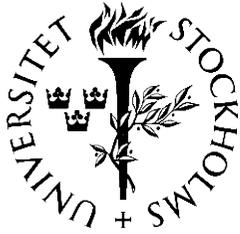




Intro to Triggering and Data Acquisition

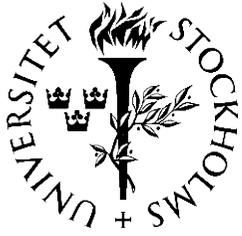
Samuel Silverstein, Stockholm University

- Trigger/DAQ basics
 - Collider TDAQ
 - Trigger algorithms
 - Dead time
-

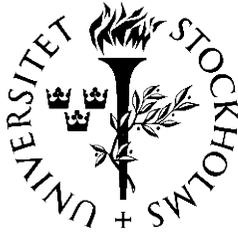


In this trigger/DAQ series:

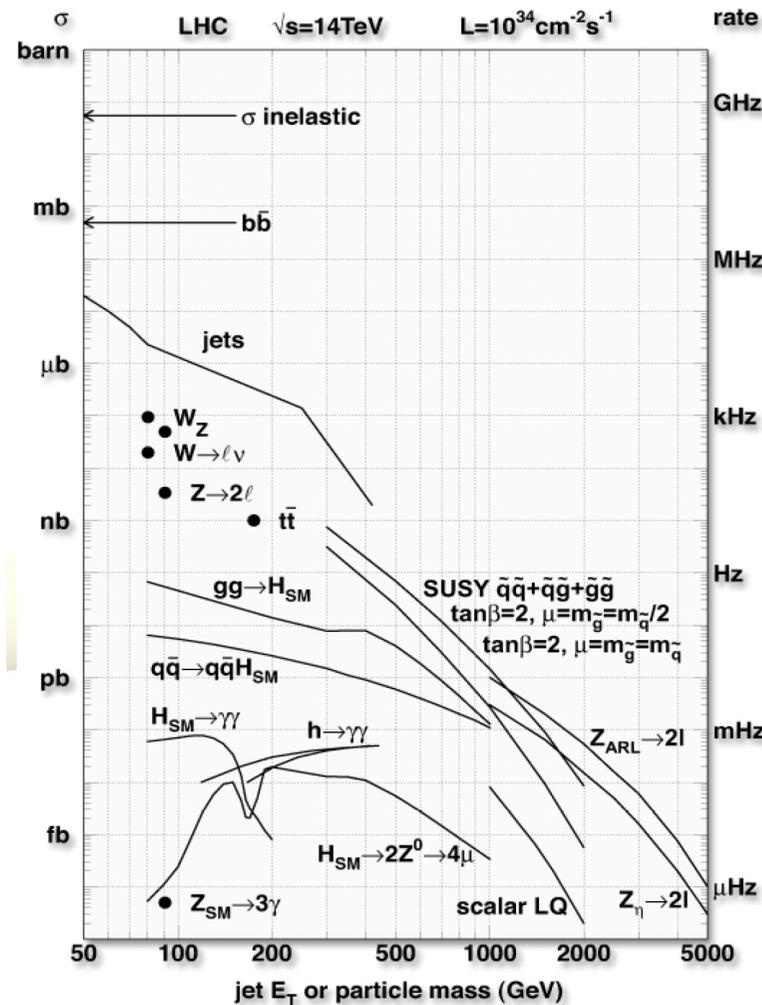
- Three lectures
 - Introduction to Trigger/DAQ
 - Algorithms and architectures
 - New directions: SLHC
- Exercises
 - Dead-time calculation
 - Field-programmable gate arrays (Lab)



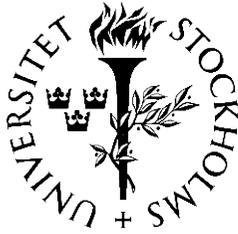
Introduction



Colliders produce a lot of data



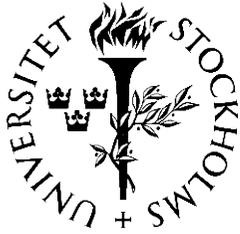
- Interesting physics at LHC: 1 event in $\sim 10^6$ - 10^{13}
- So we need a high rate:
 - 40 MHz bunch crossing rate
 - Many proton collisions per bunch crossing
- Typical ATLAS collision event is about 1.5 MB
 - Ca. 60 TB/s at 40 MHz!
- We can't look at all of it!



Why can't we look at all of it?

(In reverse order...)

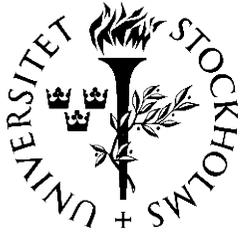
- We can't analyse every collision
 - without a million times as much processing
- We can't even store it all
 - 60 TB/s is about 3.6 petabytes/minute
- We can't even get it all off the detector!
 - High-bandwidth data links are expensive, take up space and consume power
 - Practical consequences:
 - Heat dissipation
 - "Holes" for cables, power, cooling, etc.
 - Dead material



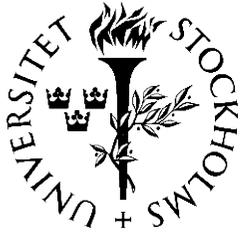
So, the challenge is:

- Throw away 99.999% of the data
- But don't throw away:
 - Interesting si
 - New signals predicted by
 - Unexpected new signals
- Doing both of these well is a challenge
 - And perfection is practically impossible

Data acquisition is a compromise between physics goals and the technically achievable

