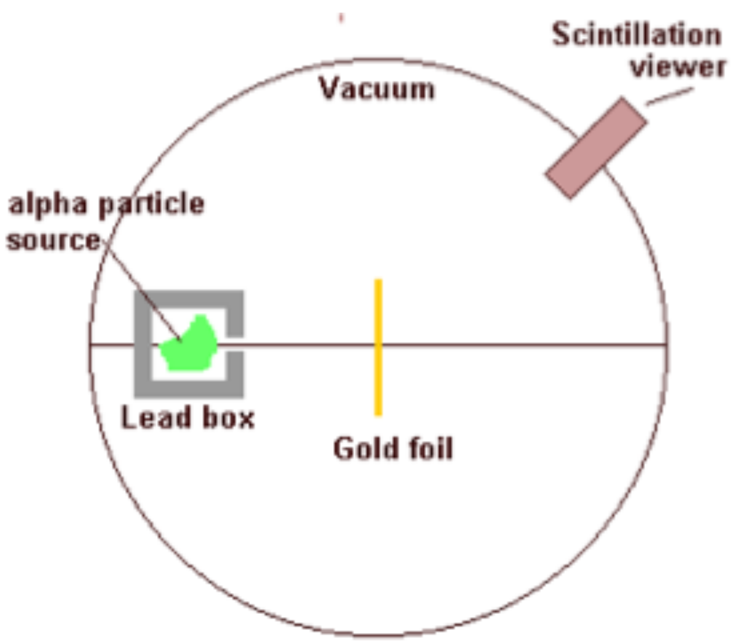
- an event or thing that causes something to happen: *the trigger for the strike was the closure of a mine.*

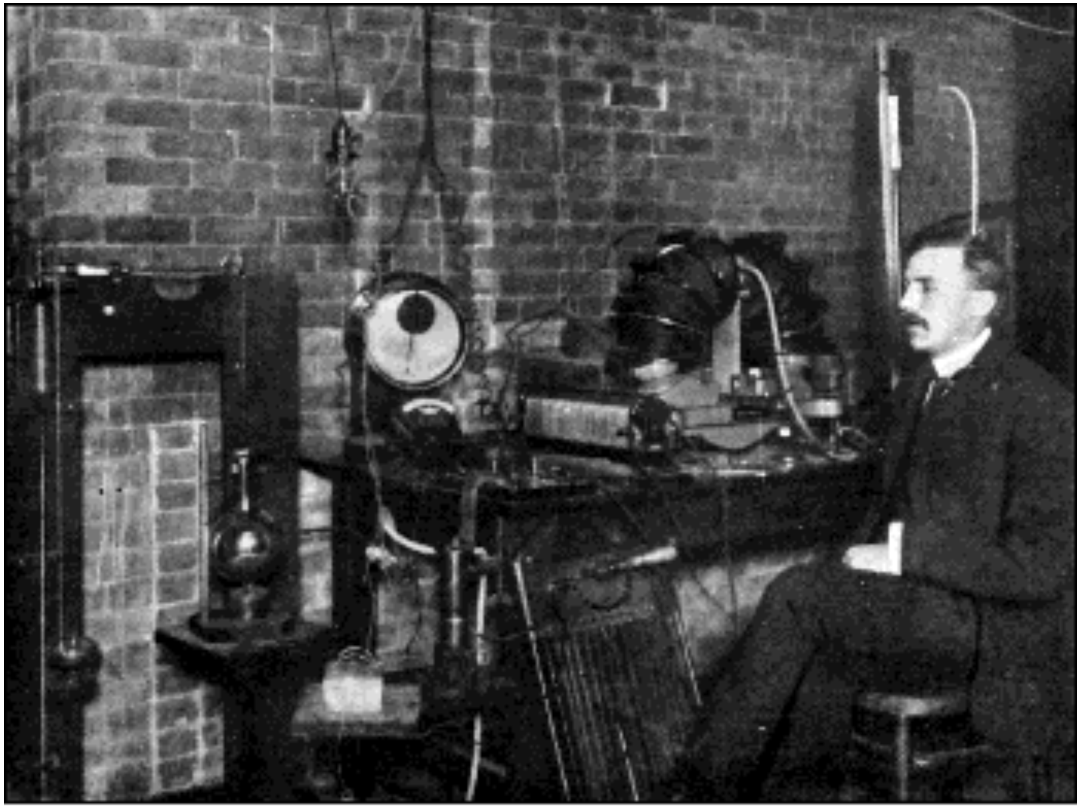
Wikipedia: "**A trigger is a system that uses simple criteria to rapidly decide which events in a particle detector to keep when only a small fraction of the total can be recorded.** "

trigger: features

- fast
- simple
- selective
  - purity
  - efficiency
- needed when only a small fraction can be recorded

- first particle physics experiments needed no trigger
- were looking for most frequent events
- physicists observed all events



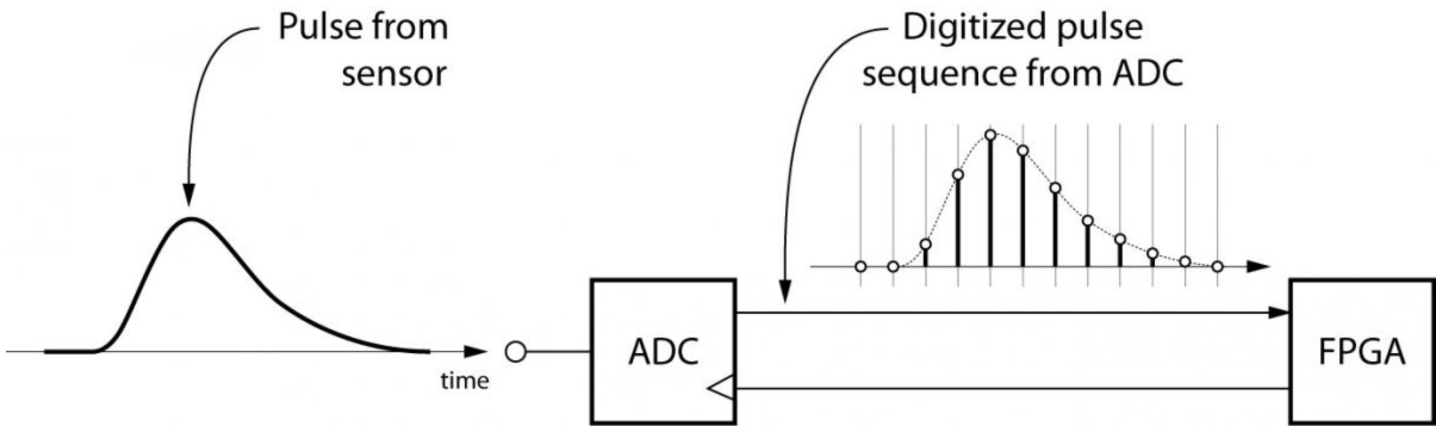


Ernest Rutherford with Gold Foil Experiment

- later physicists started to look for rare events
  - "frequent" events were known already
- searching "good" events among thousands of "background" events was partly done by auxiliary staff
  - "scanning girls" for bubble chamber photographs

periodic trigger - look all the time

- digitize at constant intervals
  - at each “clock cycle”
  - you might also say: “no trigger”, or “untriggered readout”

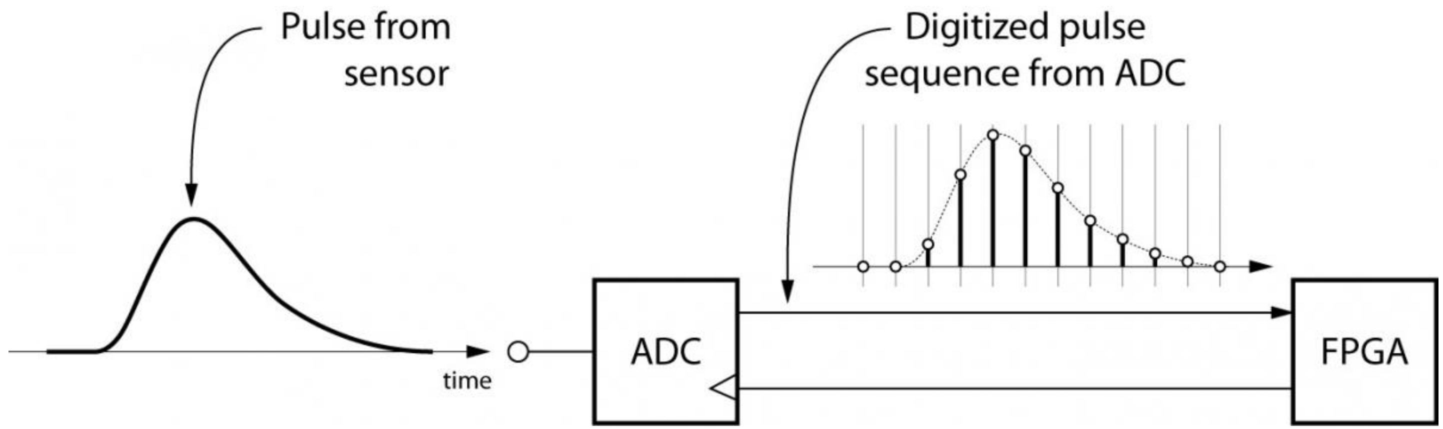


- select data afterwards
  - from digital information

- all you need ... or not?

periodic trigger - look all the time

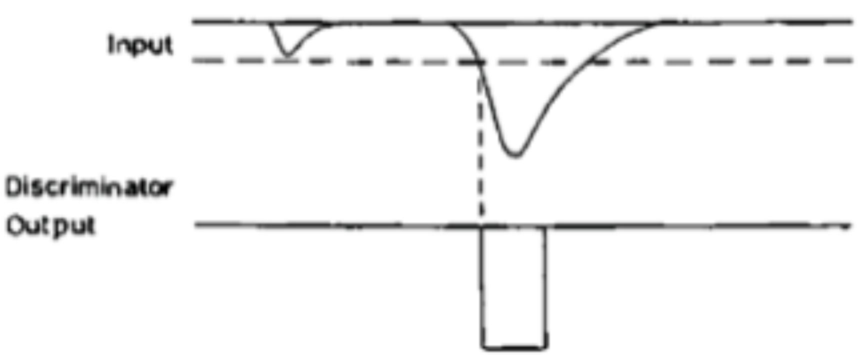
- digitize at constant intervals
  - at each "clock cycle"
  - you might also say: "no trigger", or "untriggered readout"



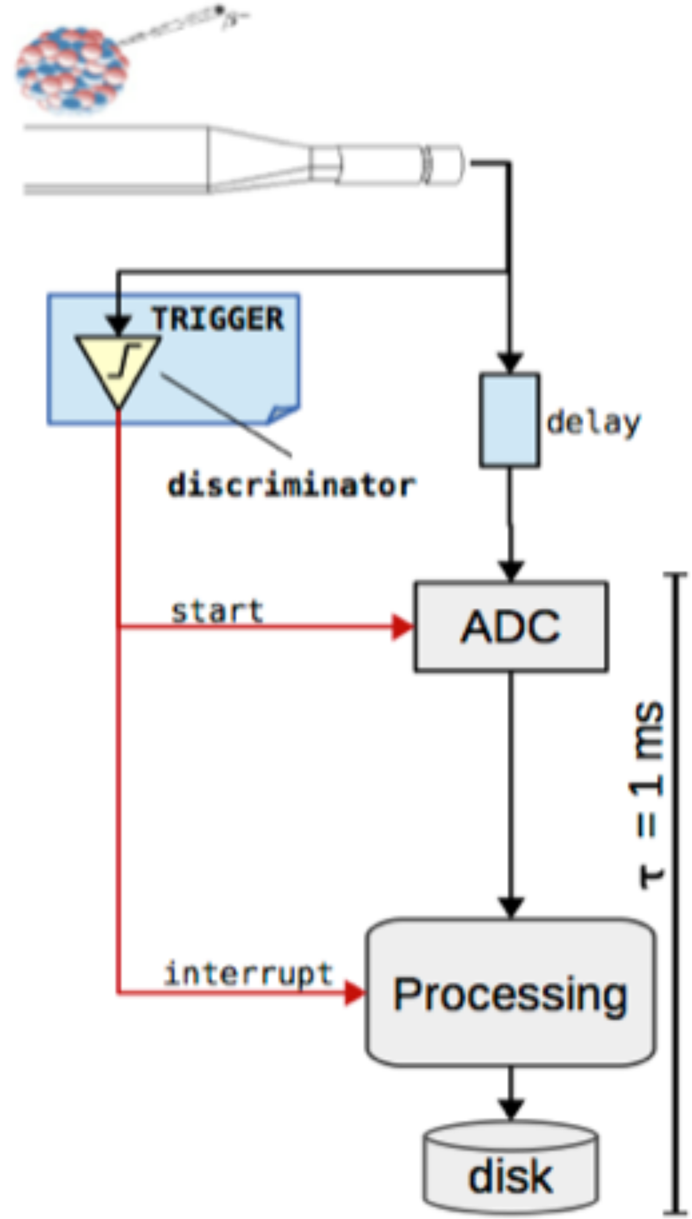
- may be not very efficient
  - needs fast "flash" ADC (FADC)
    - » ADC = Analog-to-Digital Converter
  - big data volume to handle
  - mostly zeroes → have to remove using "zero suppression mechanism"
- e.g. digitization interval $\tau = 1$ ms → readout rate = 1 kHz

trigger: "tell me when to read out"!

- events often arrive in asynchronous and unpredictable way
  - e.g. radioactive decay
- "trigger" needed to know when to digitize
  - **discriminator** generates an output signal only if amplitude of input pulse is greater than a certain threshold

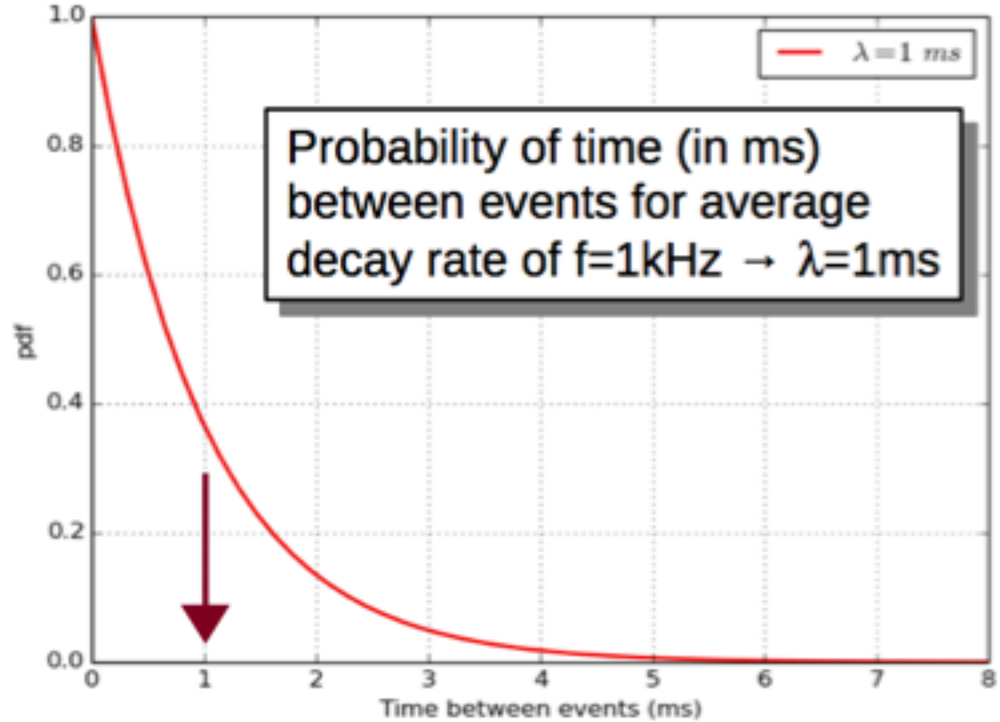


  - ADC: analog-to-digital converter



using a trigger - what may happen?

- events arriving in random way
- waiting time: exponential
- e.g. mean rate: 1 kHz
  - 1 event per millisecond on average
  - → average time between events: 1 ms
  - → we have to process one event per ms, on average

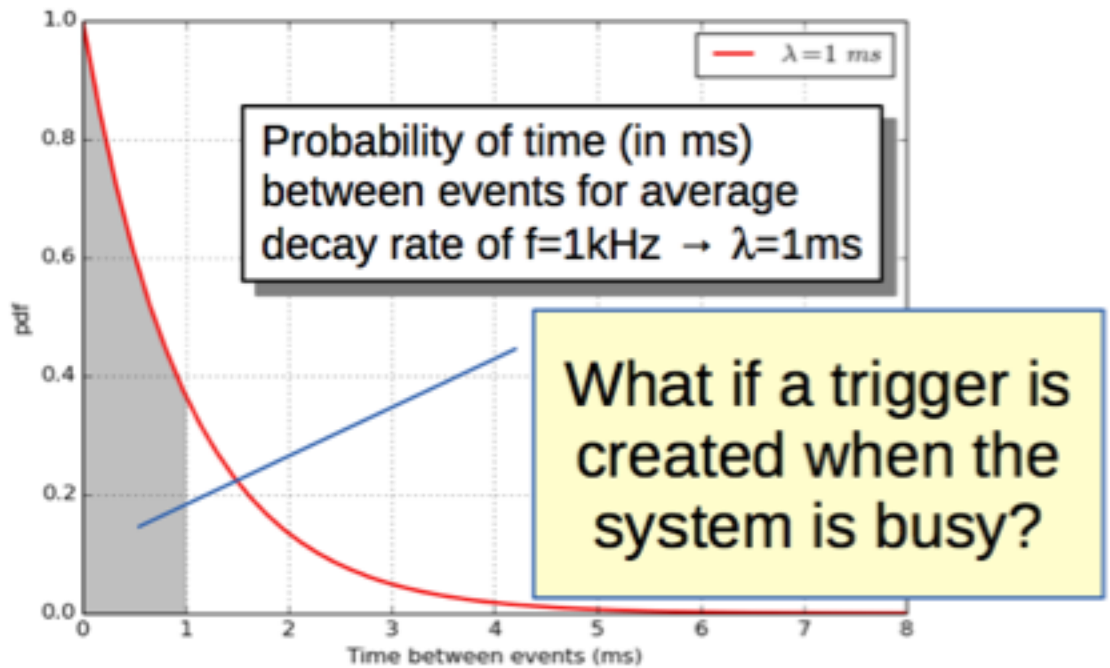


- but waiting time can be much longer or shorter!

→ can this be a problem?

using a trigger - what may happen?

- events arriving in random way:
- waiting time: exponential
- e.g. mean rate: 1 kHz
  - 1 event per millisecond on average
  - → average time between events: 1 ms
  - → we have to process one event per ms, on average



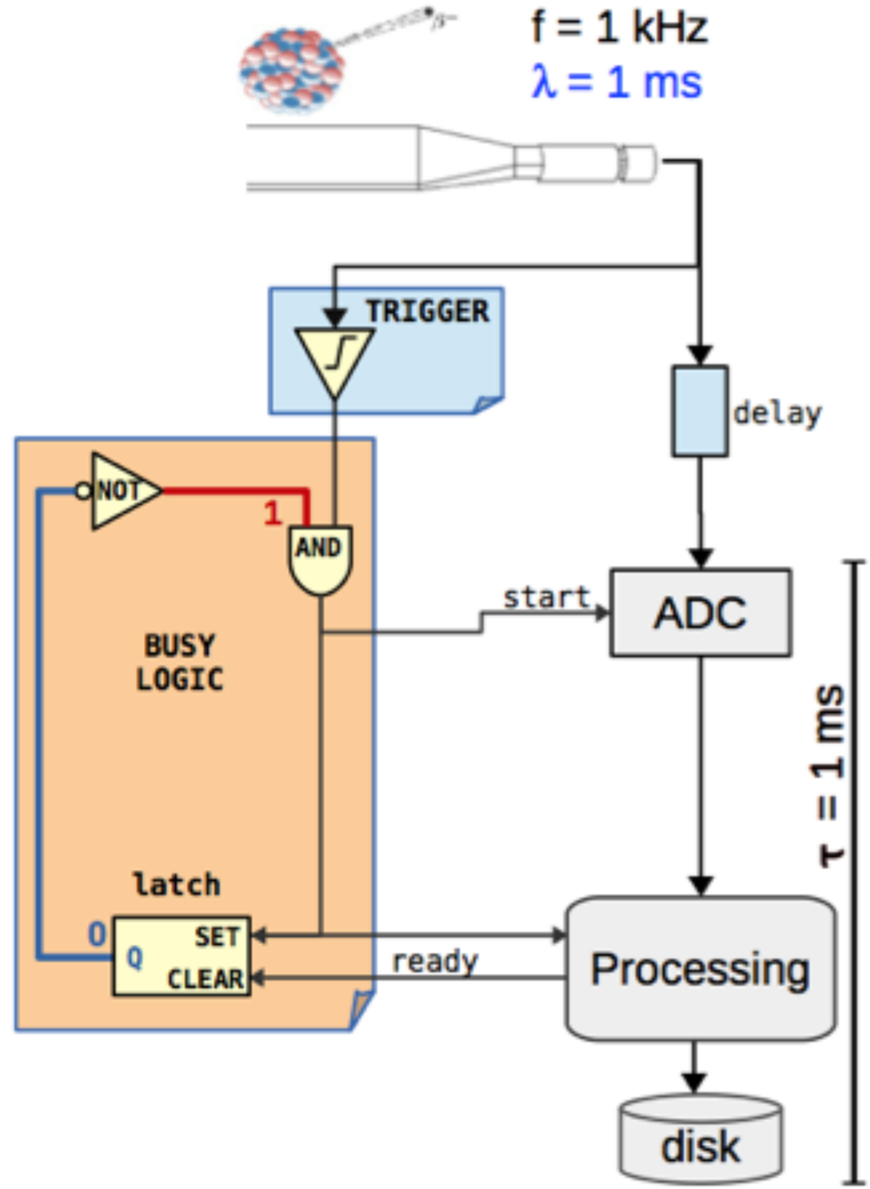
- but waiting time can be much longer or shorter!

lose the preceding event?

crash the system?

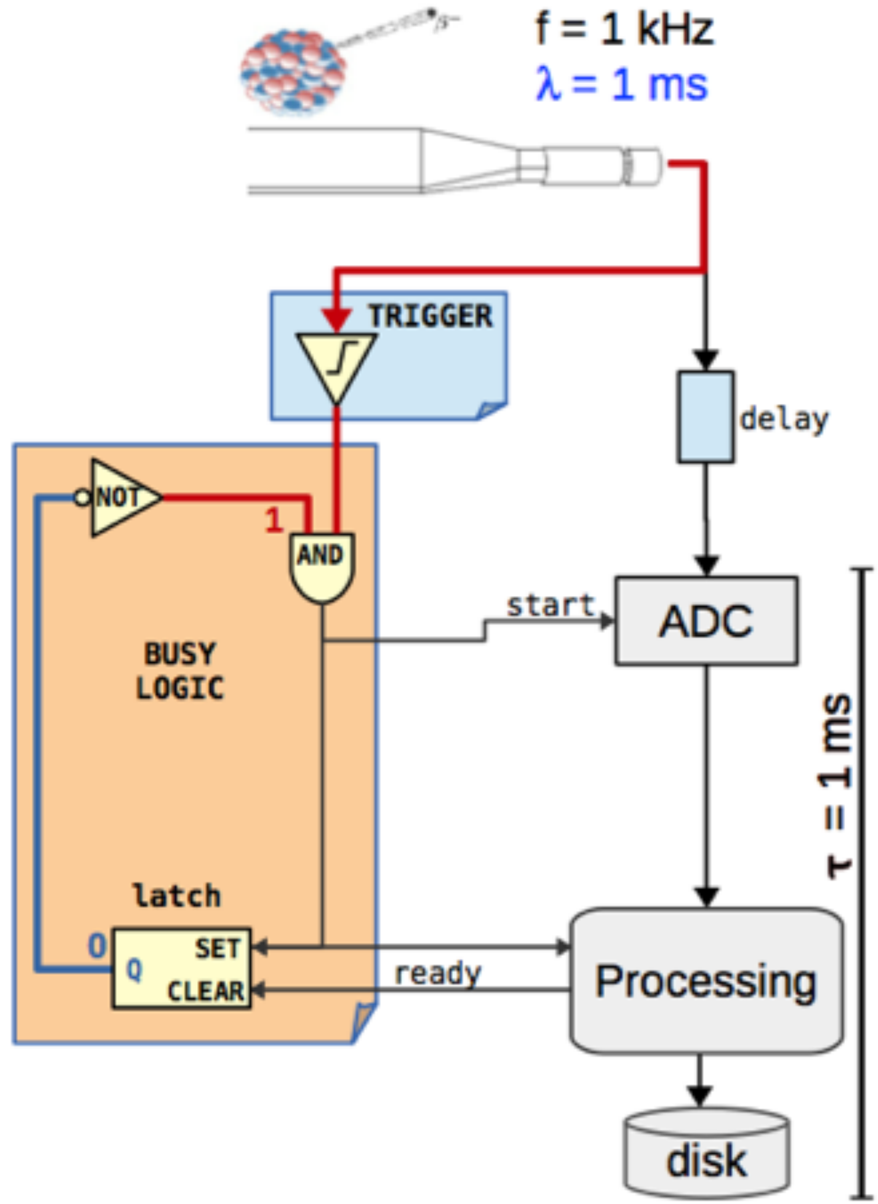
leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - at first flip-flop state is "low" (zero) and so its negated input to the AND is "high" (one)



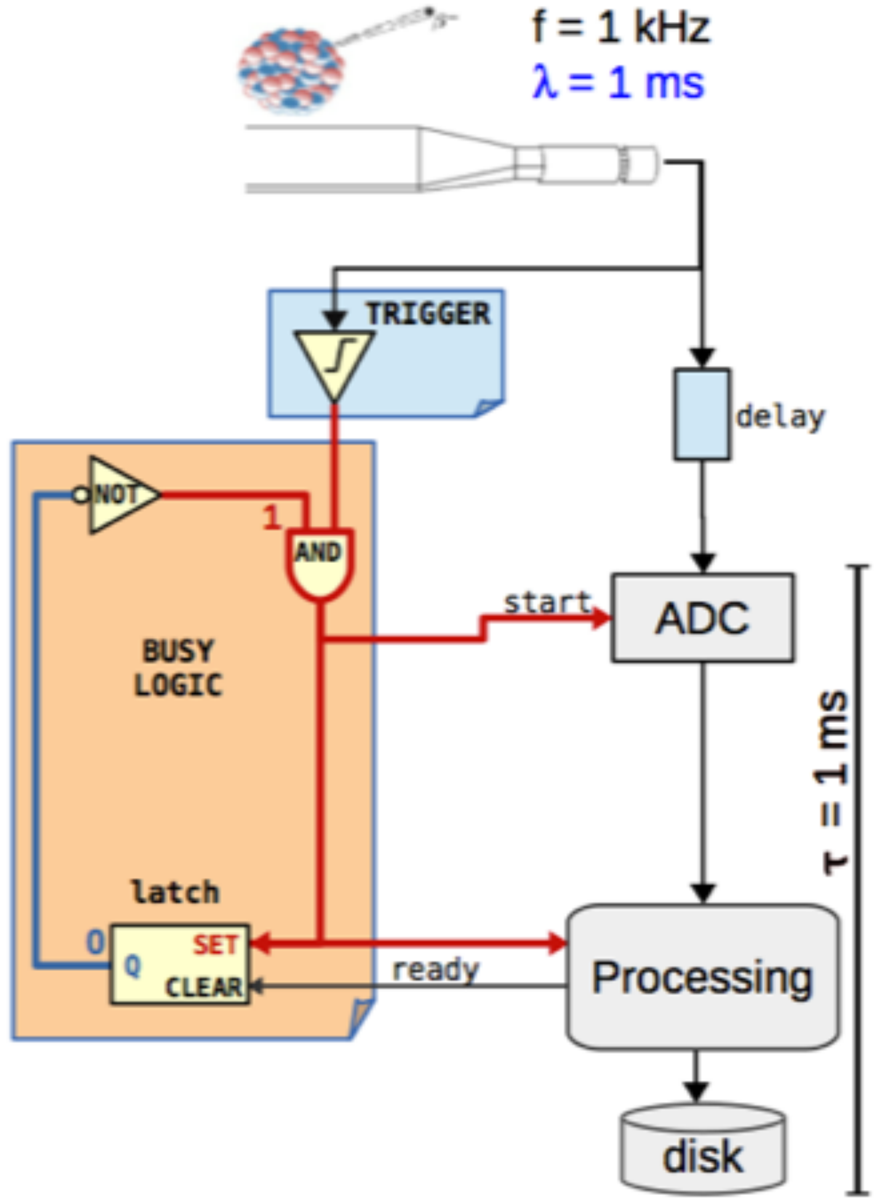
leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - when a trigger arrives, it can pass the "AND"



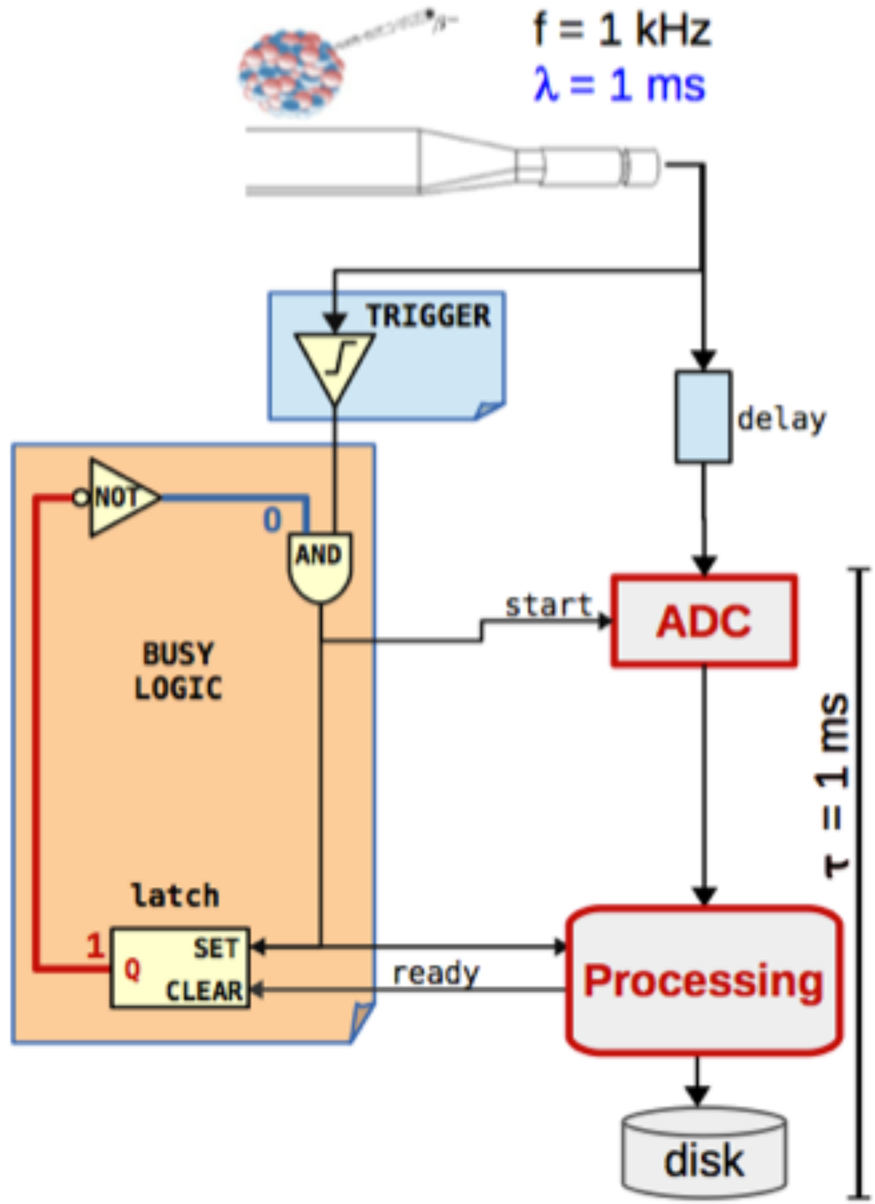
leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - when a trigger arrives, it can pass the "AND"
  - → ADC and processing start, flip-flop is switched



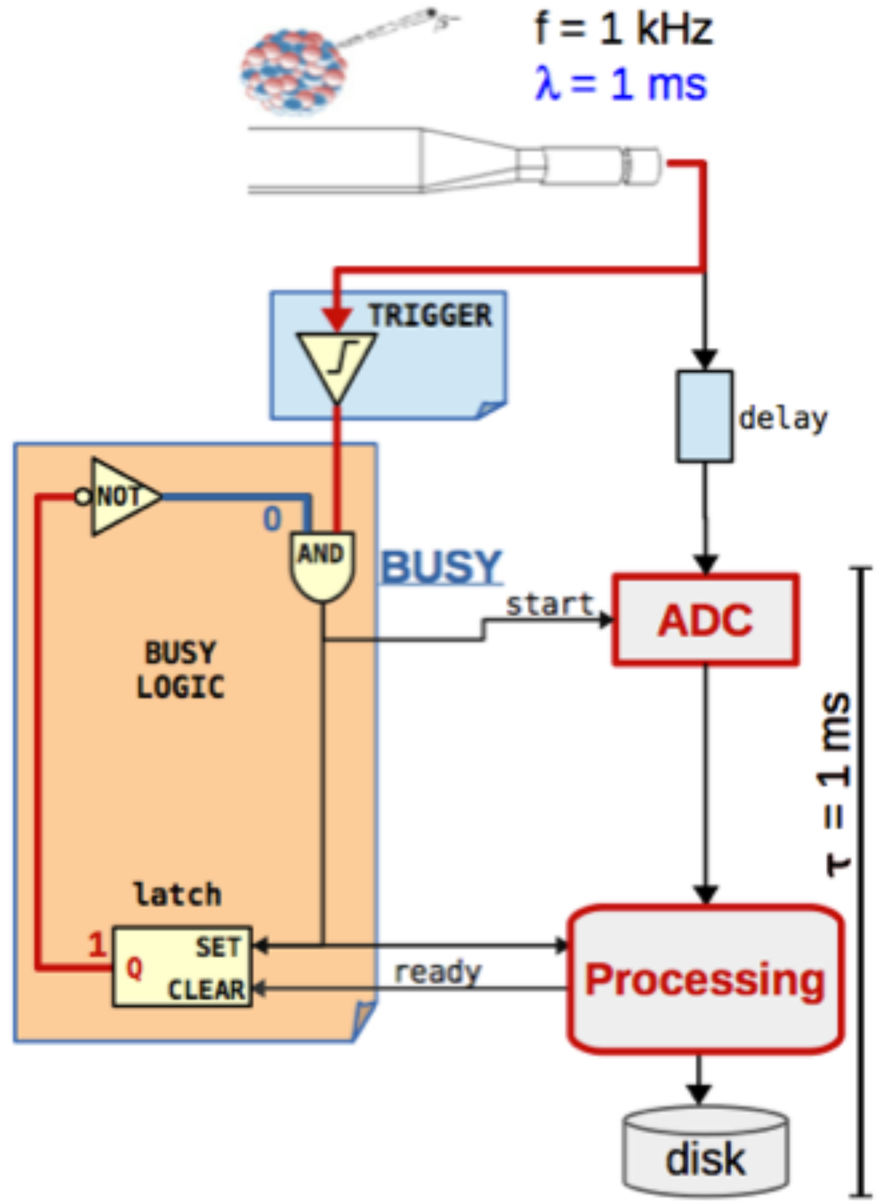
leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - negated flip-flop signal at AND is "low", no new triggers can pass



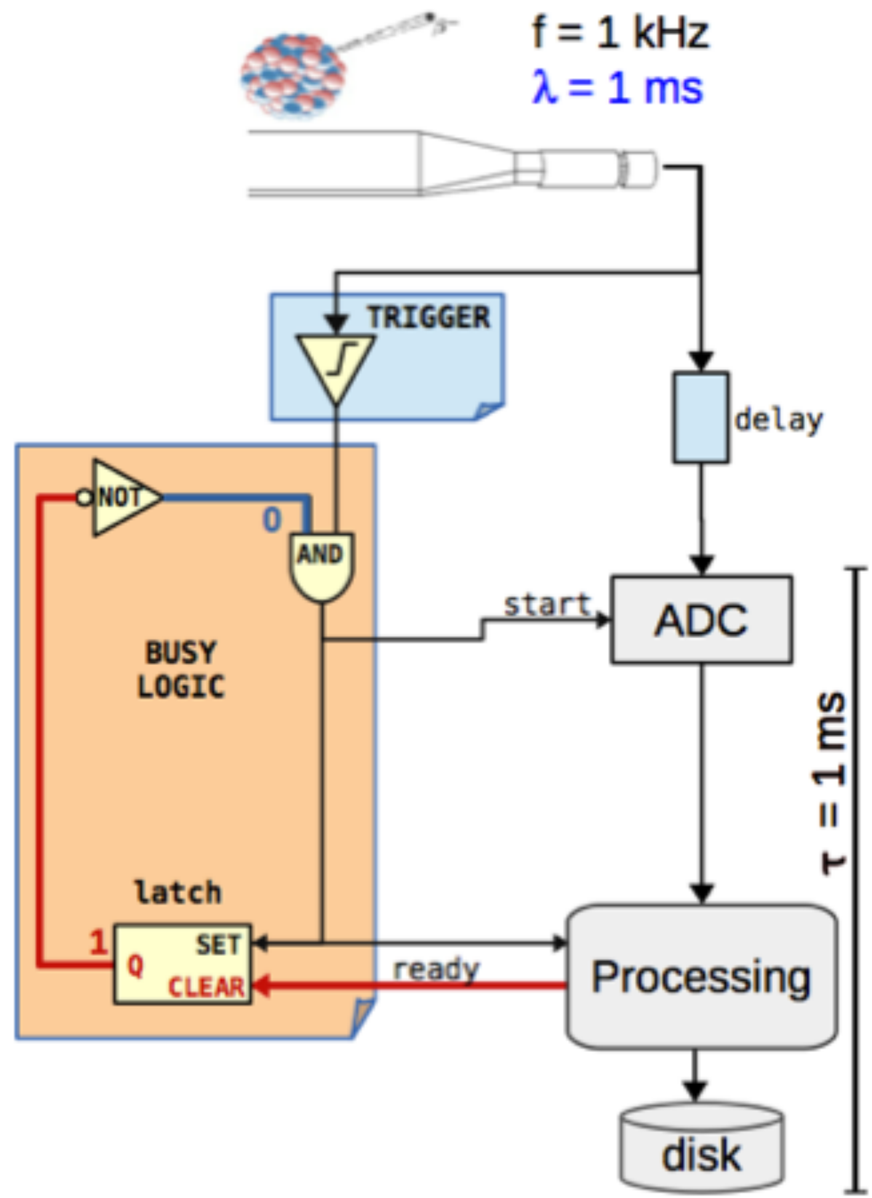
leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - negated flip-flop signal at AND is "low", no new triggers can pass: in other words, the system asserts "BUSY"



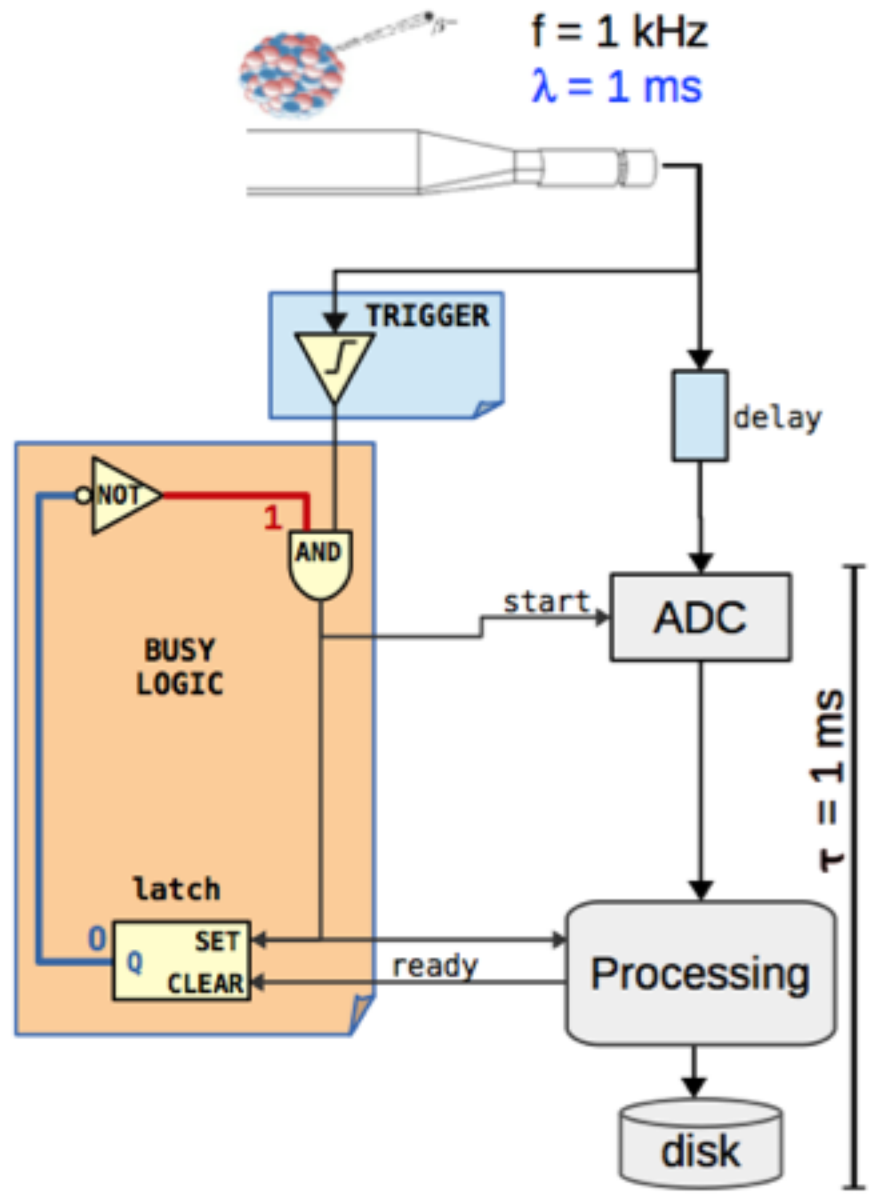
leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - when processing is done, the flip-flop is reset



leave me alone – I'm BUSY !