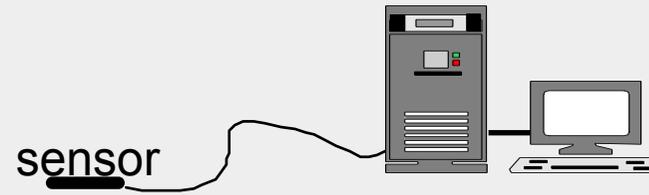


Data Acquisition Basics

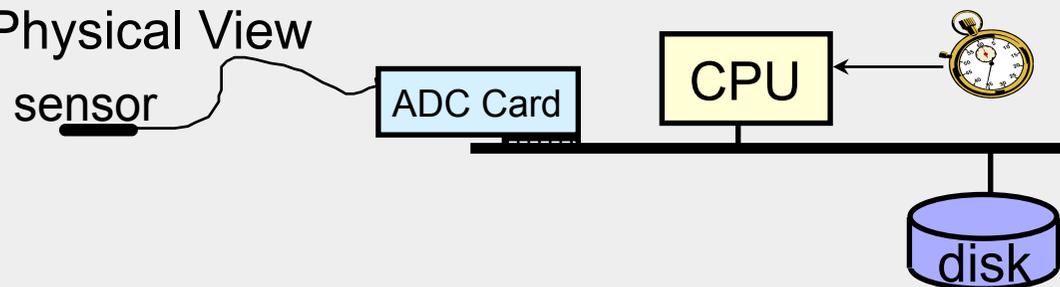


Example: trivial DAQ

External View

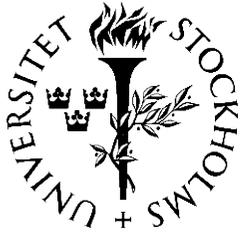


Physical View



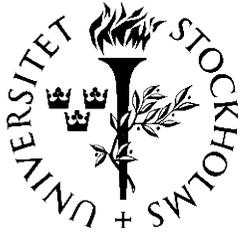
Logical View





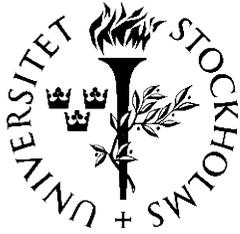
Trivial DAQ

- How it works:
 - Sensor produces analog output
 - ADC periodically converts the analog output to digital values
 - CPU reads digital values from ADC and writes them to disk (readout)
- Problem:
 - If physics rate is much lower than readout rate, you record a lot of empty (uninteresting) data
- Solution:
 - Implement a 'trigger' to initiate readout only if there is an interesting signal



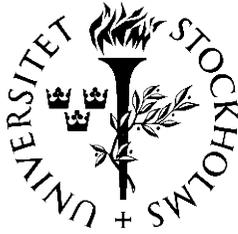
What is a trigger?

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. “

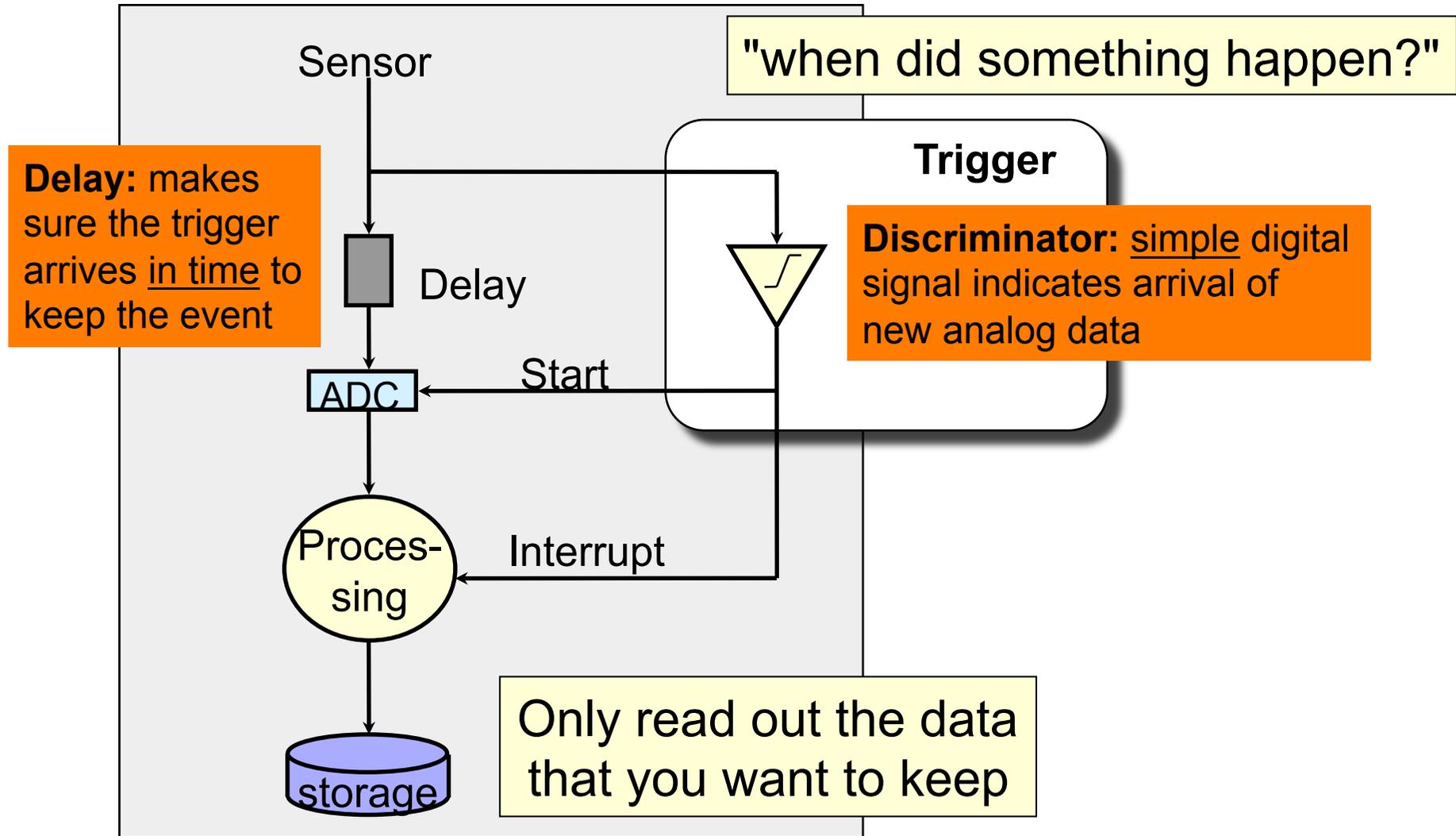


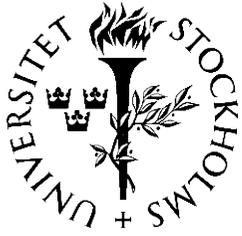
What is a trigger?