- **BUSY logic** avoids triggers while the system is busy in processing
  - e.g., AND port and latch
- latch (**flip-flop**):
  - a bi-stable circuit that changes state (Q) by signals applied to the control inputs (SET, CLEAR)
  - when processing is done, the flip-flop is reset to zero, and its negated output ("1") opens the AND-gate again for the next trigger



deadtime and trigger efficiency

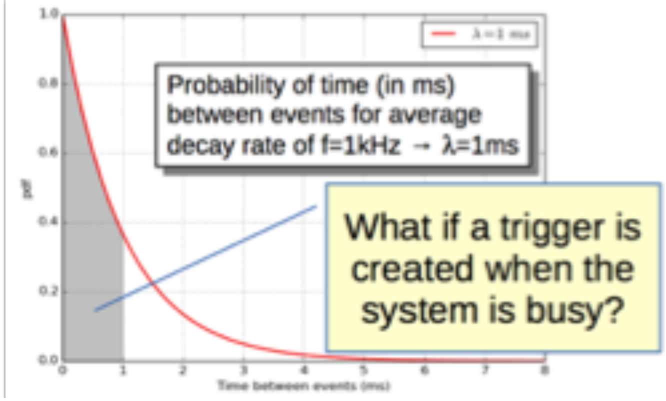
- with "clock trigger" (= untriggered readout):
- e.g. digitization interval τ = 1 ms → readout rate = 1 kHz
  - readout rate = sampling rate

- using a "real" trigger:

  definitions:
  - f: average input rate (physics events)
  - ν: average output rate (DAQ = data acquisition system)
  - τ: deadtime (time needed to process an event)
  - probability for "BUSY": $P(busy) = \nu\tau$
  - probability for "not BUSY": $P(ready) = 1 - \nu\tau$
- $\nu = f\,P(ready)$
- → $\nu = f\,(1 - \nu\tau)$
- → $\nu = f\ /\ (1+f\tau)$

deadtime and trigger efficiency

- events come at irregular intervals (stochastic fluctuations)
  → DAQ rate < event rate : $\nu = f \;/\; (1+f\tau) \;<\; f$
  → efficiency due to DAQ : $\varepsilon = \nu/f = 1 \;/\; (1+f\tau) \;<\; 100\%$
  – e.g. $f = 1$ kHz, $\tau = 1$ ms → $\nu = 0.5$ kHz, $\varepsilon = 50\%$

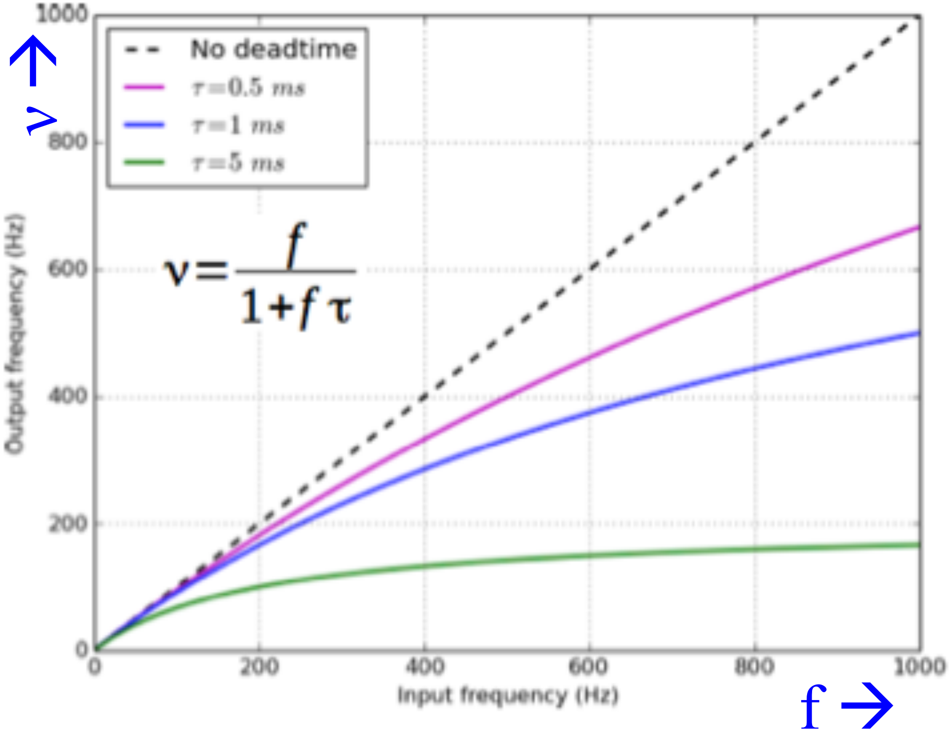


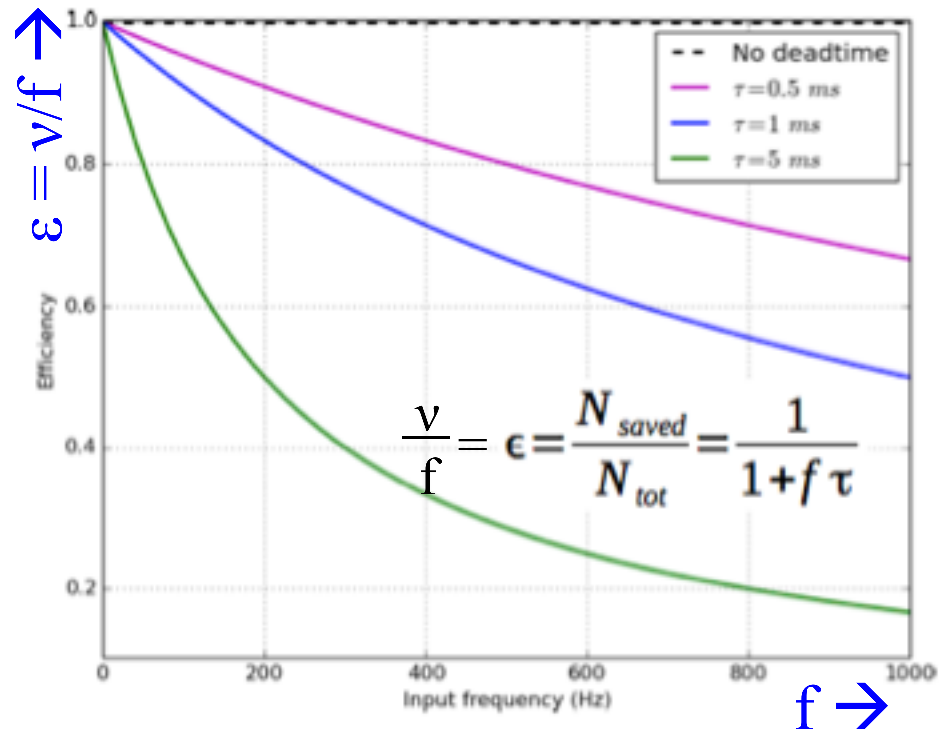
definitions:
– f: average input rate (physics events)
– ν: average output rate (DAQ)
– τ: deadtime (time needed to process an event)
– probability for "BUSY": P(busy) = $\nu\tau$
– probability for "not BUSY": P(ready) = $1 - \nu\tau$

- $\nu = f\,P(ready)$
- → $\nu = f\,(1 - \nu\tau)$
- → $\nu = f \;/\; (1+f\tau)$

HEPHY Institute of High Energy Physics

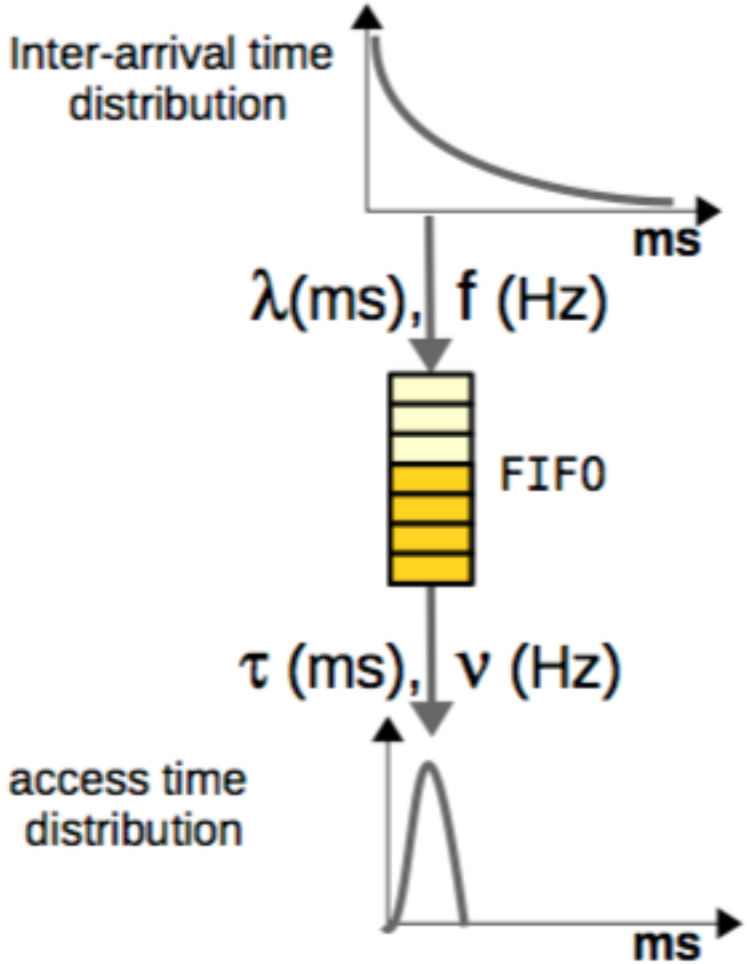
deadtime and trigger efficiency





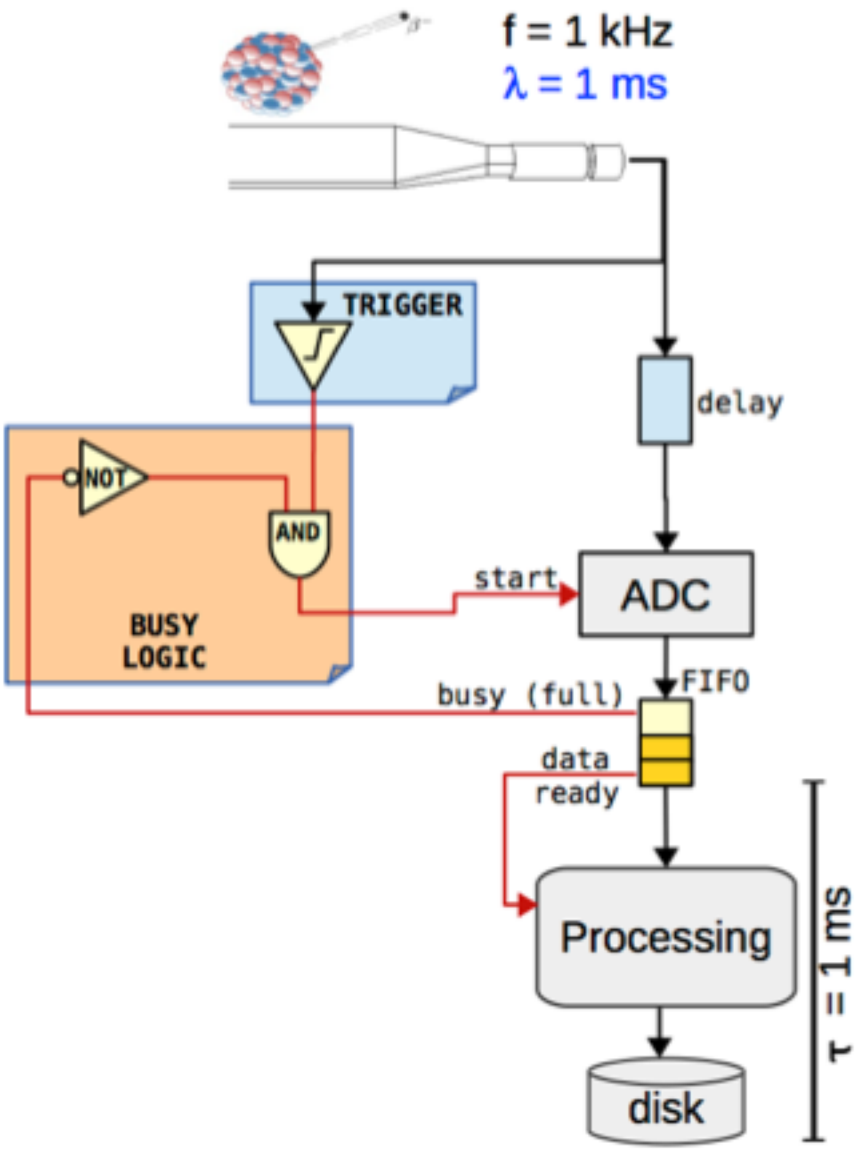
- in order to obtain $\varepsilon \sim 100\%$ ($\nu \sim f$) → $f\tau \ll 1$ → $\tau \ll 1/f$
  - $\varepsilon \sim 99\%$ for $f = 1$ kHz → $\tau < 0.01$ ms → $1/\tau > 100$ kHz
- to cope with the input signal fluctuations, we have to over-design our DAQ system by a factor of 100 ! ☹
- any clever ideas?

"de-randomization"



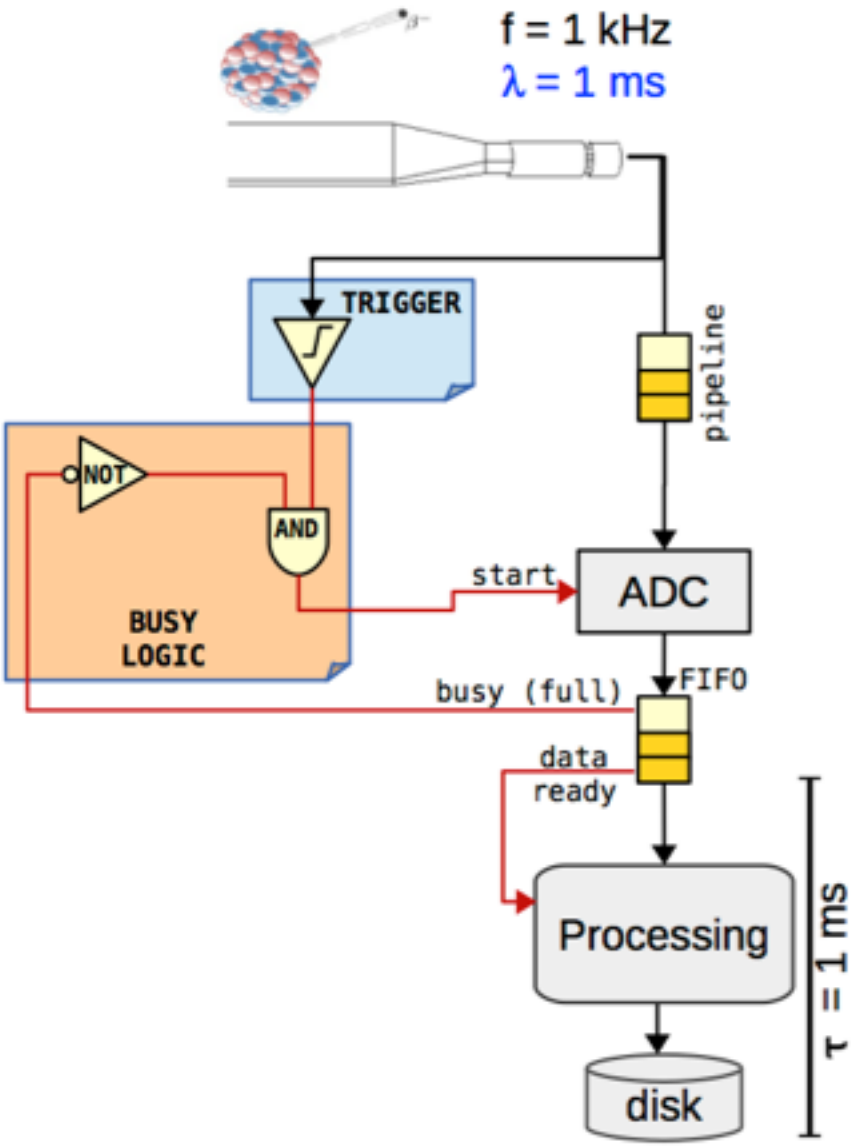
- fluctuations in arrival time absorbed by queue
- FIFO
  - first in, first out
  - "de-randomized" output rate
- additional latency

de-randomized DAQ with FIFO



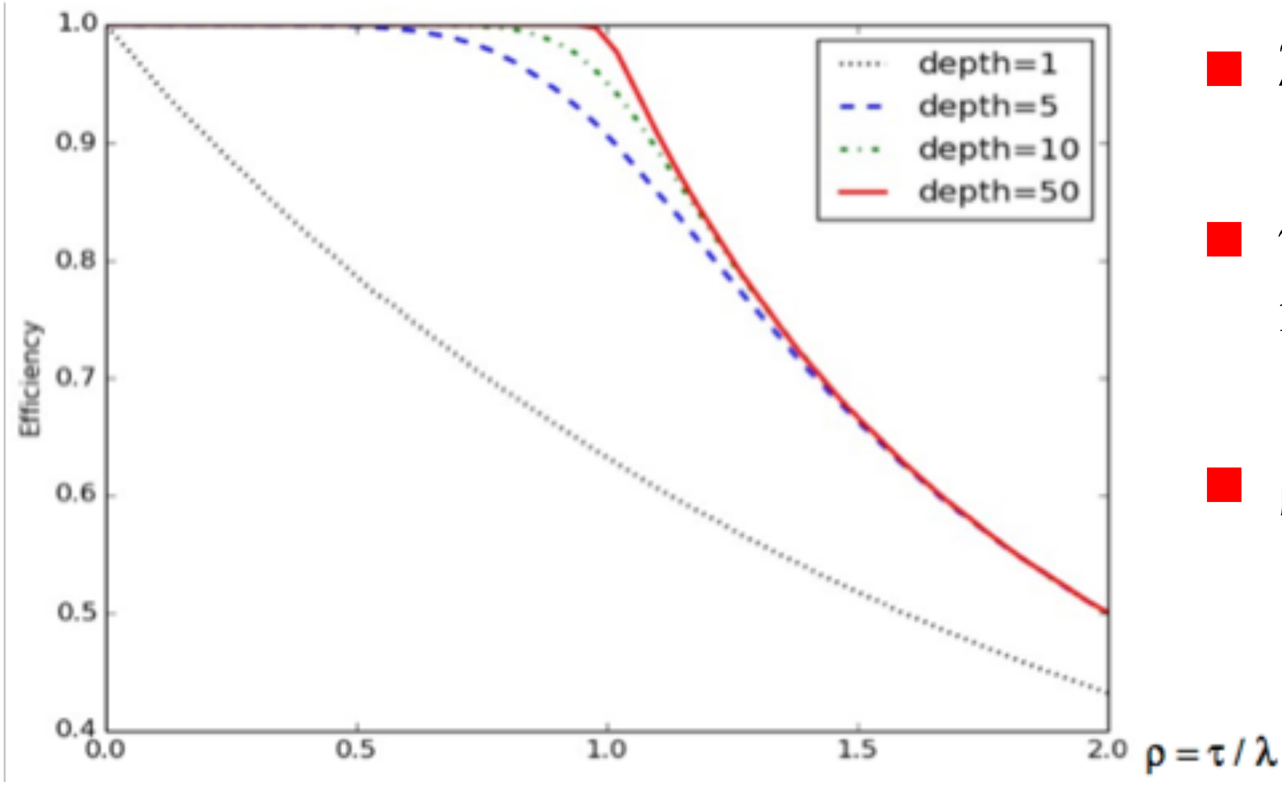
- can achieve high efficiency
  - small deadtime
  - ADC much faster than input rate
  - data processing at input rate
- ... and what if the ADC is challenged by the data rate?
  - could we put a buffer somewhat like a FIFO *before* the ADC?

analog pipeline



- analog pipeline before ADC
  - de-randomizing also the digitization step

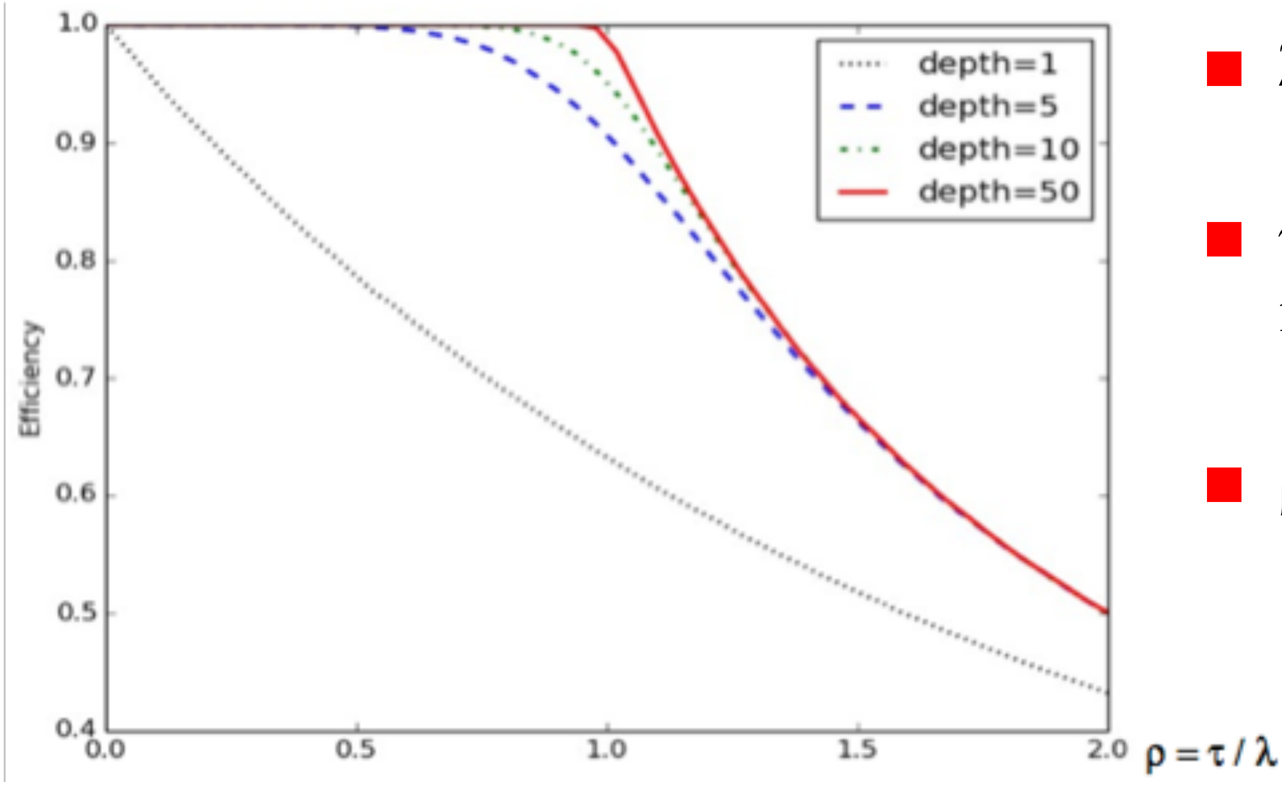
queuing theory



- λ ... event interval
  - at input
- τ ... processing interval
  - at output
- $\rho = \tau / \lambda$

- which value of ρ is best?

queuing theory

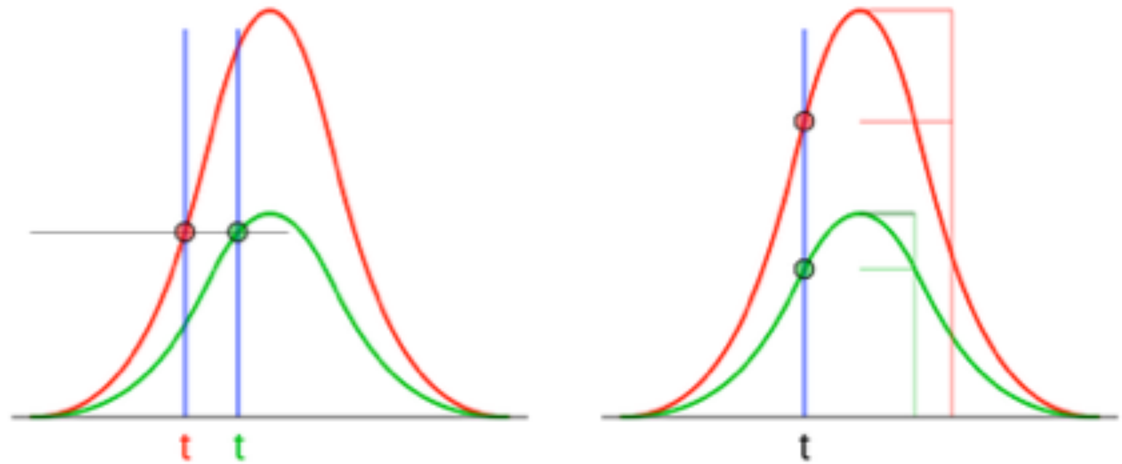


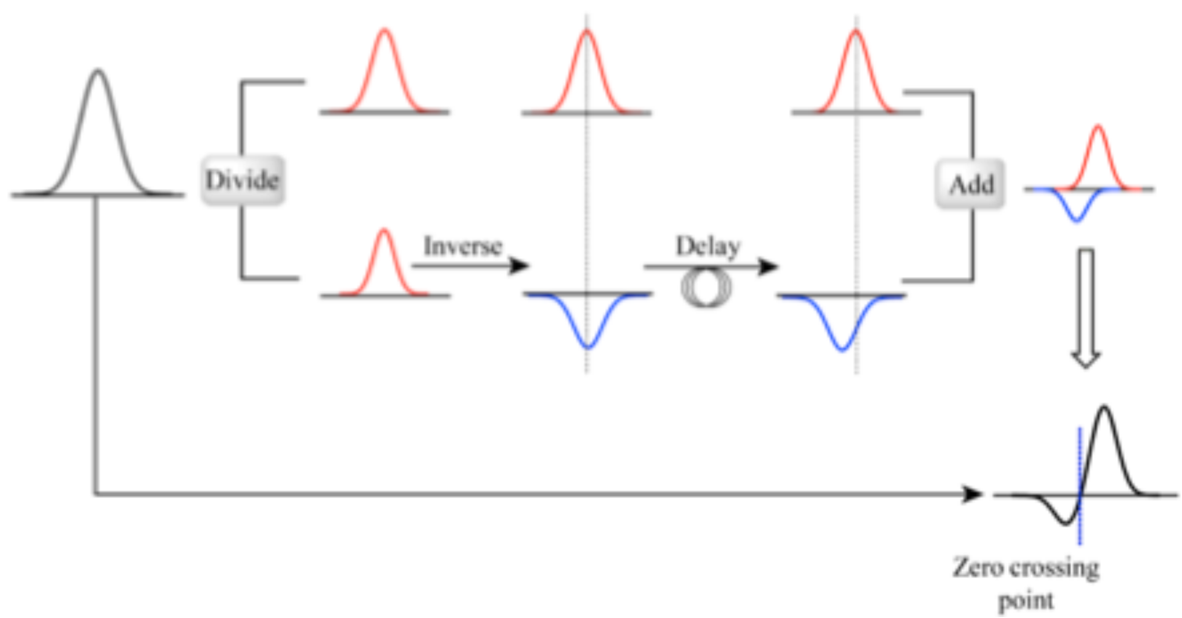
- λ ... event interval
  - at input
- τ ... processing interval
  - at output
- $\rho = \tau / \lambda$

- $\rho > 1$: system is overloaded (cannot cope with input rate)
- $\rho << 1$: system faster than needed (over-design, waste of money)
- $\rho \sim 1$: optimum design

time walk: constant-fraction discriminator (CFD)

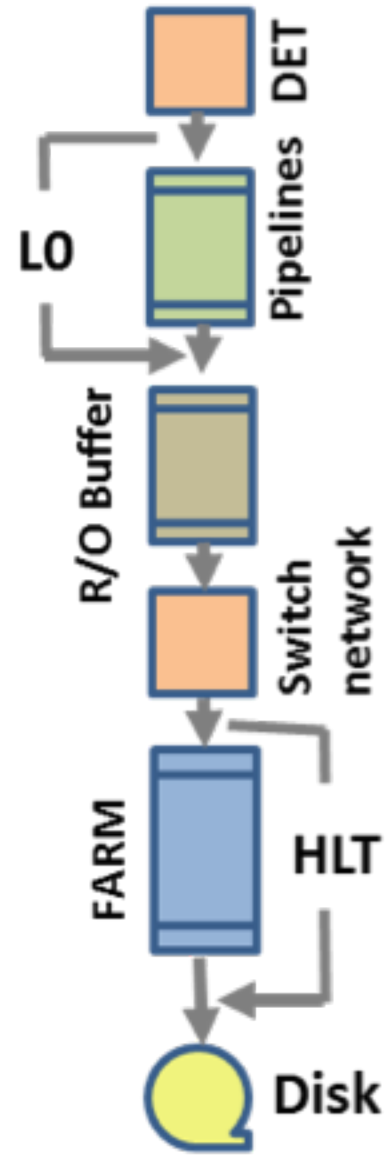
- fixed threshold: dependence of the trigger time on the signal's peak height
- constant fraction of total height → independent of signal size
- achieved in electronics by dividing, inverting, delaying the signal
- measure time at zero-crossing





multi-level trigger

- first triggering criteria may be very simple
  - e.g., just wait for ADC (digitizer) to be ready
  - or simple selection criteria (such as minimum signal strength)
- additional triggering criteria may involve complicated calculations
- → benefit from multi-level trigger
  - first levels easy and quick
  - later levels complicated and more time-consuming
- first levels already remove many events → later, more complex levels face a smaller event frequency to cope with



Level-1 Trigger

Second-Level Trigger

Data analysis

latency: *make up your mind quickly – or she will be gone!*

triggering at a particle collider

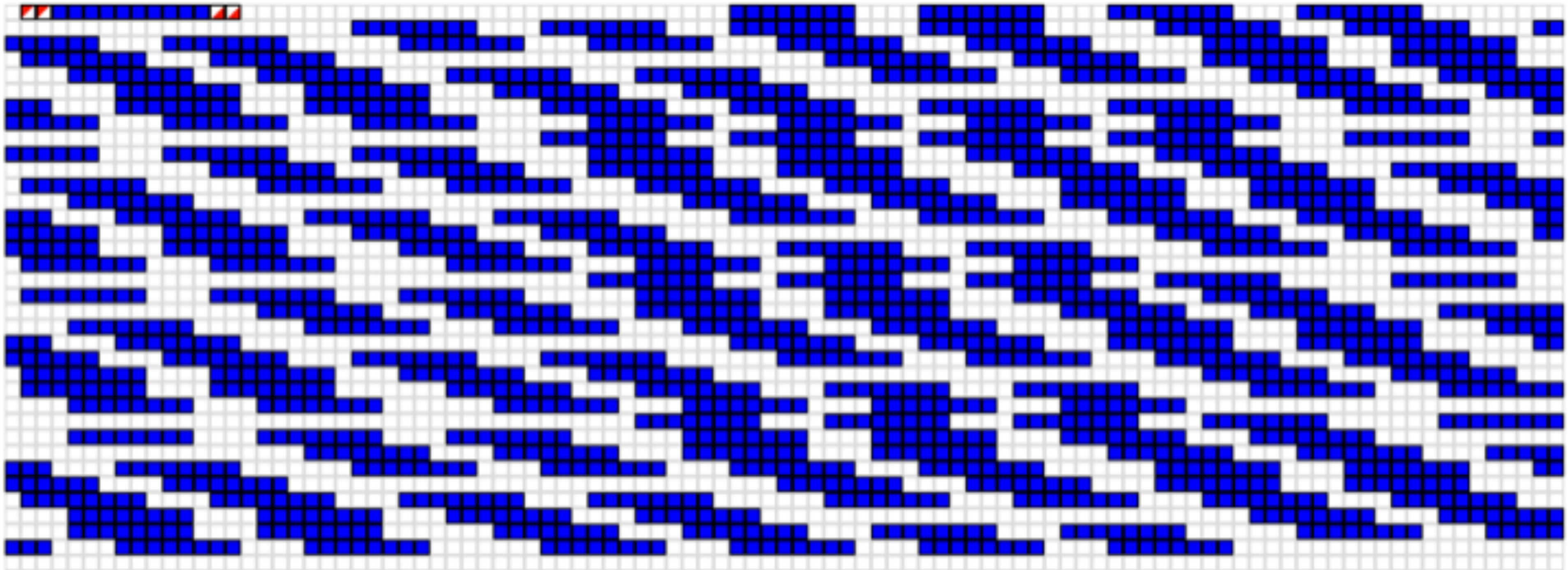
- particle colliders are usually "synchrotrons": particles can only arrive at certain times
  - and then, there is usually a lot of them
  - "bunched" structure

- so, maybe we do not need all this queuing business?
  - just record whatever happens when two bunches meet ("bunch crossing")

- any argument against this?

LHC bunch filling scheme

LHC orbit with 3564 "bunch crossings"
(colliding bunches in CMS: blue; single bunches in CMS: red/white):



LHC bunch crossing frequency: ~ 40 MHz (collisions every ~25 ns)
25 ns ~ 7.5 meters → LHC circumference: 3564 * 7.5 m ~ 27 km

bunch crossing frequency

- is it constant?
  - why yes, why no?
  - what would it be for different kinds of particles (protons, electrons, heavy ions)
- if it is not constant – is this a problem?
  - why?
  - how can it be overcome?
- what do we mean by “accelerator”?
  - does it make particles faster?
  - by how much?
  - what else does it achieve?

LHC bunch crossing frequency

- proton energy at injection: 450 GeV
- proton energy at collision: 6800 GeV
- speed is slightly different
  - homework: calculate the speed of the protons!
- tunnel length is fixed → have to vary frequency
  - homework: by how much?
- electronics (in front-ends, trigger, DAQ) must be able to cope with this
  - then we can take the collider frequency as unit – although it is not quite constant
  - just plot and calculate everything in terms of “bunch crossings” – “BX”


CMS
CMS Experiment at LHC, CERN
Data recorded: Mon May 28 01:16:20 2012 CEST
Run/Event: 195099 / 35438125
Lumi section: 65
Orbit/Crossing: 16992111 / 2295
pile-up of events

triggering at a particle collider

- particle colliders are usually "synchrotrons": particles can only arrive at certain times
  - and then, there is usually a lot of them
  - "bunched" structure

- so, maybe we do not need all this queuing business?
  - just record whatever happens when two bunches meet ("bunch crossing")

- any argument against this?

triggering at a particle collider

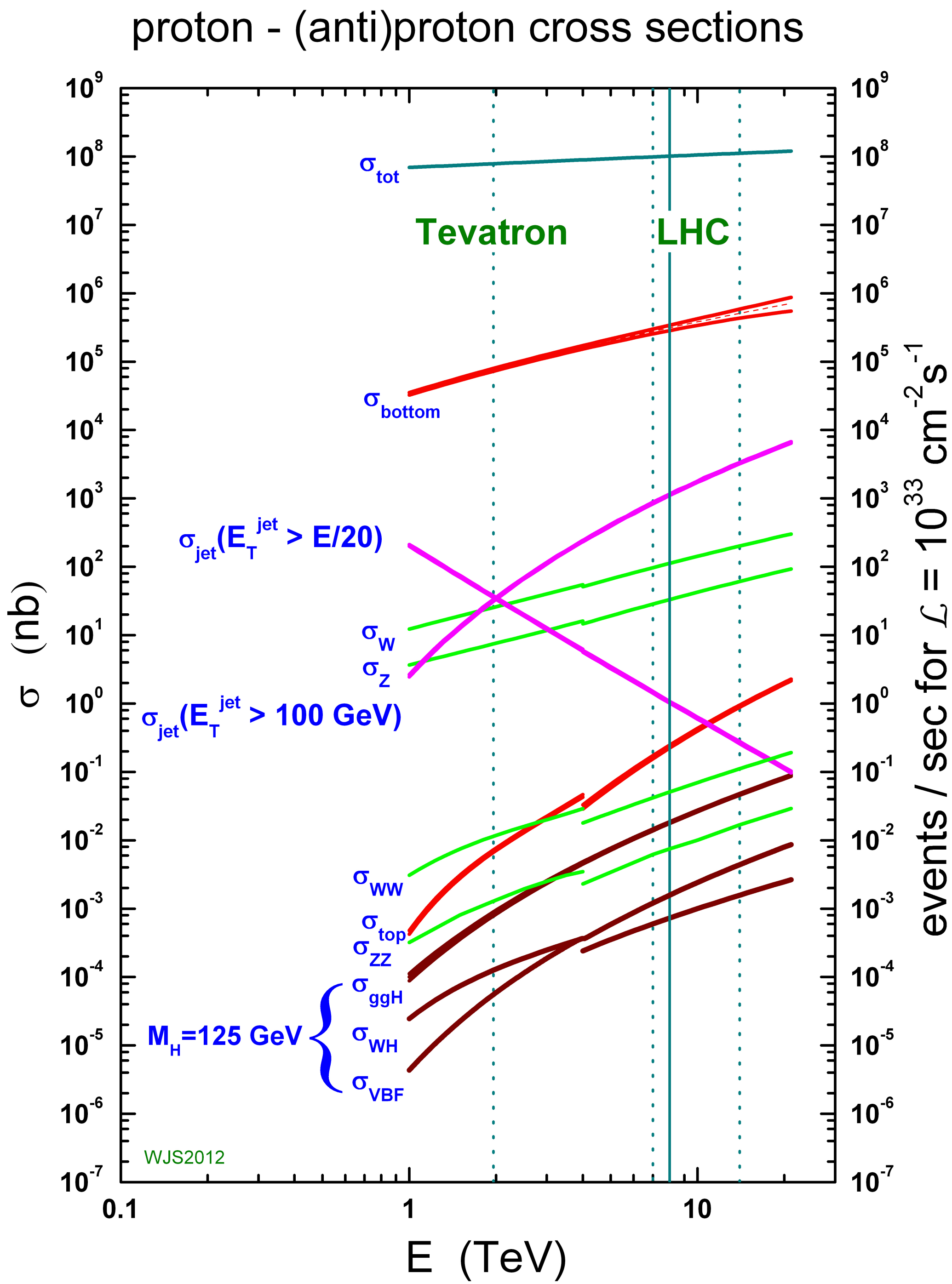
- particle colliders are "synchrotrons": particles can only arrive at certain times
  - and then, there is usually a lot of them
  - "bunched" structure

- so, maybe we do not need all this queuing business?
  - just record whatever happens when to bunches meet ("bunch crossing")

- a system recording each bunch crossing may be too complicated
  - cannot even retrieve data fast enough from detector – even if I could afford enough computers for the processing!
    - too many cables, too much power, too many cooling lines ...
  - and on the other hand, in most bunch crossings nothing interesting may be happening

When do we trigger ?