- Simple
 - Just need to decide whether to keep the data. Detailed analysis comes later
- Rapid
 - The data you want to keep may be lost if you wait too long to decide
- Selective
 - Need to achieve a sufficient reduction in readout rate, because...
 - “Only a small fraction can be recorded”

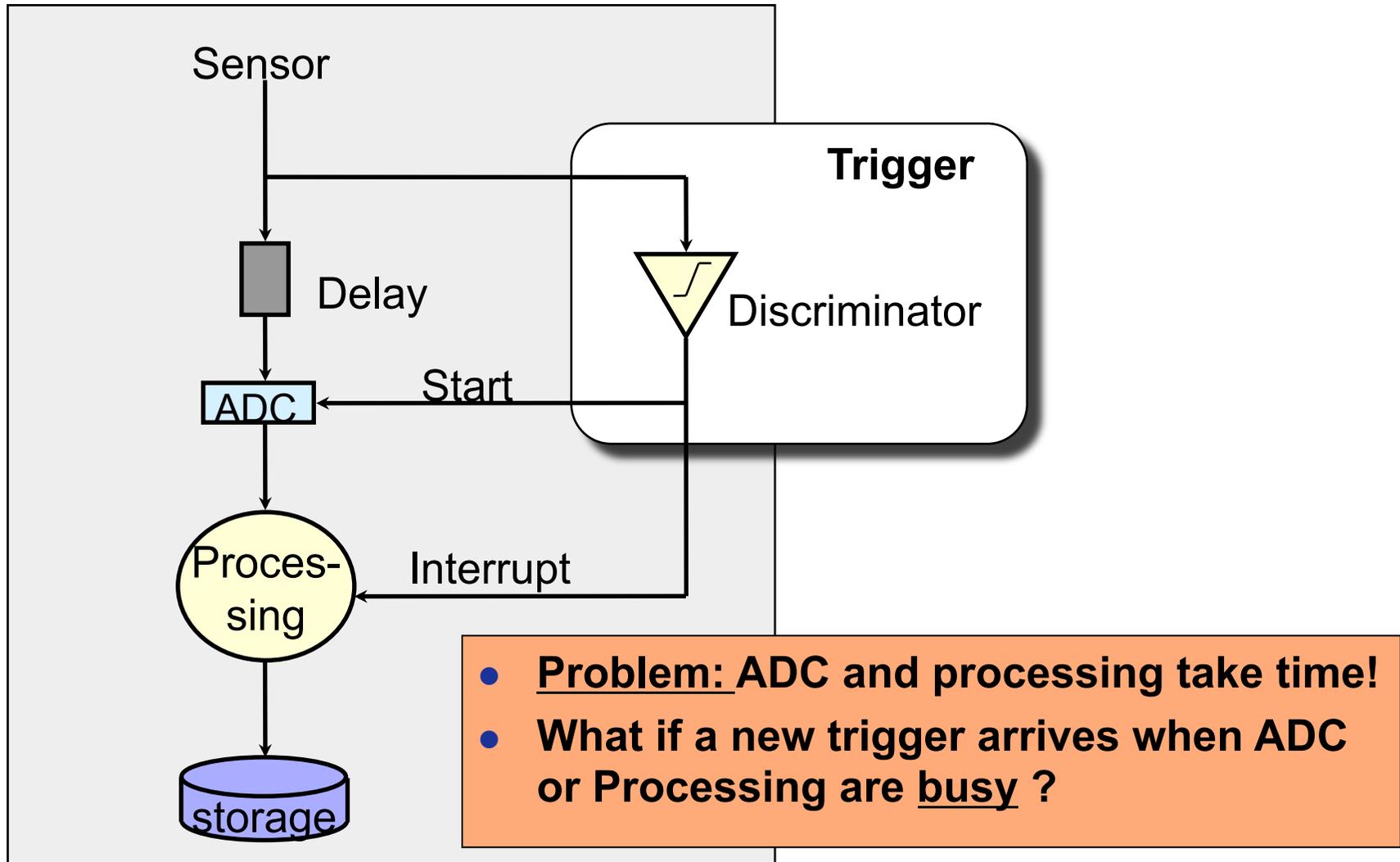


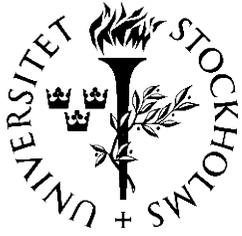
Trivial DAQ with trigger



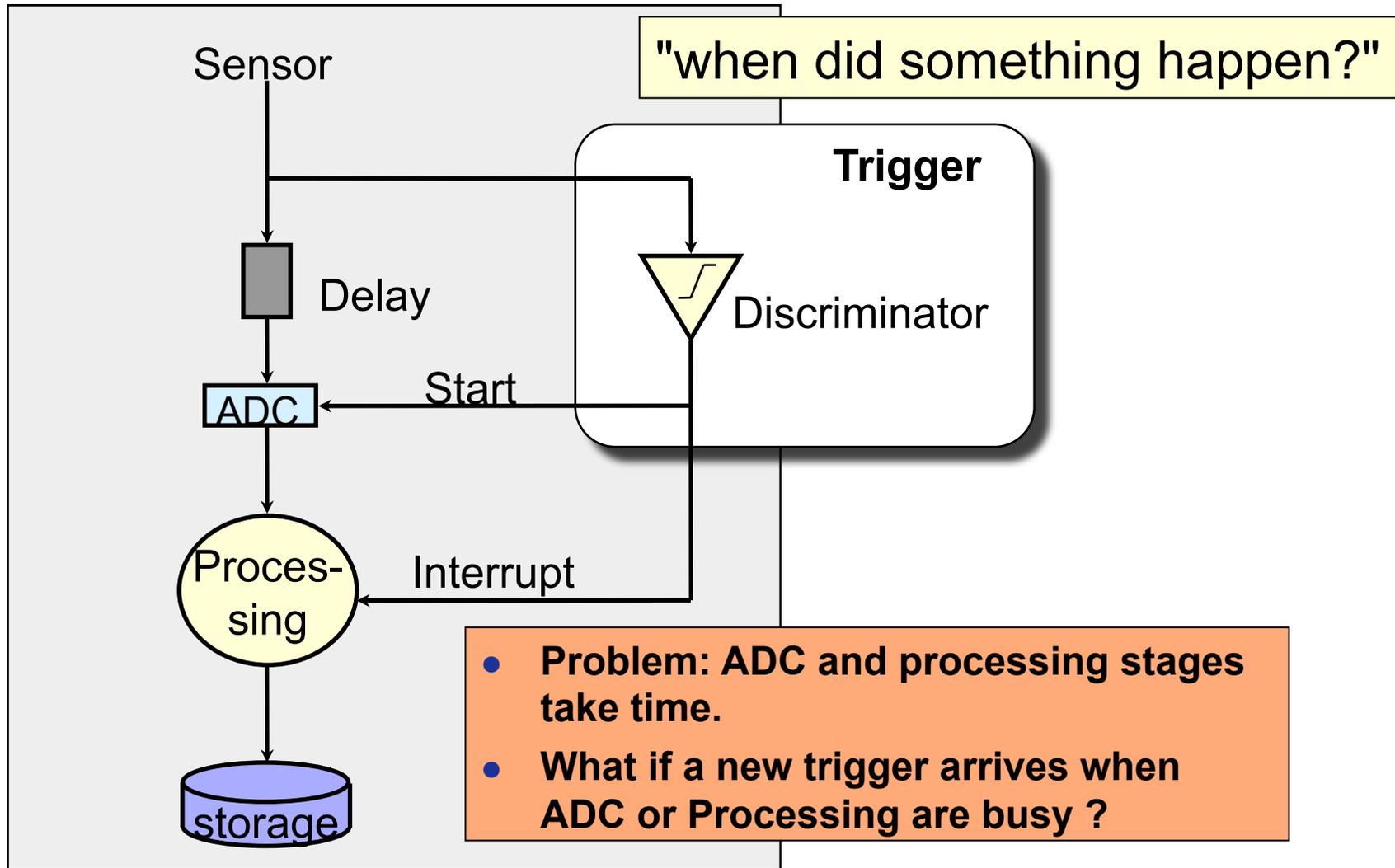


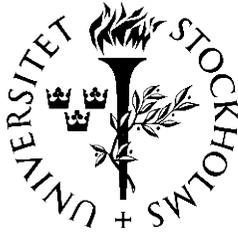
Trivial DAQ with trigger



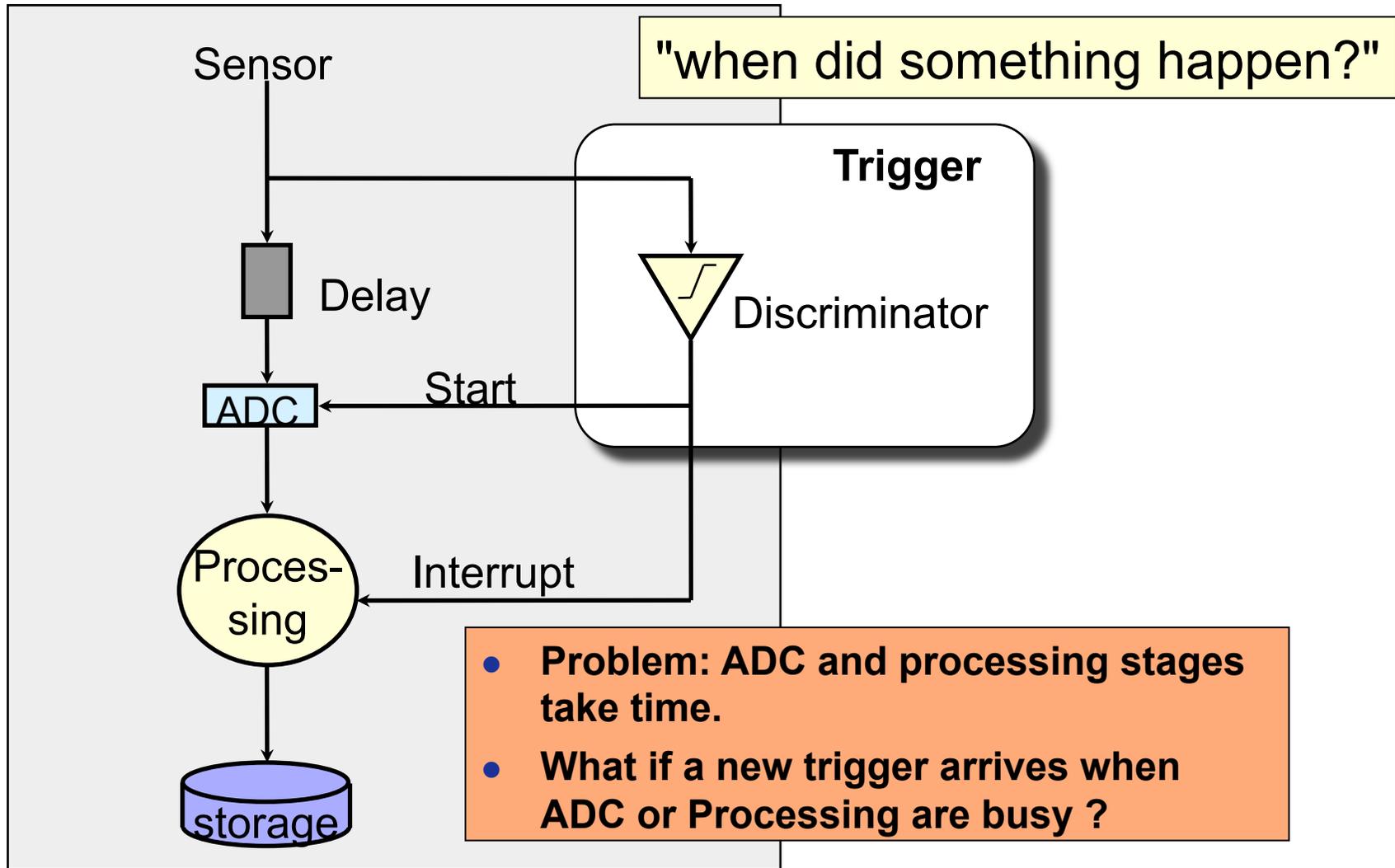


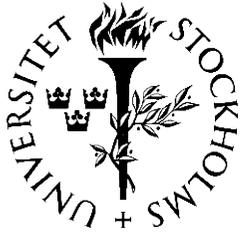
Trivial DAQ with trigger





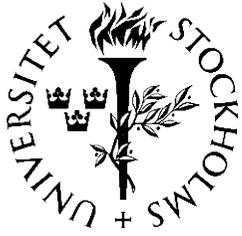
Trivial DAQ with trigger



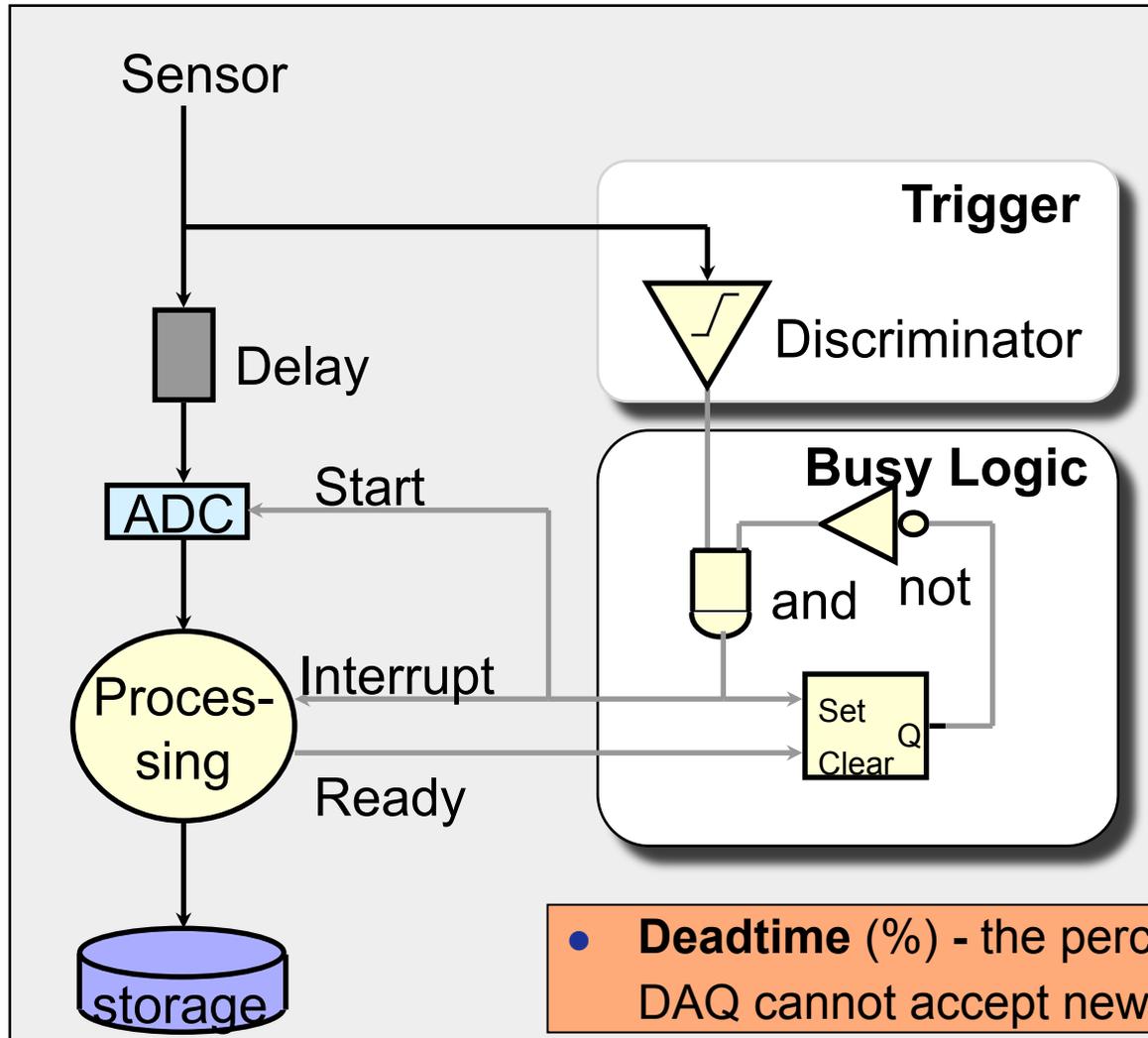


Flow control

- New triggers can interfere with the ADC or readout if a previous event is still being processed
- Solution: flow control
 - ADC/readout assert a 'Busy' signal to temporarily disable the trigger
 - Events occurring while the 'Busy' signal is asserted are not recorded.

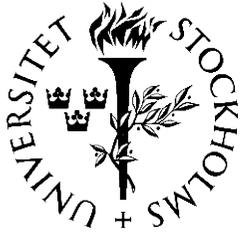


DAQ with trigger & busy logic



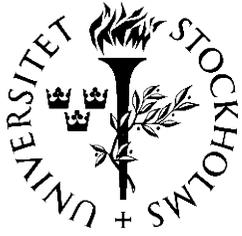
- **Busy time =**
ADC readout
+
processing time

- **Deadtime (%)** - the percentage of time that the DAQ cannot accept new events



Dead time