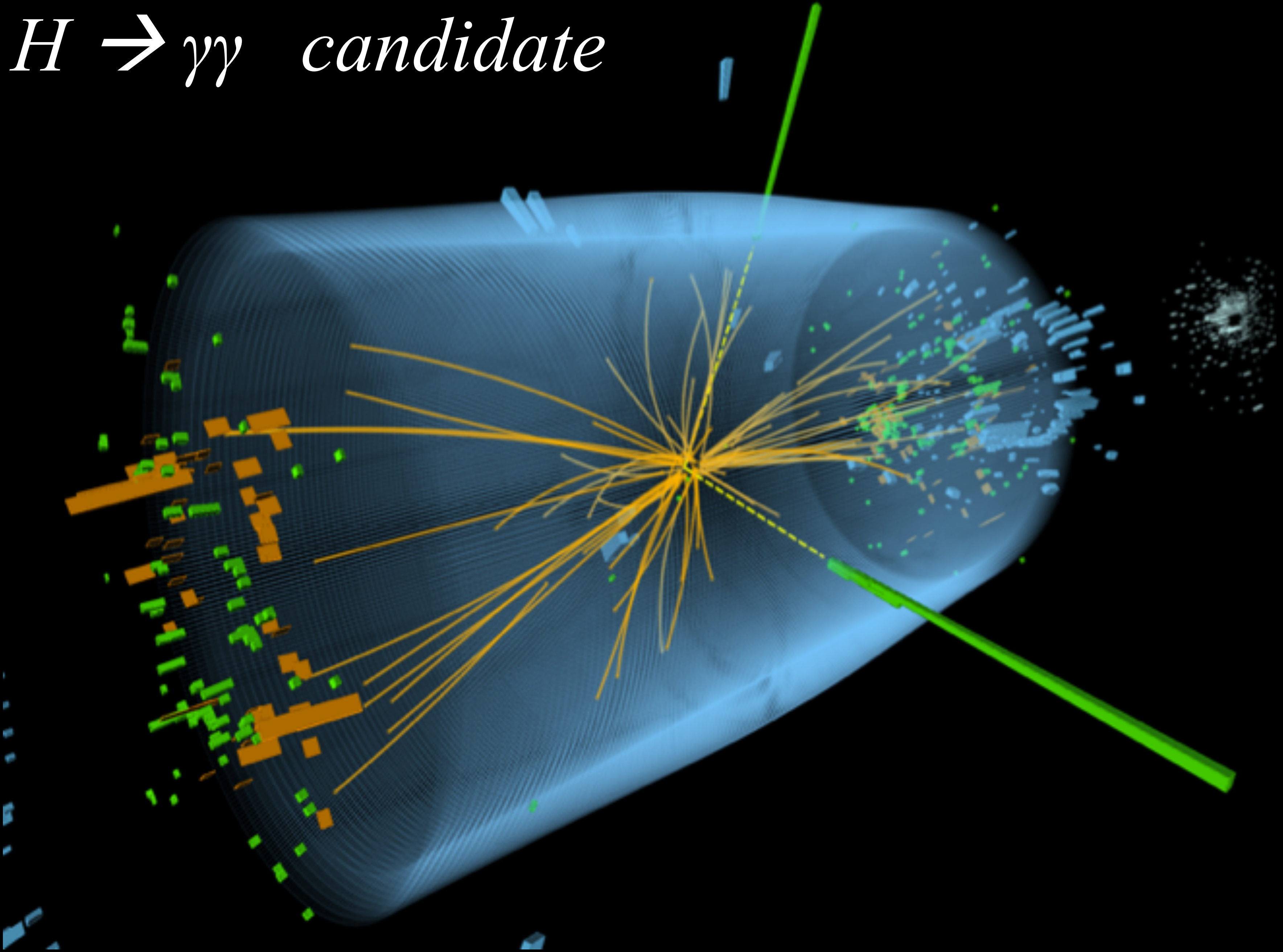
- „bunch" structure of the LHC collider
  - „bunches" of particles
  - 40 MHz
    - a bunch arrives every 25 ns
    - bunches are spaced at 7.5 meters from each other
    - bunch spacing of 125 ns for heavy-ion operation
- at present luminosity of the LHC collider (> 2 $*10^{34}$ cm^{-2} s^{-1}) we have about 60 proton-proton interactions for each collision of two bunches
  - only a small fraction of these "bunch crossings" contains at least one collision event which is potentially interesting for searching for "new physics"
  - in this case all information for this bunch crossing is recorded for subsequent data analysis and background suppression
  - luminosity quoted for ATLAS and CMS
    - reduced luminosity for LHCb (b-physics experiment)
    - heavy-ion luminosity much smaller

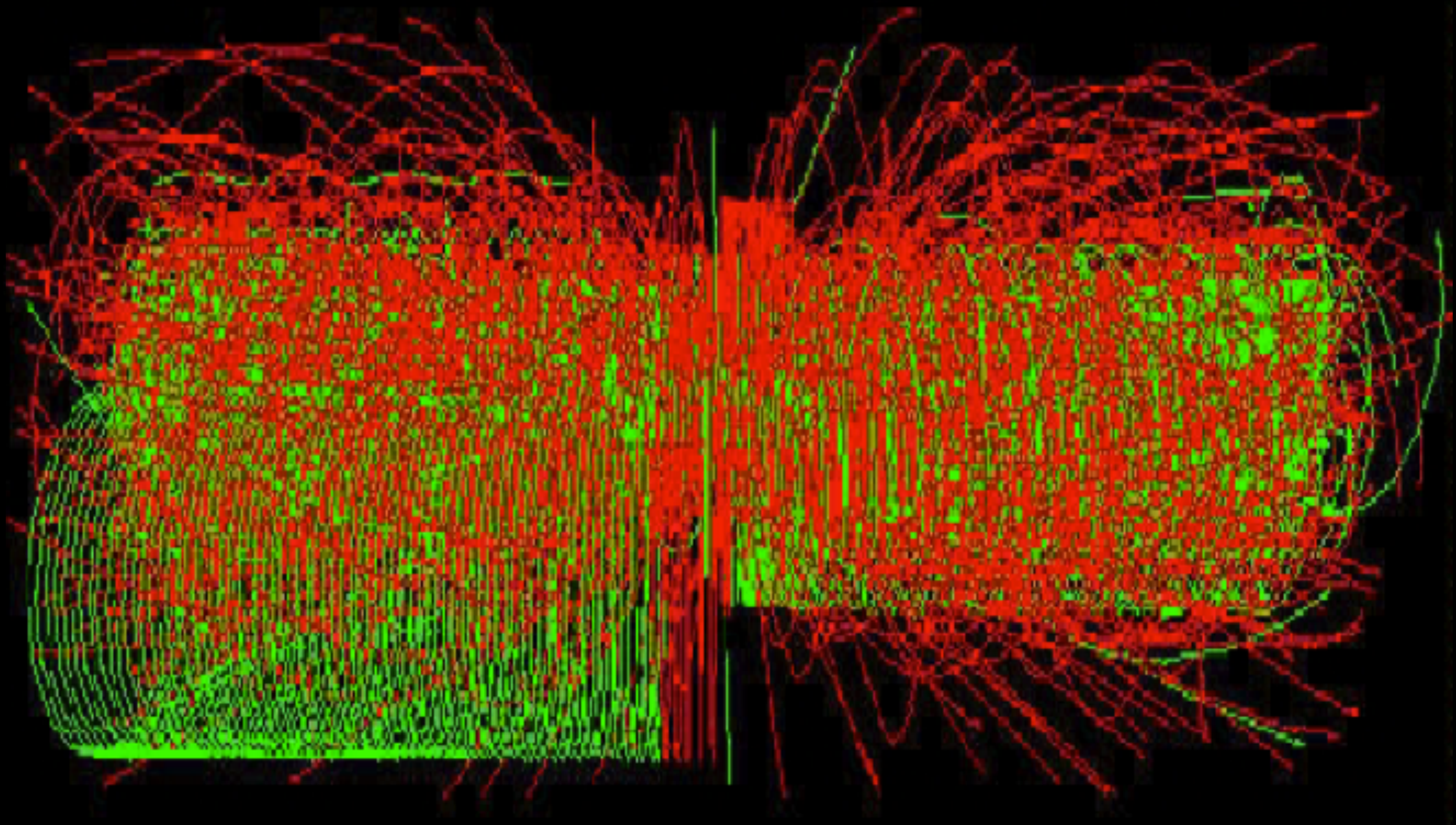


Cross sections (relative probabilities) of processes at LHC

- due to the extremely small cross sections of processes now under investigation it is impossible to check all events "by hand"
  - ~ 10^{12} background events to one signal event
- it would not even be possible to read out and record all data in computer memories
- we need a fast, automated decision ("trigger") for an event to be recorded or not

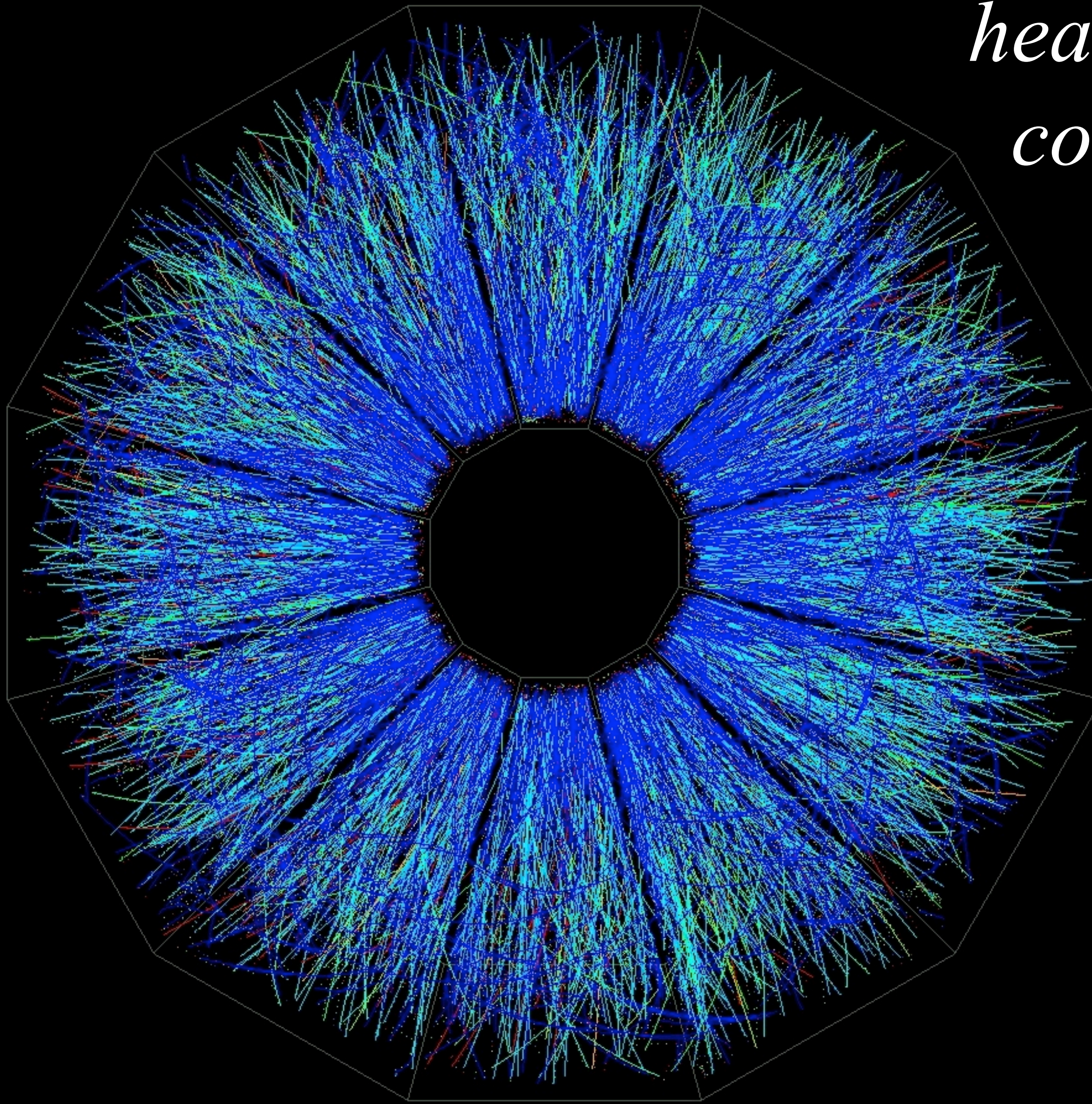
$H \rightarrow \gamma\gamma$ *candidate*





Higgs -> 4μ

+30 MinBias



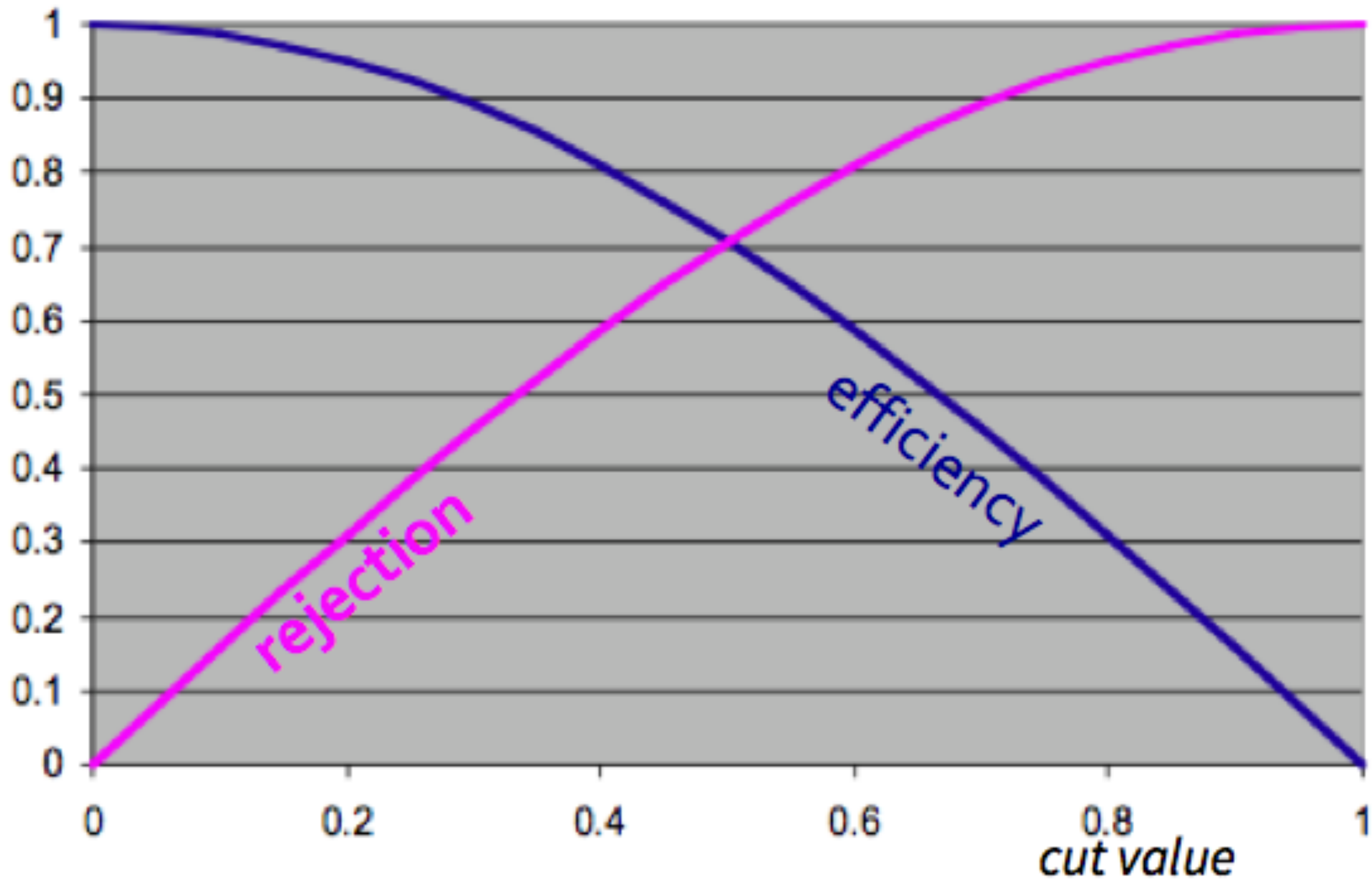
heavy-ion collision

General triggering requirements

- efficiency: retain all (most) good events
  - don't eat the good ones!
- purity: reject all (as many as possible) bad events
  - don't put bad ones into the pot!
- no bias: do not distort the result!
  - even if you lose some good events, take care not to affect the measured values

The good ones go into the pot,
The bad ones go into your crop.

efficiency vs. purity (rejection power)



The risk of throwing the baby out with the bath water!

TRIGGER: HOW?

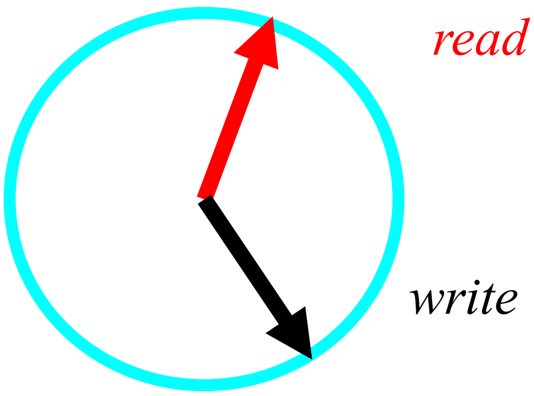
The CMS trigger system

- two-layer trigger setup:
- Level-1 Trigger ("L1")
  - reduce LHC's 40-MHz bunch-crossing rate to 100 kHz
  - hardware based (custom electronics)
  - pipe-lined architecture
  - L1-accept: read out full CMS detector
- High-Level Trigger ("HLT")
  - reduce 100 kHz to a few hundred Hz (1 kHz maximum)
  - computer farm running CMS analysis software

The good ones go into the pot,
The bad ones go into your crop.

trigger with digital pipeline

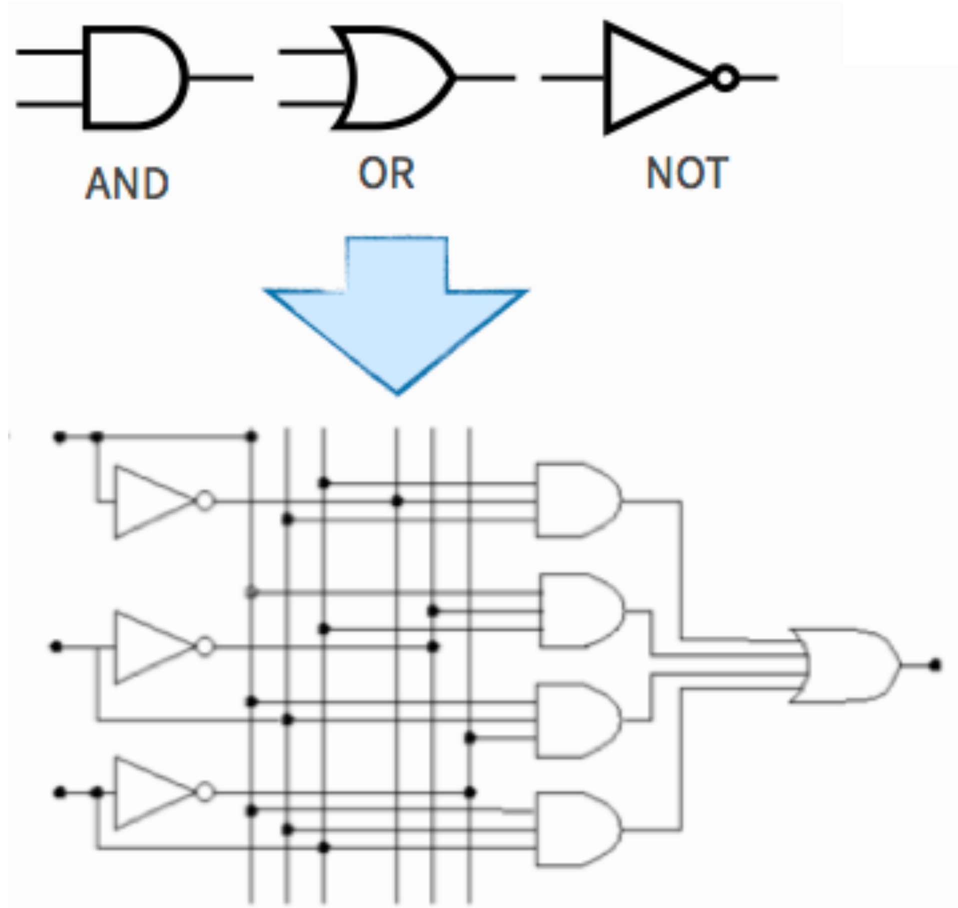
- use as much information about the event as possible
  - allows for the best separation of signal and background
  - ideal case: "complete analysis" using all the data supplied by the detector
- often impossible to read out all detector data
- preliminary decision based on part of the event data only
- be quick!
  - in case of positive trigger decision all detector data must still be available
  - data are stored temporarily in a "pipeline" in the detector electronics
    - "short term memory" in detector front-ends
    - "ring buffer"
    - in hardware, can only afford short pipeline (e.g. in CMS at present: 4 μs)



- how to reconcile these contradictory requirements ?

trigger logic

- decision logic can be described with a sequence of simple logic (mathematical) operators
- straightforward to implement in electronics
  - or, of course, on a computer
  - AND ... &&
  - OR ... ||
  - NOT ... !



lookup tables

- binary logic operators can be described with "truth table"

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

- more complex assignments can be stored in a "lookup table"

A	B	C_{in}	S	C_{out}
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

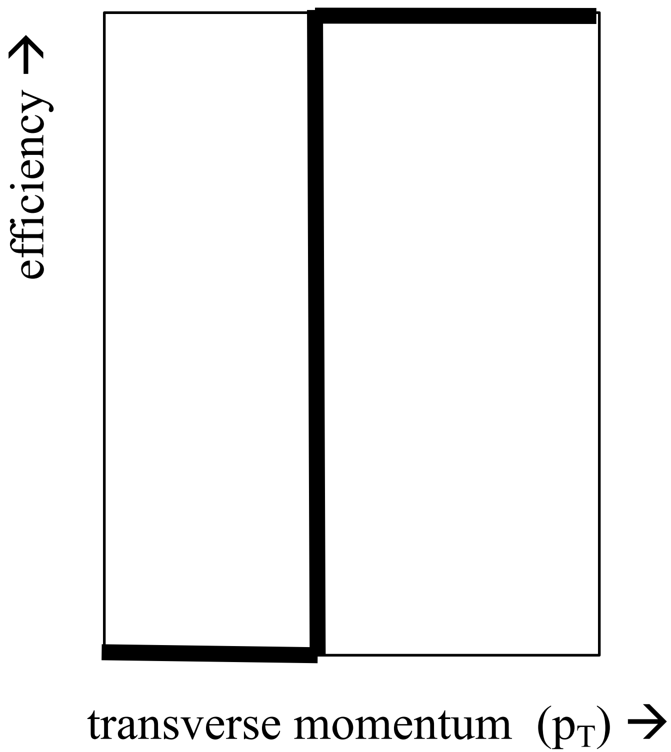
option: special trigger detectors

- in some cases, it may be better to use special, fast but low-resolution trigger detectors
  - e.g., high-resolution detectors may be too slow
  - trigger detector resolution not competitive with other, "precision" detectors → do not use them in final data analysis
  - example: "Resistive Plate Chambers" as muon detectors in ATLAS and CMS

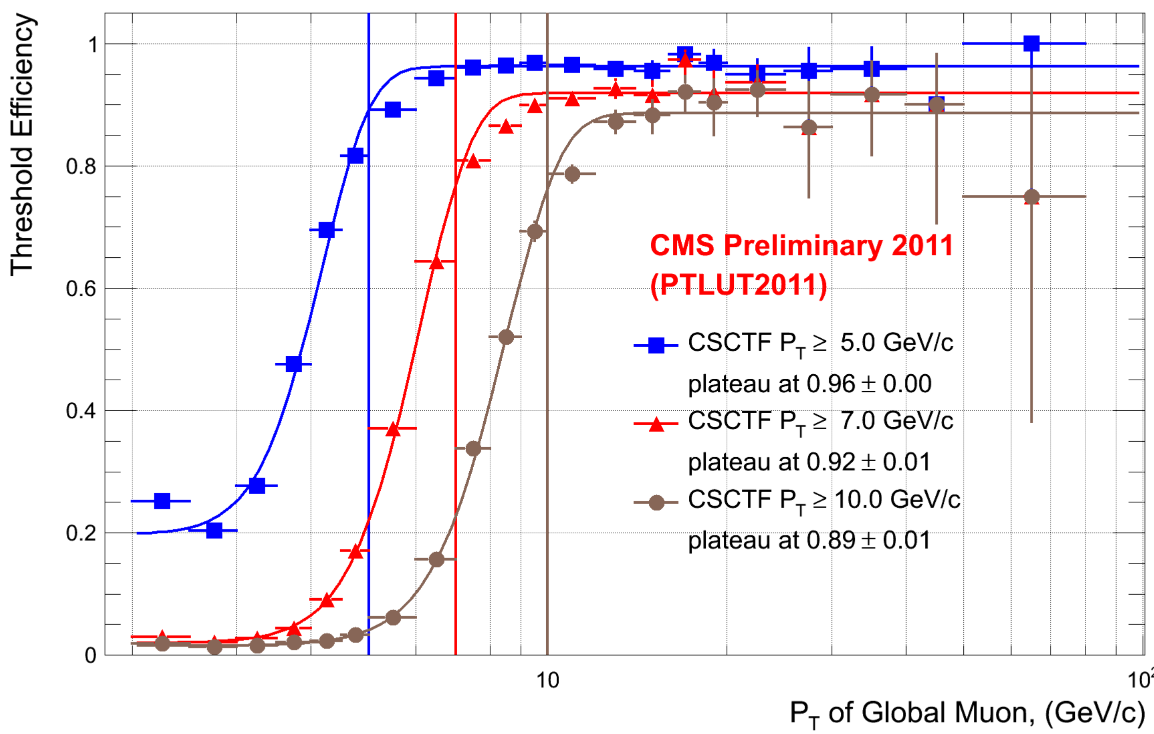
- the other option is to split signals from precision detectors and use the split signals for fast triggering
  - often using analog and/or digital summing over channels
  - speed up processing at cost of accuracy
  - example: "Drift Tubes" and "Cathode Strip Chambers" in CMS

turn-on curves

ideal:

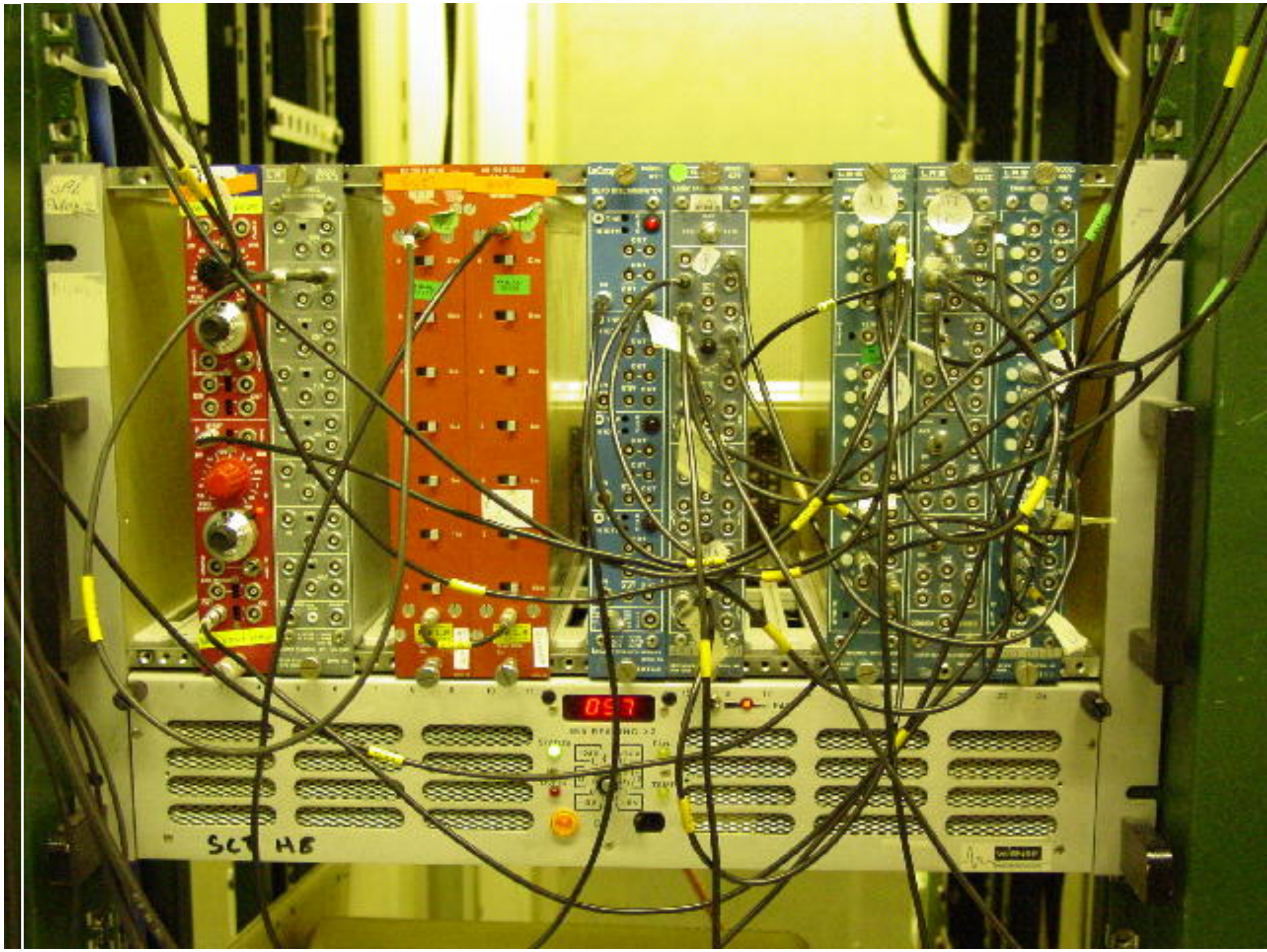


reality:



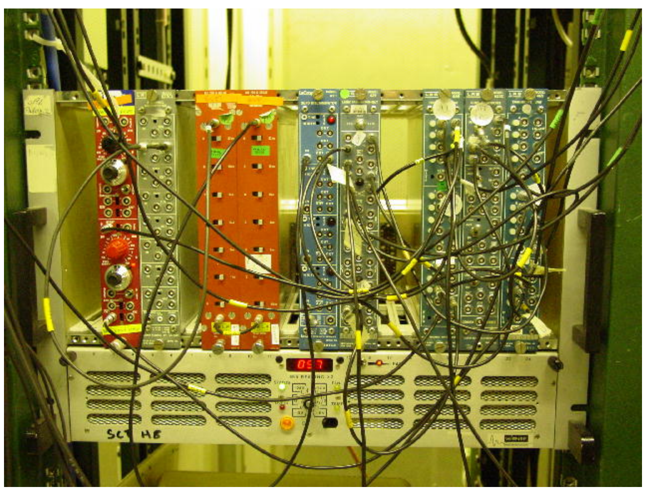
detectors yielding electrical output signals allow to select events to be recorded by electronic devices

- thresholds (discriminators)
- logical combinations (AND, OR, NOT)
- delays
- available in commercial "modules"
- connections by cables ("LEMO" cables)



"NIM" crate

- because of the enormous amounts of data at major modern experiments electronic processing by such individual modules is impractical
  - too big
  - too expensive
  - too error-prone
  - too long signal propagation times
- $\Rightarrow$ use custom-made highly integrated electronic components ("chips")
- stay flexible by using Field-Programmable Gate Arrays (FPGAs)

400 x  $\Rightarrow$ 1x 

~ 10 logical operations / module $\Rightarrow$ ~ 40000 logical operations in one chip

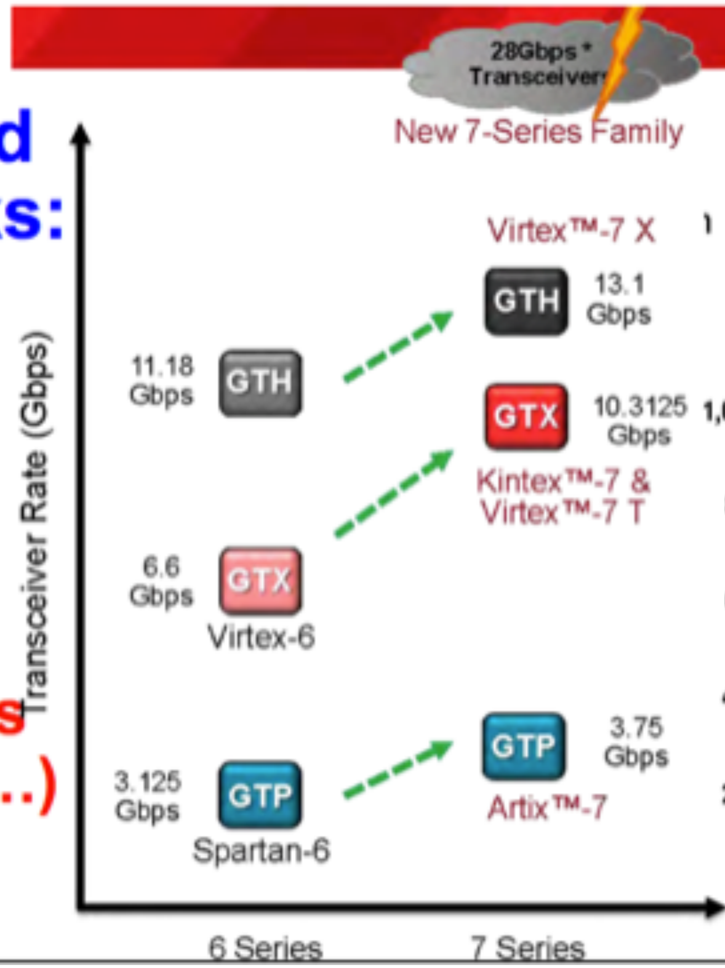
LHC Run 1

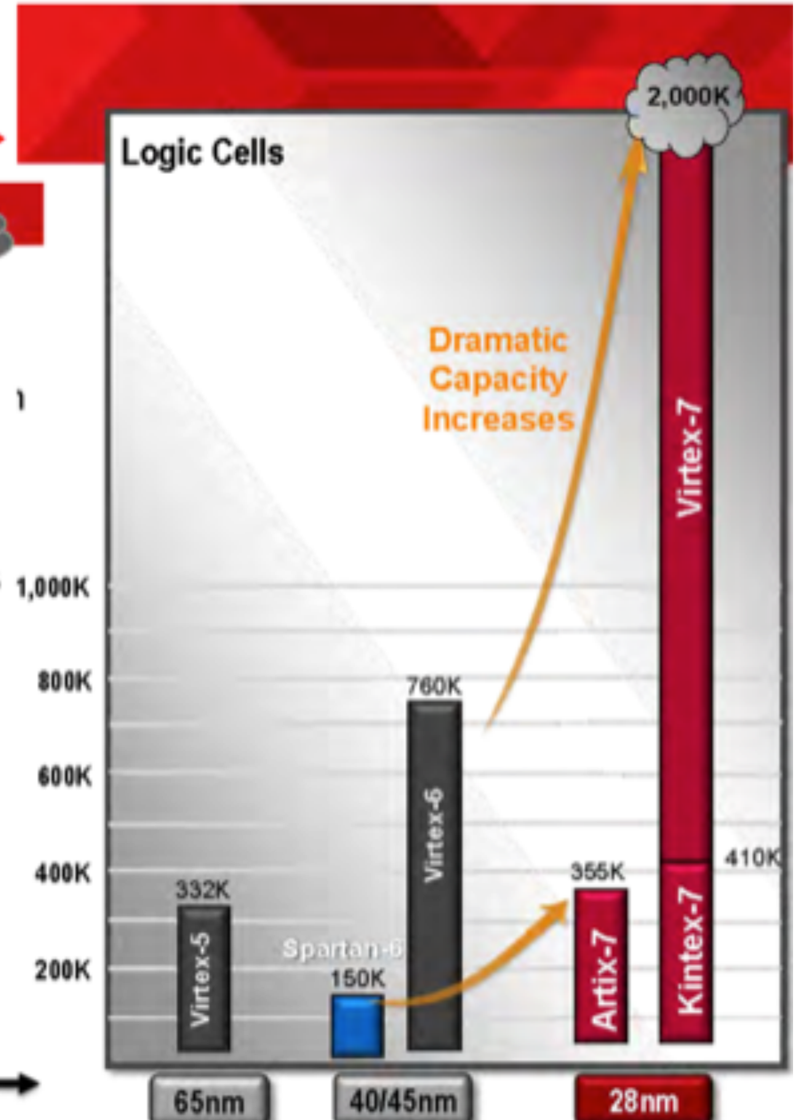
Logic Cells

- ➢28 nm: > 2X gains over 40 nm→

On-Chip High Speed Serial Links:

- ➢Connect to new compact high density optical connectors (SNAP-12…)



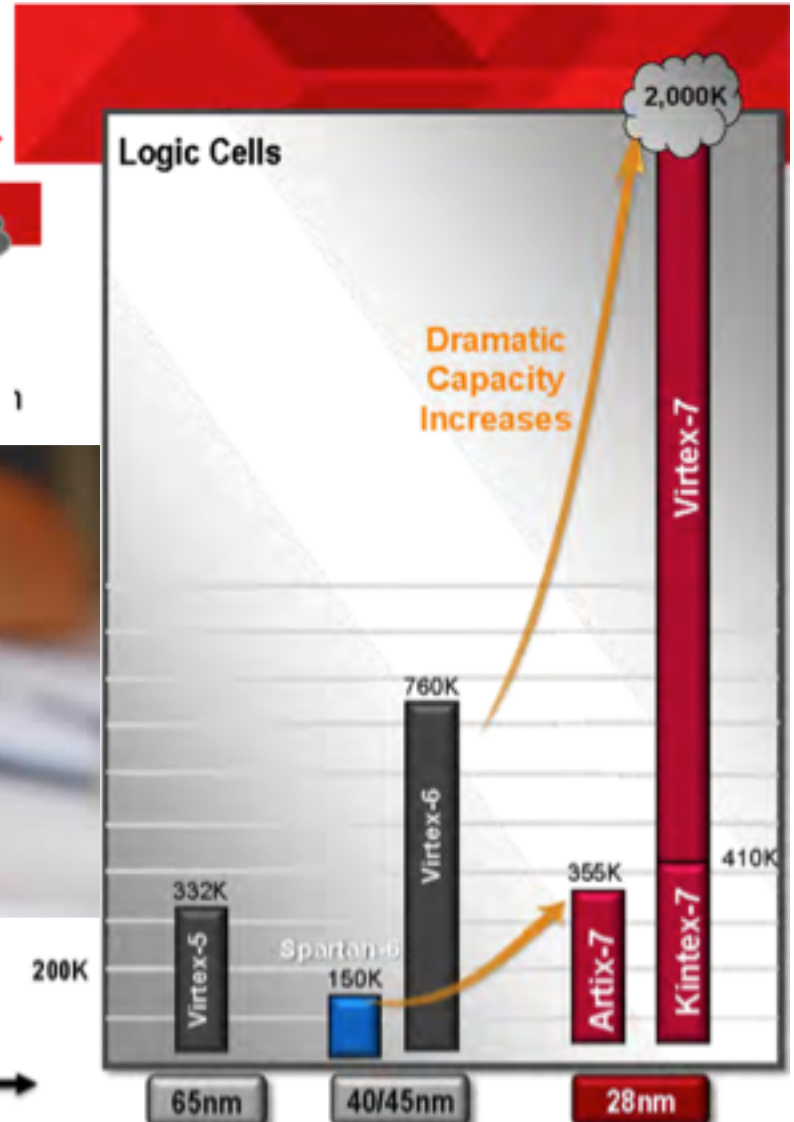


Logic Cells

- 28 nm: > 2X gains over 40 nm→

On-Chip High Speed Serial Links:

- Connect to new compact high density optical connectors (SNAP-12…)



synchronous vs asynchronous trigger processing

- some calculations are harder, others easier
  - example: there may be many or just a few tracks
- if you put data onto a computer: some events take longer to calculate than others
  - overall computing resources will be optimally used
  - so, is this fine?

synchronous vs asynchronous trigger processing

HEPHY Institute of High Energy Physics

- some calculations are harder, others easier
  - example: there may be many or just a few tracks
- if you put data onto a computer: some events take longer to calculate than others
  - overall computing resources will be optimally used
  - so, is this fine? - NO!
- danger! what if an event takes too long to process and is outside latency?
  - "watchdog" events: the watchdog will bark if you take too long!
  - just take all such events? - But there may be far too many of them!
  - just drop them? - But these may be the most interesting events. You might be killing all the "New Physics" events!
  - just take the percentage of them that you can afford? - Compromise, but may be a nightmare to analyze!

the beauty of synchronous trigger processing

- guaranteed latency – even the most complicated calculations fit into the available processing time
  - you are just “wasting resources” in case of “simple” events
  - like an assembly line: if a worker is fast, he will be idle part of the time and you lose salary money; if he is too slow, the whole production process will crash!
- enormous resources of present-day integrated circuits (ASICs and FPGAs) make this possible
- take care to choose correct programming style!
  - no loops
  - no conditional jumps
  - make everything parallel as much as possible

latency

I'm late! I'm late!