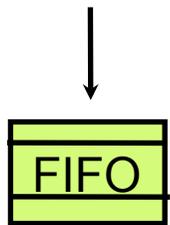
- Problem:
 - Time between trigger is essentially random (Poisson process)
 - But it takes about the same time to read out each event (Gaussian)
- As the trigger rate approaches the readout rate, dead time increases (and with it, data loss)
- Solution: De-randomising buffer



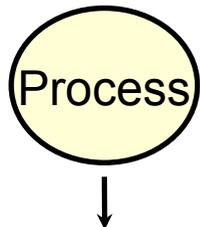
Buffers reduce dead time

- Dead time depends on trigger rate, readout time, processing time per event, and buffer (FIFO) depth.
- Buffers "average out" arrival rate of new events

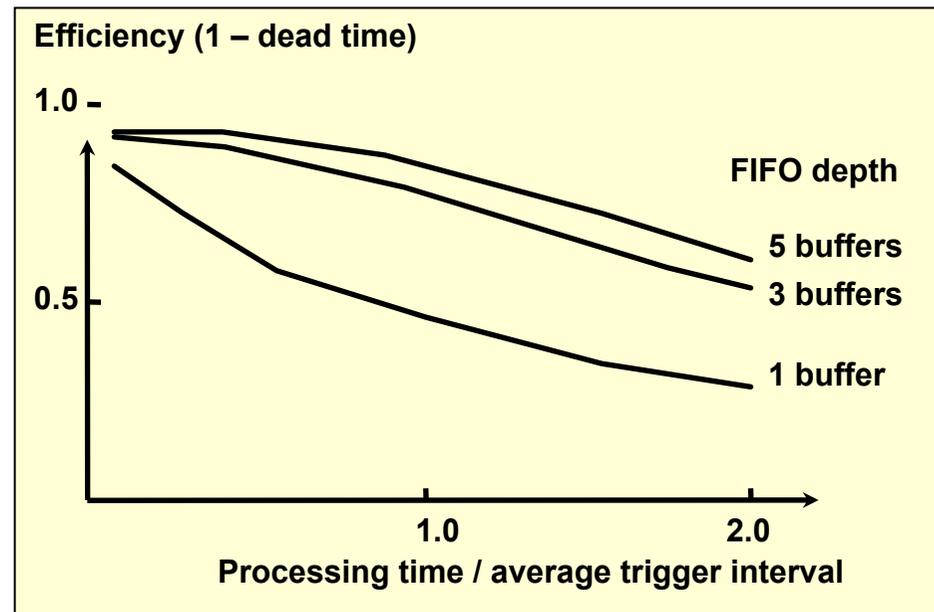
Arrival time follows exponential distribution (Poisson process)