- latency is an important constraint on trigger architecture
- pipeline memory is expensive
  - in terms of money, space, energy consumption
- → need fast algorithms
- no iterative loops
- small propagation times → put trigger electronics close to detector
  - but not on detector (radiation protection!)

CMS Global Trigger in μTCA crate

BACKUP

funnel structure of trigger logic

- trigger schemes look a bit like a funnel:
- a lot of input information is used and compressed to yield eventually just one bit:
- YES or NO
  - take the event, or leave it?

luminosity

- (instant) luminosity is **rate per cross section**
- usual units: $cm^{-2}\ s^{-1}$
  - e.g., $10^{30}\ cm^{-2}\ s^{-1}$ corresponds, for a reaction cross section of $10^{-30}\ cm^{-2}$ (= 1 μbarn), to a rate of 1 event per second
- for a collider, the luminosity can be calculated as follows:

$$L = fn\frac{N_1 N_2}{A}$$

where

f is the revolution frequency

n is the number of bunches in one beam in the storage ring.

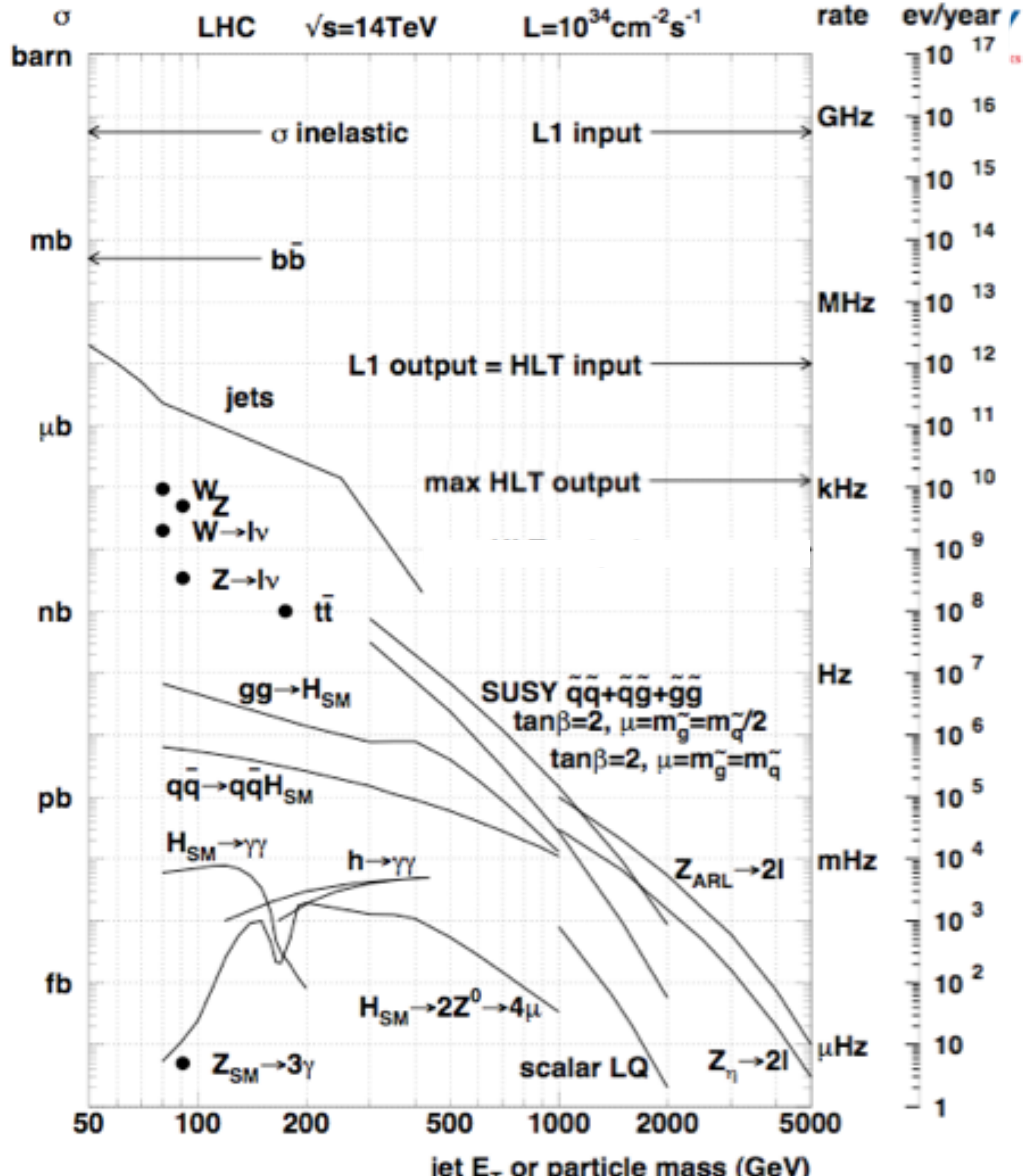
N_i is the number of particles in each bunch

A is the cross section of the beam.

integrated luminosity

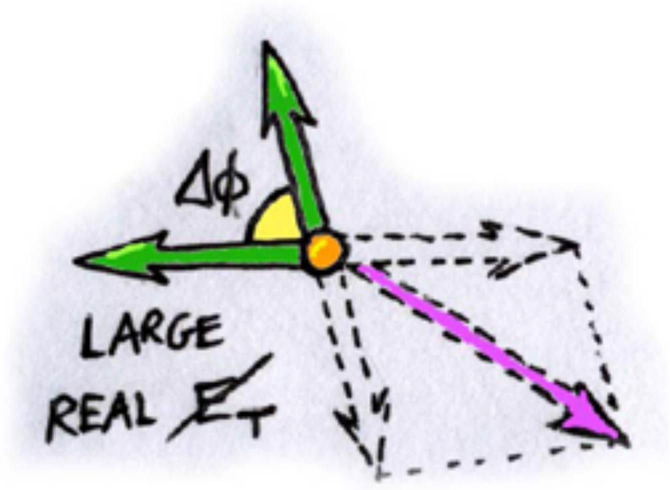
- number of events collected divided by the cross section
- usual units: fb^{-1} ("inverse femtobarn"),
  ab^{-1} ("inverse attobarn")
- an integrated luminosity of $1\ fb^{-1}$ means that for a process with a cross section of 1 fb, 1 event (on average) should have been collected
  - or 1000 events for a cross section of 1 nb, etc.
  - so, 1 inverse attobarn = 1000 inverse femtobarns :
    - $1\ ab^{-1} = 1000\ fb^{-1}$
- physicists are now looking for very rare events, so it is vital to reach not only high energies (so that heavy particles can be produced) but also high luminosities
  - handling the resulting data rates is a challenge also for the detectors, trigger systems, and readout electronics

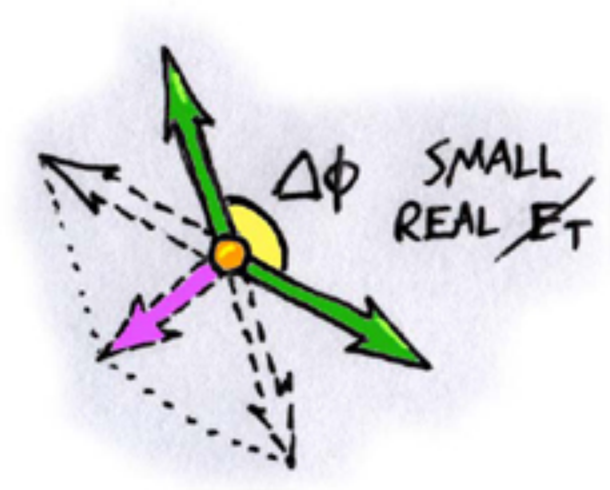
event rates as function of transverse momentum of jets, or of particle mass



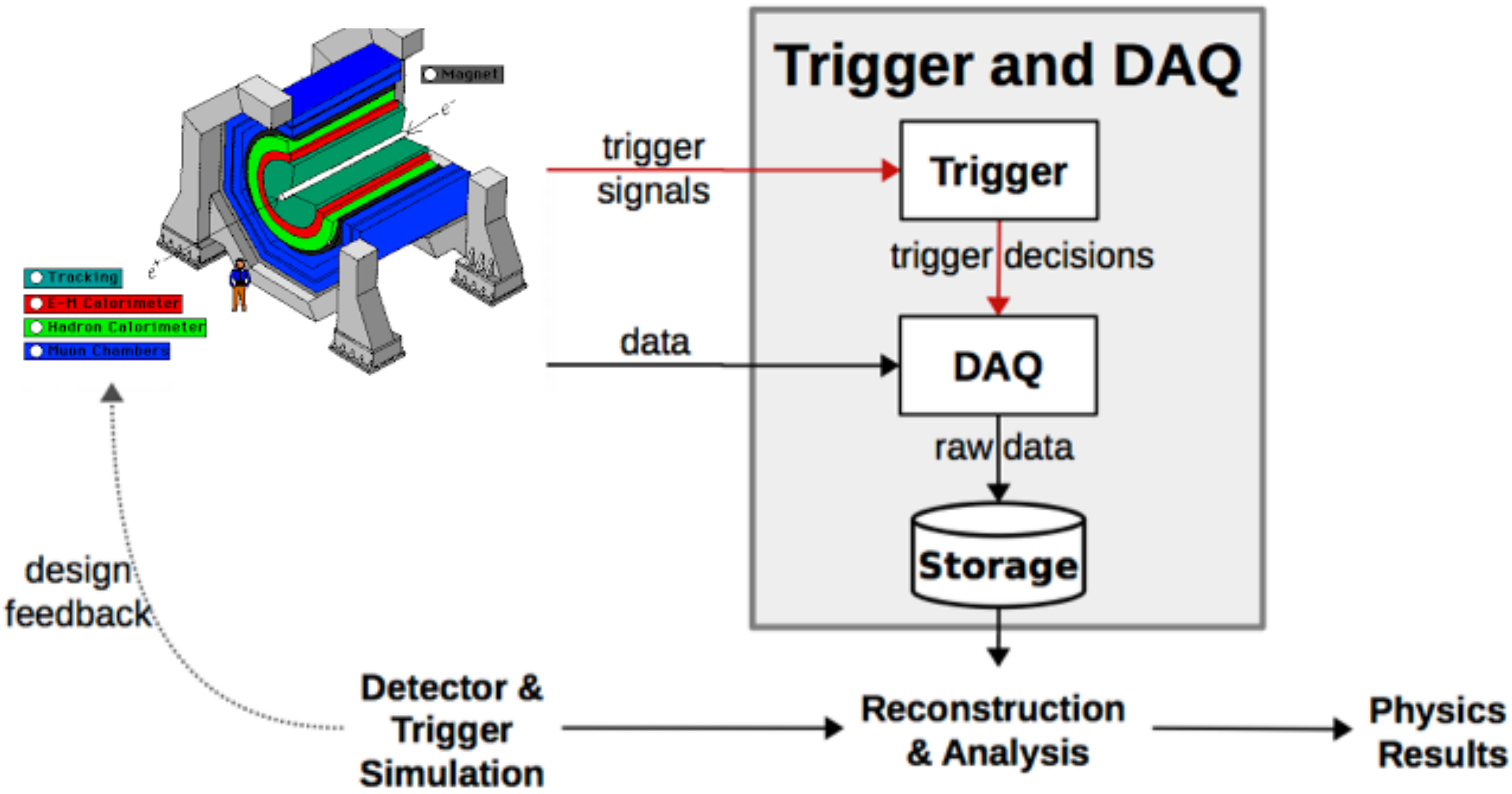
MET: missing transverse energy

- MET = “missing E_T”
- more precisely: “missing transverse momentum (p_T)”
- but at LHC energies momentum and energy is almost the same





trigger and DAQ at a collider



How does the trigger actually select events ?

- the first trigger stage has to process a limited amount of data within a very short time
  - relatively simple algorithms
  - special electronic components
    - ASICs (Application Specific Integrated Circuits)
    - FPGAs (Field Programmable Gate Arrays)
  - something in between "hardware" and "software": "firmware"
    - written in programming language ("VHDL") and compiled
    - fast (uses always same number of clock cycles)
    - can be modified at any time when using FPGAs
- the second stage ("High-Level Trigger") has to use complex algorithms
  - not time-critical any more (all detector data have already been retrieved)
  - uses a "computer farm" (large number of PCs)
  - programmed in high-level language (C++)

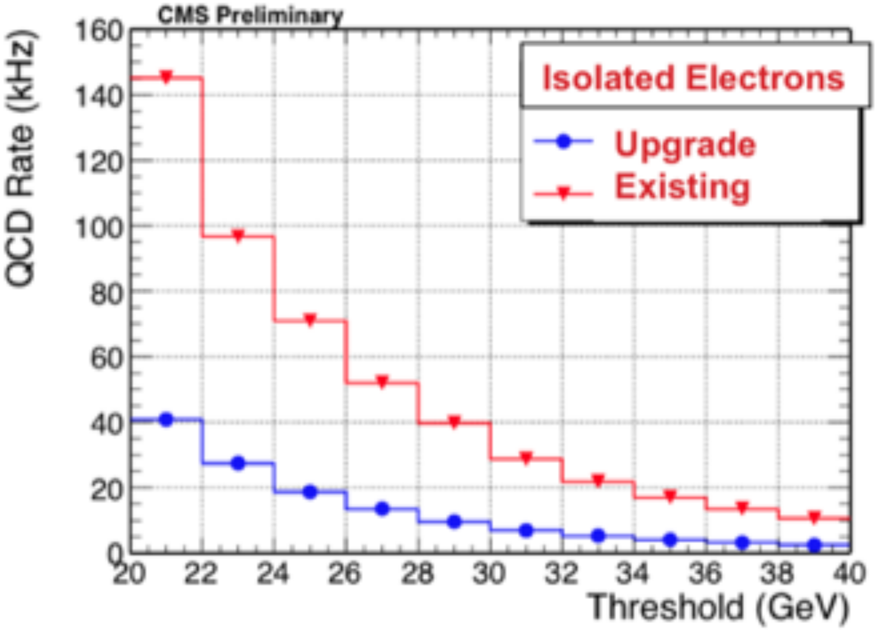
How does the trigger actually select events ?

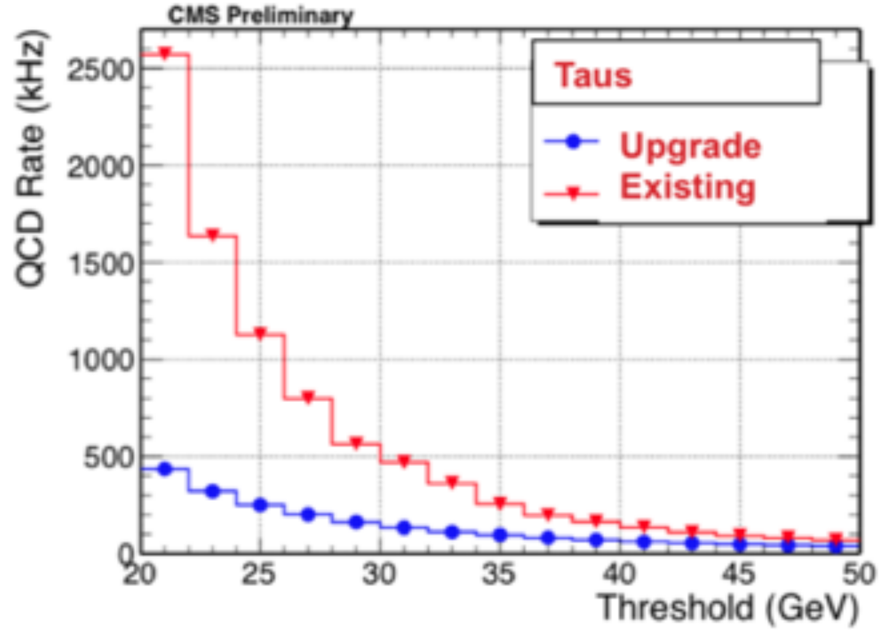
- the first trigger stage has to process a limited amount of data within a very short time
  - relatively simple algorithms
  - special electronic components
    - ASICs (Application Specific Integrated Circuits)
    - FPGAs (Field Programmable Gate Arrays)
  - something in between "hardware" and "software": "firmware"
    - written in programming language ("VHDL") and compiled
    - fast (uses always same number of clock cycles)
    - can be modified at any time when using FPGAs
- the second stage ("High-Level Trigger") has to use complex

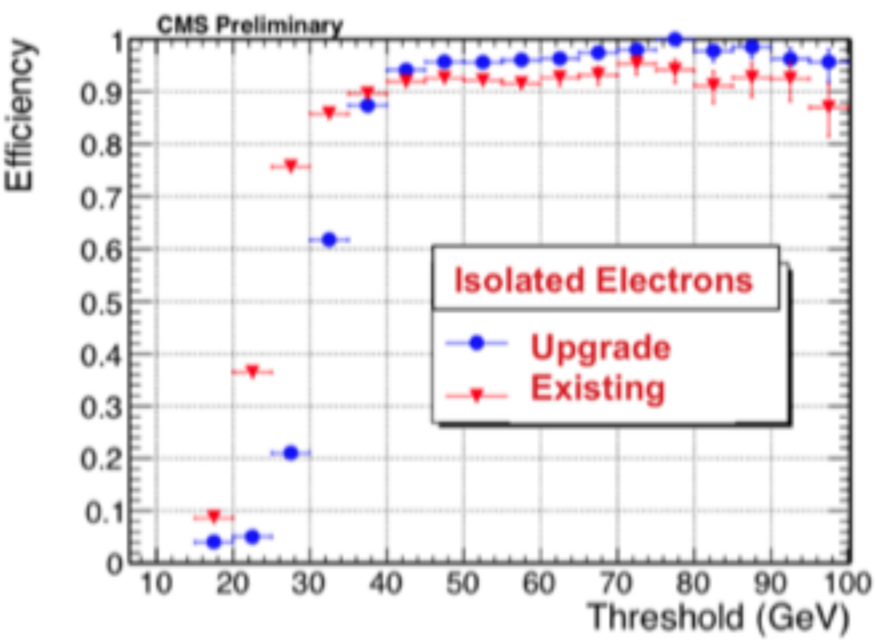
ved)

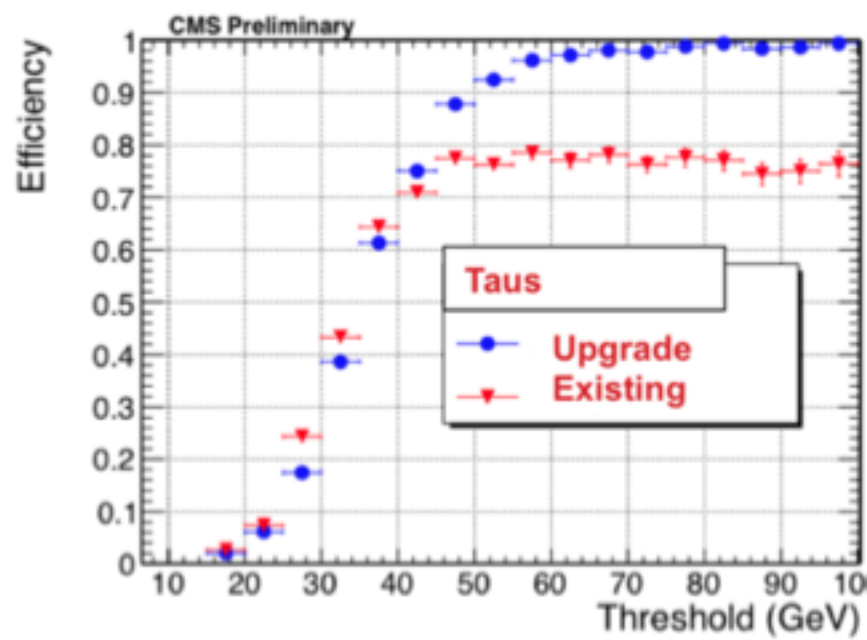
```
pre_algo_a(54) <= tau_2_s(2);
pre_algo_a(55) <= tau_2_s(1);
pre_algo_a(56) <= muon_1_s(10) AND ieg_1_s(2);
pre_algo_a(57) <= muon_1_s(6) AND ieg_1_s(28);
pre_algo_a(58) <= muon_1_s(8) AND (ieg_1_s(25) OR eg_1_s(7));
pre_algo_a(59) <= muon_1_s(9) AND (jet_1_s(9) OR fwdjet_1_s(5) OR tau_1_s(26));
pre_algo_a(60) <= muon_1_s(4) AND (jet_1_s(8) OR fwdjet_1_s(4) OR tau_1_s(25));
pre_algo_a(61) <= muon_1_s(7) AND (jet_1_s(4) OR fwdjet_1_s(20) OR tau_1_s(16));
pre_algo_a(62) <= muon_1_s(3) AND (jet_1_s(20) OR fwdjet_1_s(15) OR tau_1_s(10));
pre_algo_a(63) <= muon_1_s(2) AND tau_1_s(9);
pre_algo_a(64) <= muon_1_s(1) AND tau_1_s(20);
pre_algo_a(65) <= ieg_1_s(26) AND (jet_1_s(7) OR fwdjet_1_s(3) OR tau_1_s(24));
pre_algo_a(66) <= ieg_1_s(24) AND (jet_1_s(19) OR fwdjet_1_s(14) OR tau_1_s(8));
pre_algo_a(67) <= ieg_1_s(10) AND (jet_1_s(5) OR fwdjet_1_s(1) OR tau_1_s(19));
pre_algo_a(68) <= ieg_1_s(9) AND (jet_1_s(3) OR fwdjet_1_s(19) OR tau_1_s(15));
pre_algo_a(69) <= ieg_1_s(8) AND tau_1_s(7);
```