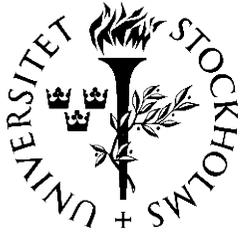


Depth 1, 3, 5 ...

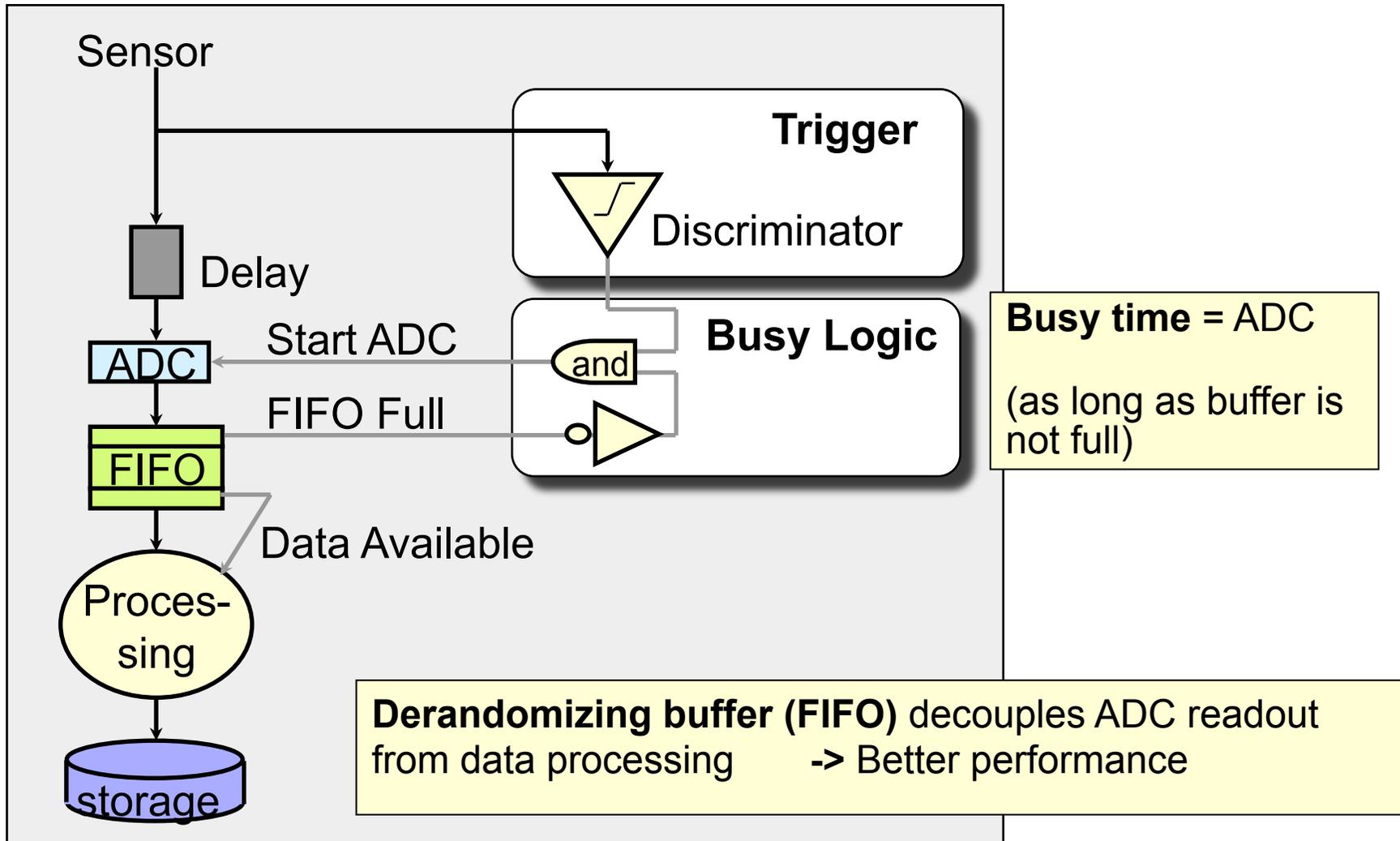


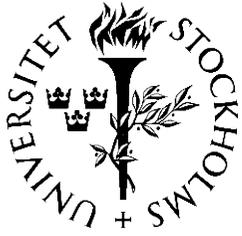
Processing time follows gaussian distribution



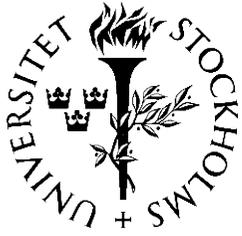


Buffered DAQ



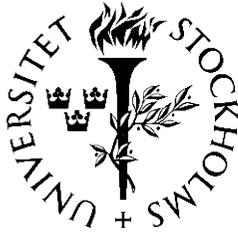


Collider Trigger/DAQ

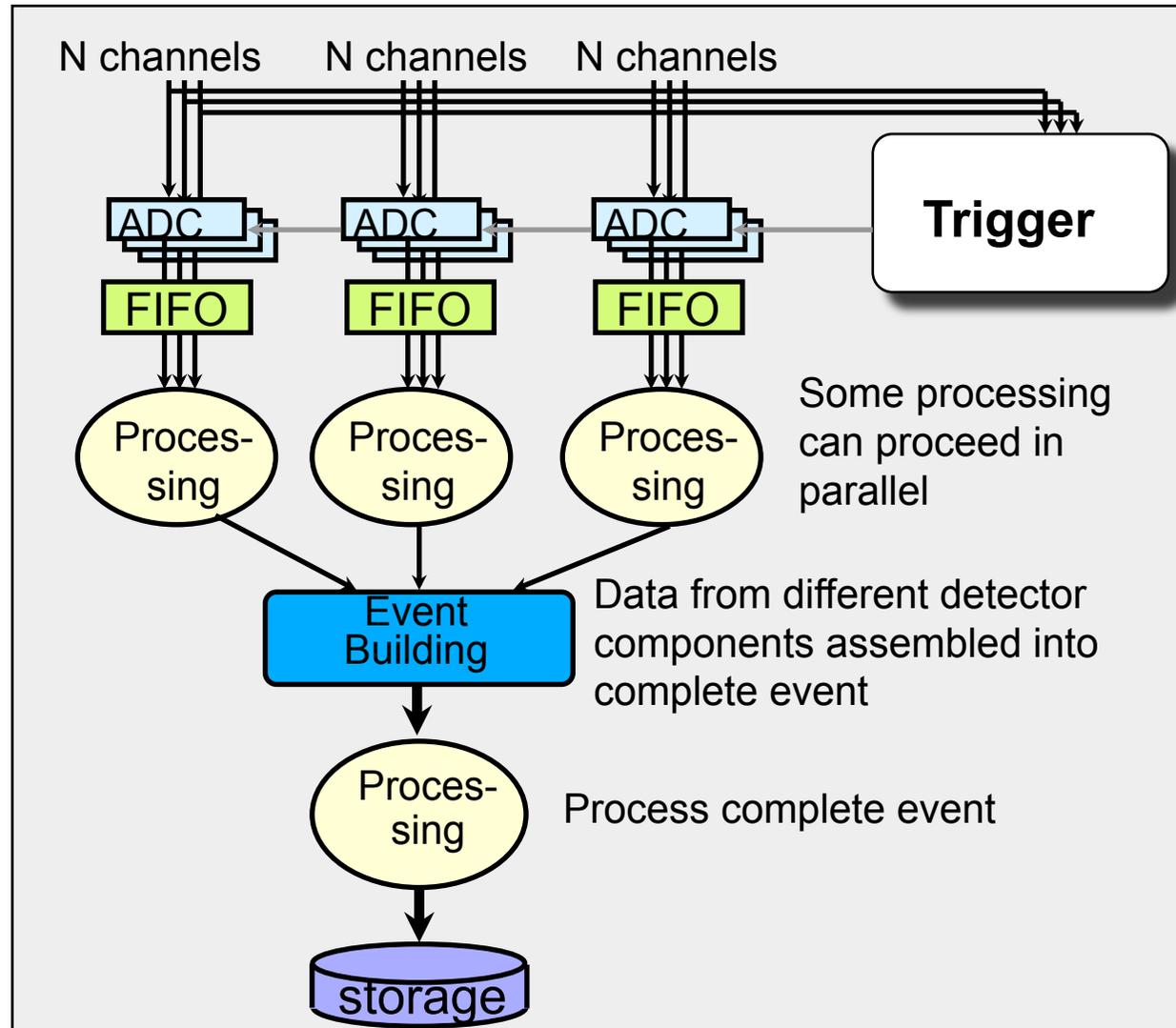


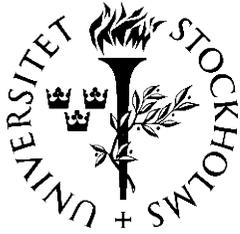
Triggering multiple systems

- Collider detectors are collections of detector systems working together
 - Tracking detectors
 - Calorimeters
 - Muon spectrometers
- Central trigger initiates readout for all detectors
- DAQ collects and records data from all detectors for each triggered event