Rates and efficiencies of current and upgraded calorimeter trigger

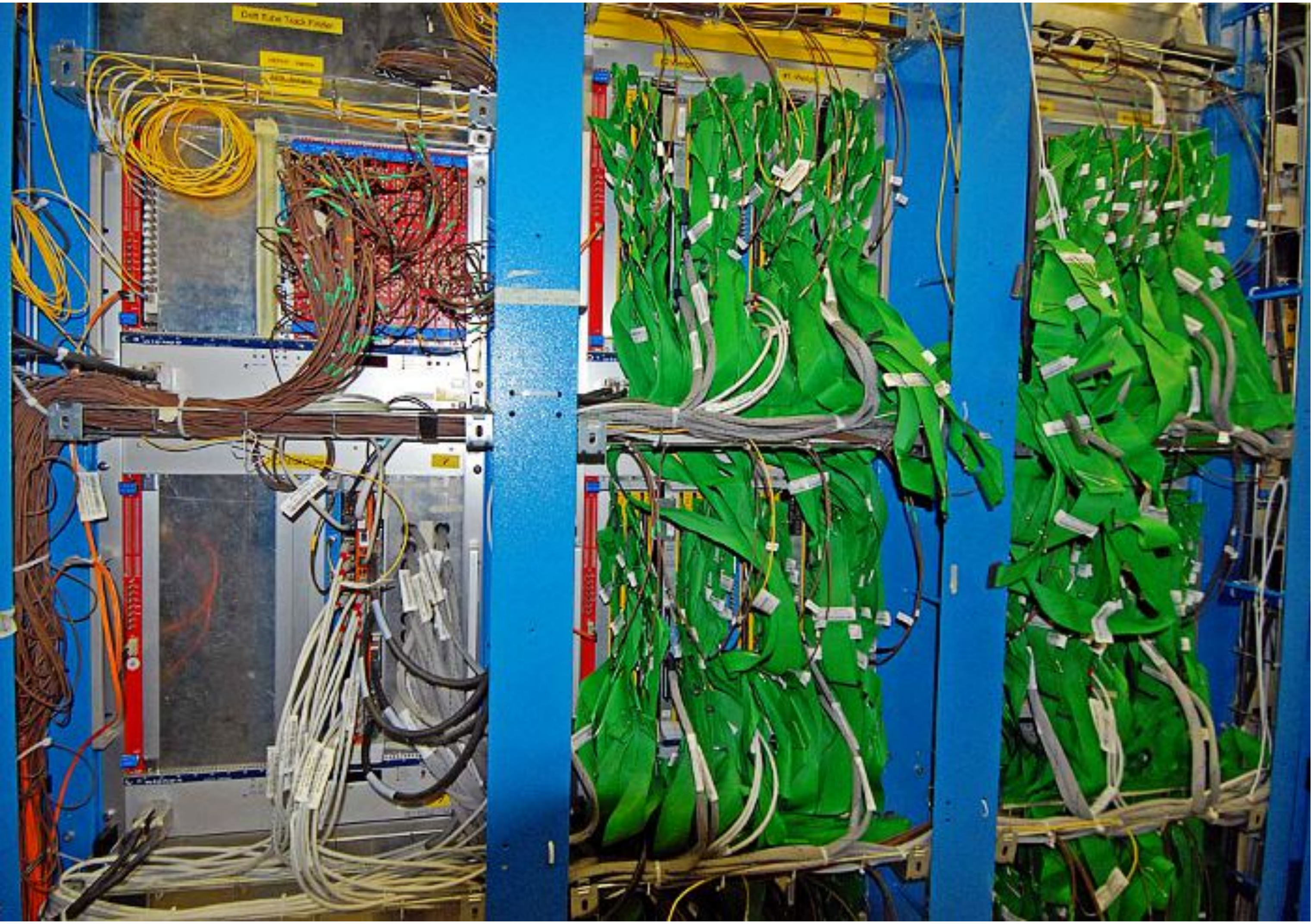








LHC Run 1 (<=2012):
many parallel galvanic connections



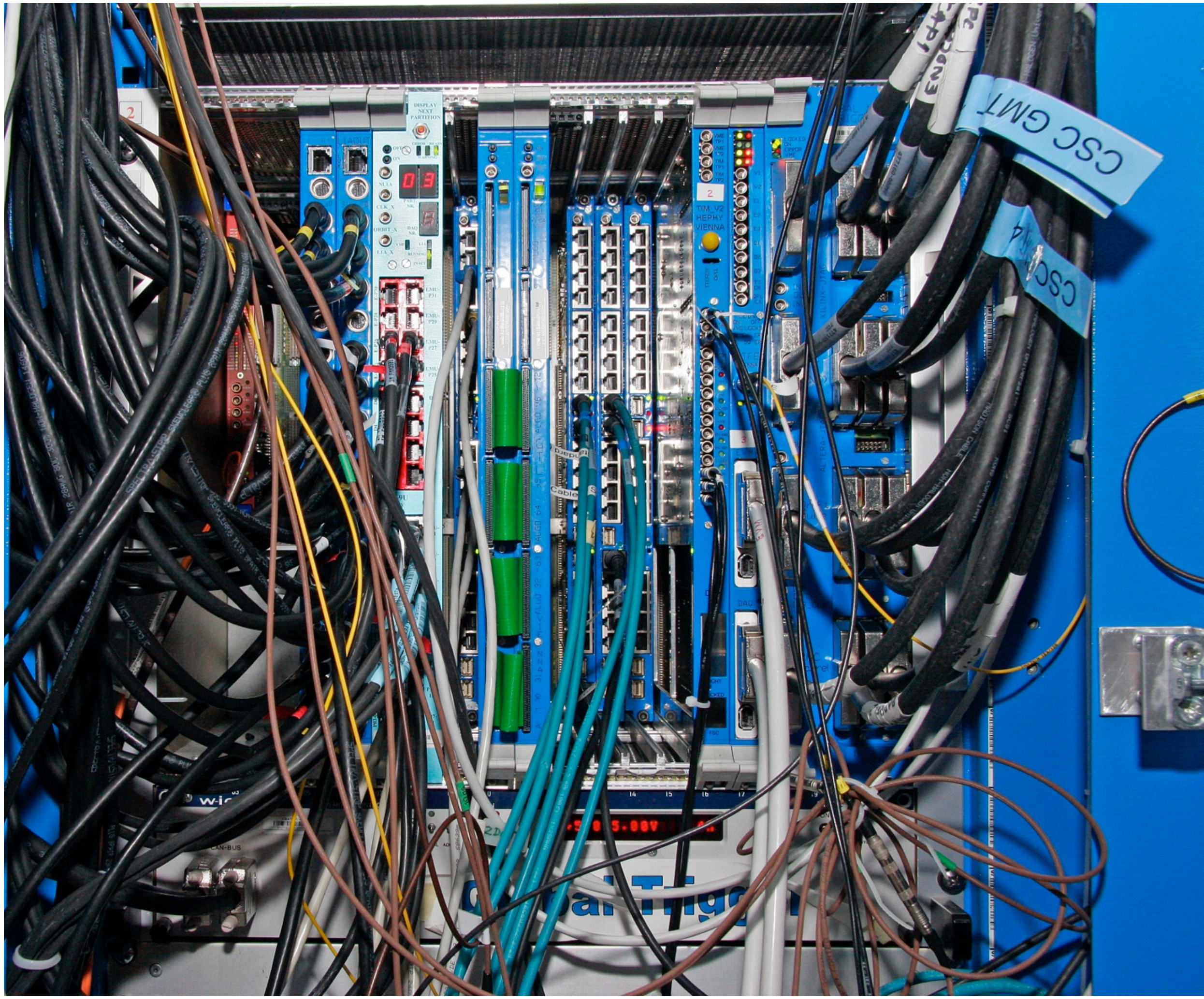
Example:

Drift Tube Track Finder (part of muon trigger) of the CMS experiment at CERN's LHC

the nightmare of interconnections!

LHC Run 1 (<=2012): many different custom-built electronics modules (VME)

HEPHY
Institute of High Energy Physics

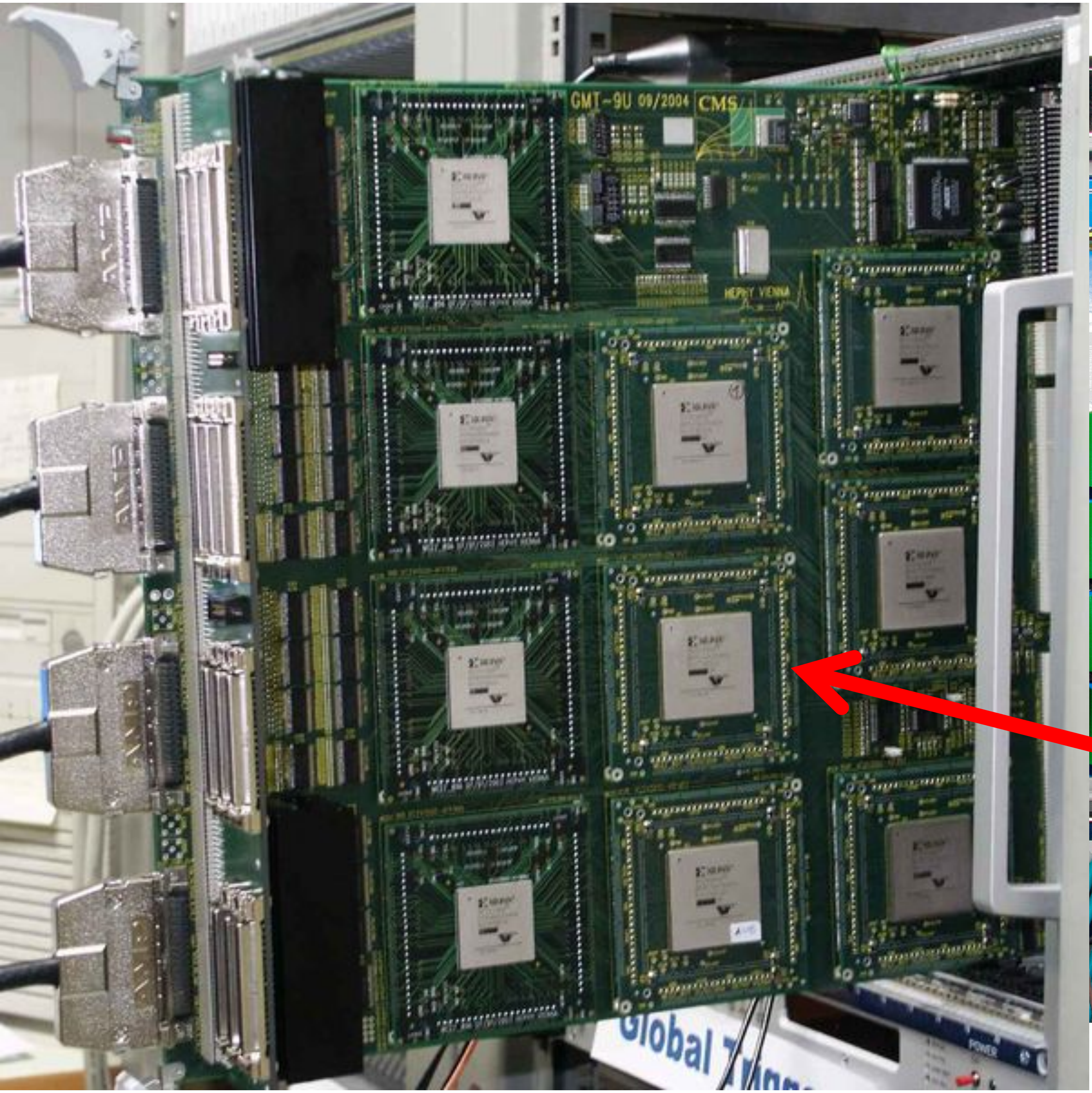


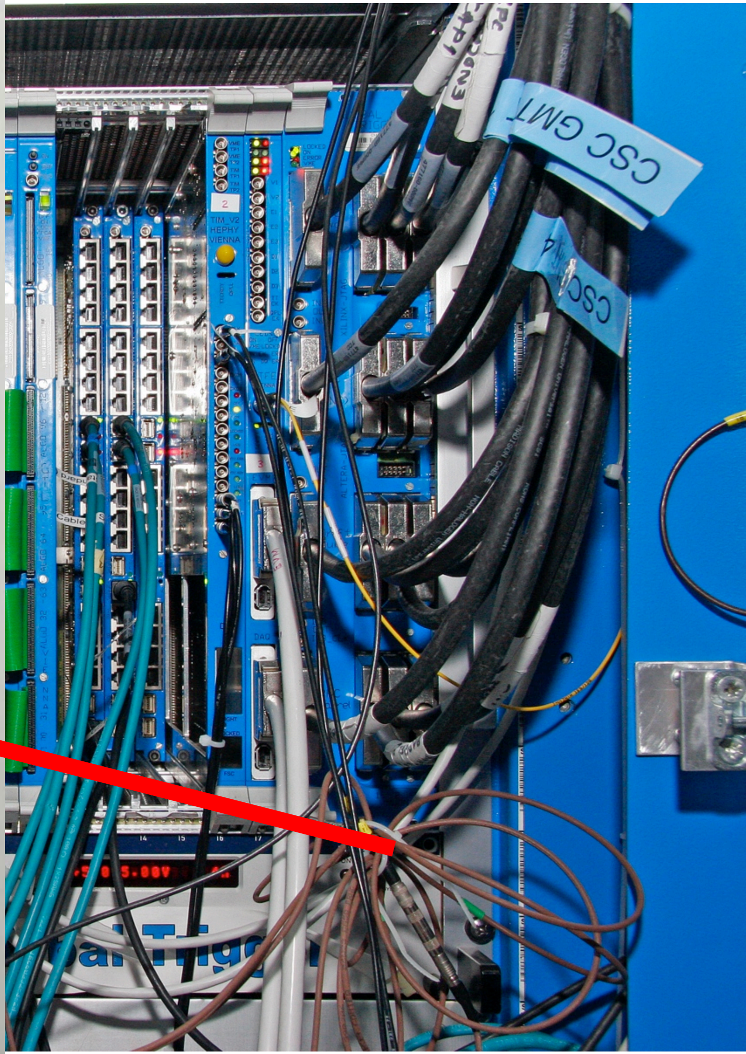
Example:

Global Trigger (left)
and
Global Muon Trigger (right)
of the CMS experiment at CERN's LHC

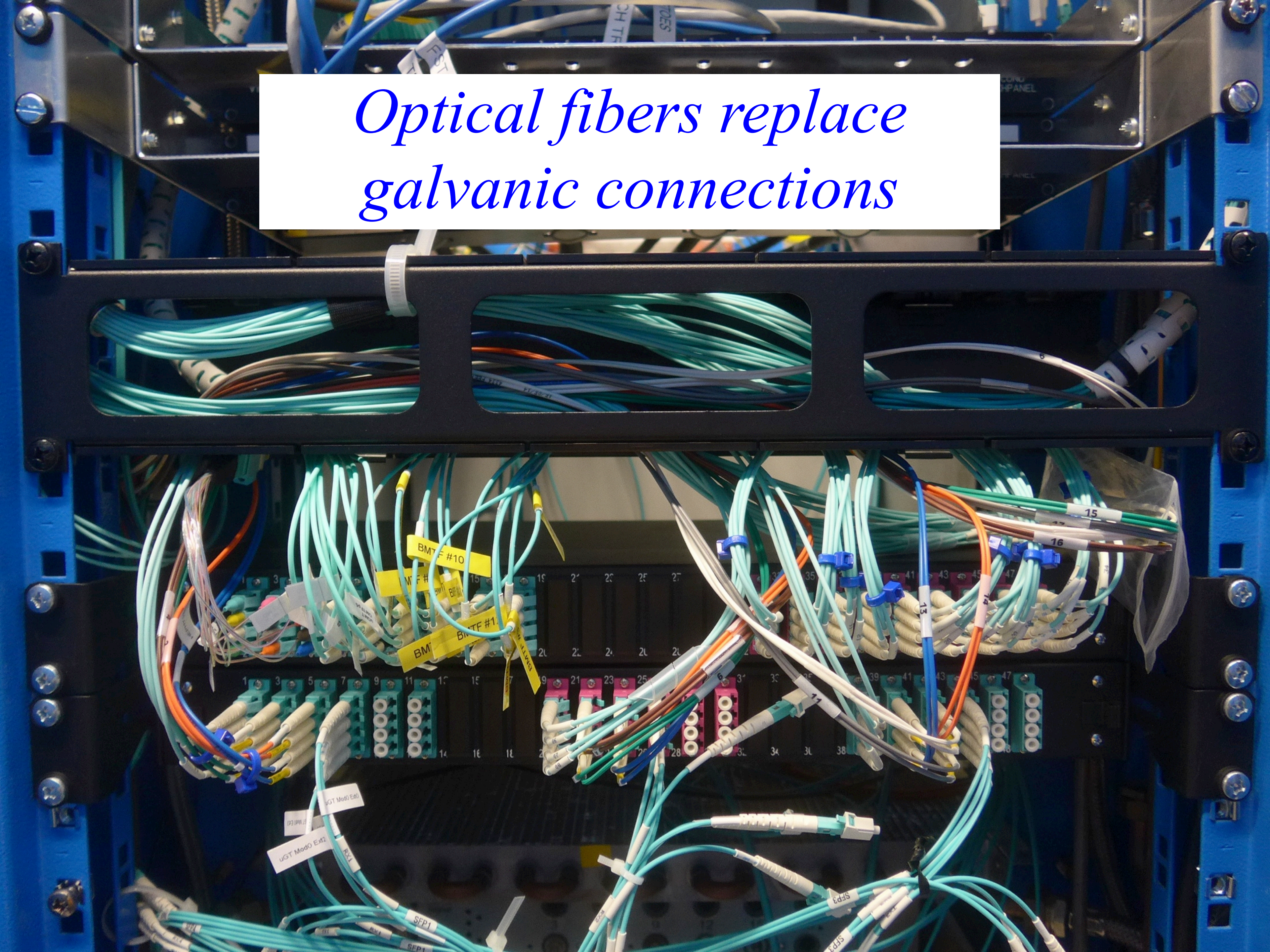
the nightmare of having enough spares!

Level-1 muon trigger





Optical fibers replace galvanic connections



optical fibers

☺ :

- "faster" in terms of more data volume per second over one line
- "cleaner": no electronic "cross-talk"

☹ :

- "slower" because serialization / deserialization needs time
- conversion into galvanic signal needed for processing
- no easy way to check signals on oscilloscope

ASICs and FPGAs

- ASIC: Application Specific Integrated Circuit
  - cheaper when using large quantities
- FPGA: Field Programmable Gate Array
  - cheaper when only few chips are needed
  - flexible: can be re-programmed in case of bugs or changes in requirements
  - the best of all worlds: fast as ASICs, flexible as computers (for a bit of extra money)
  - few vendors world-wide: Xilinx, Altera and just a few others

further reading

- W. R. Leo, "Techniques For Nuclear And Particle Physics Experiments", Springer, 1994
- CERN Summer Student Lectures
  - every year
  - 2023: https://indico.cern.ch/event/1254879/timetable/
- ISOTDAQ lectures
  - "International School of Trigger and Data AcQuisition", various years
  - 2023: https://indico.cern.ch/event/1182415/
- Technical Design Reports (TDR)
  - of big experiments such as ATLAS, CMS, BaBar, LHCb, D0
  - baselines, upgrades
  - different publication dates