Multi-system Trigger/DAQ



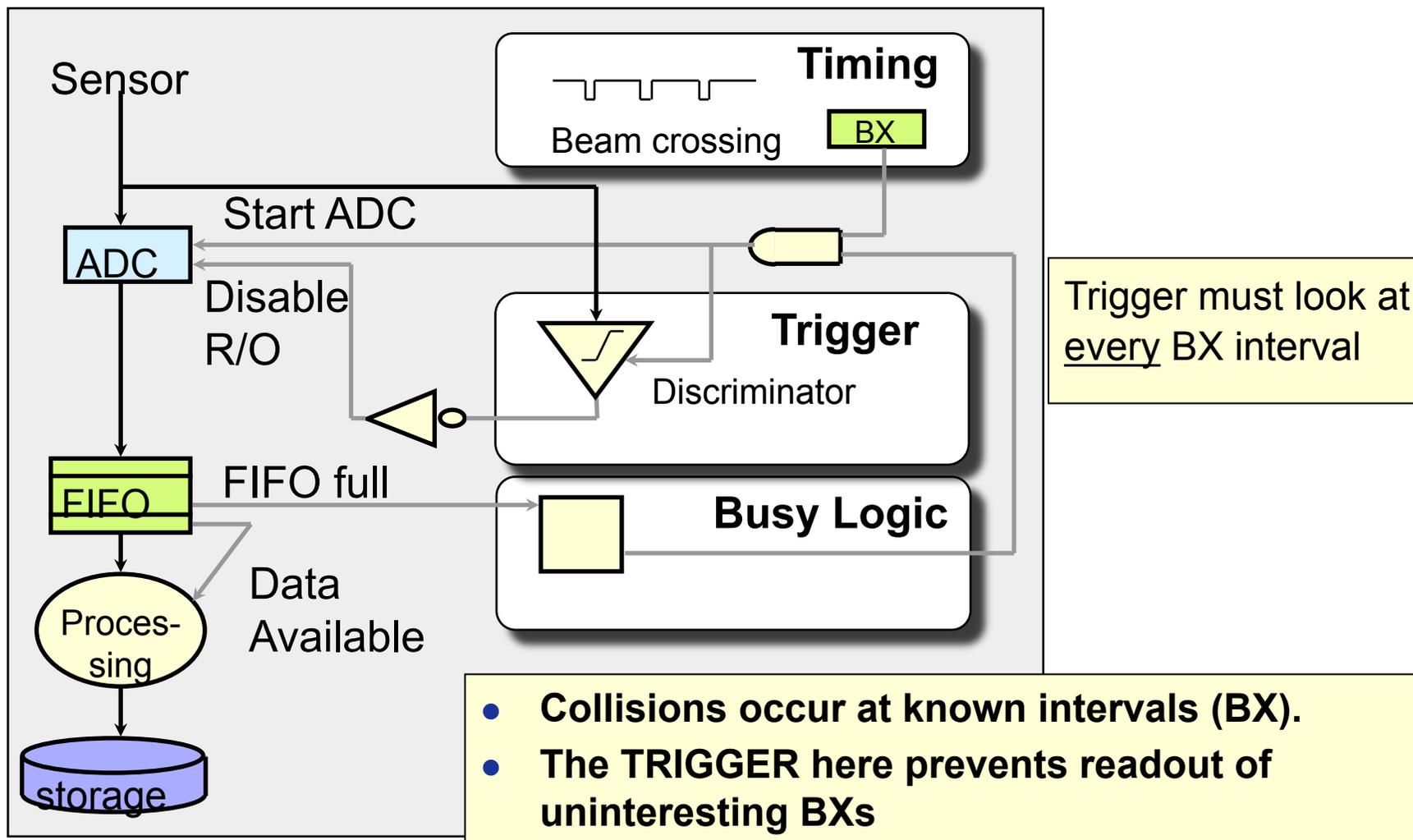


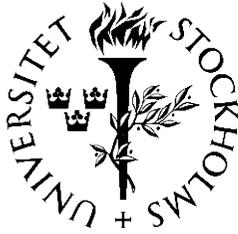
Triggering on bunch crossings

- Particle colliders produce collisions at a known bunch crossing (BX) interval
- Trigger system must analyze and produce a yes/no trigger decision for every bunch crossing
- DAQ readout occurs only for BXs selected by the trigger
 - Data from the other BXs are effectively “thrown away”.



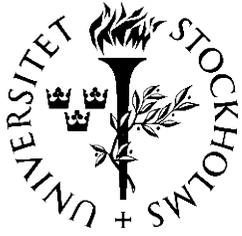
Trivial collider trigger





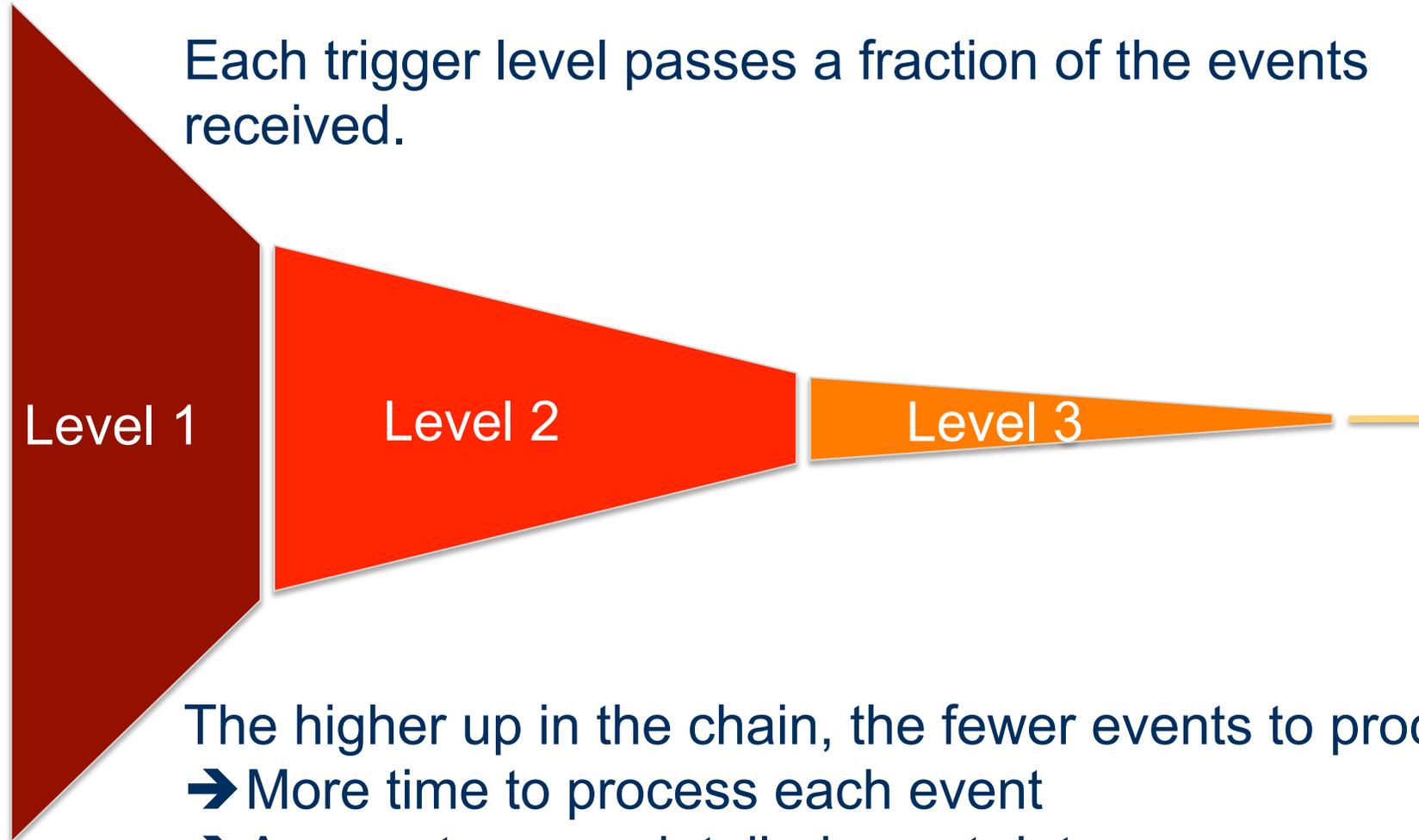
Multi-level triggering

- Modern colliders like LHC need to reject ~99.999% of events
 - But keep most of the valuable physics!
- Cannot achieve this performance with a single trigger.
 - An algorithm that selective needs full detector information, and algorithms too complex to run 40 million BXs/second
- Solution: Multi-level trigger



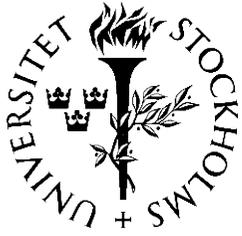
Multi-level trigger

Each trigger level passes a fraction of the events received.



The higher up in the chain, the fewer events to process

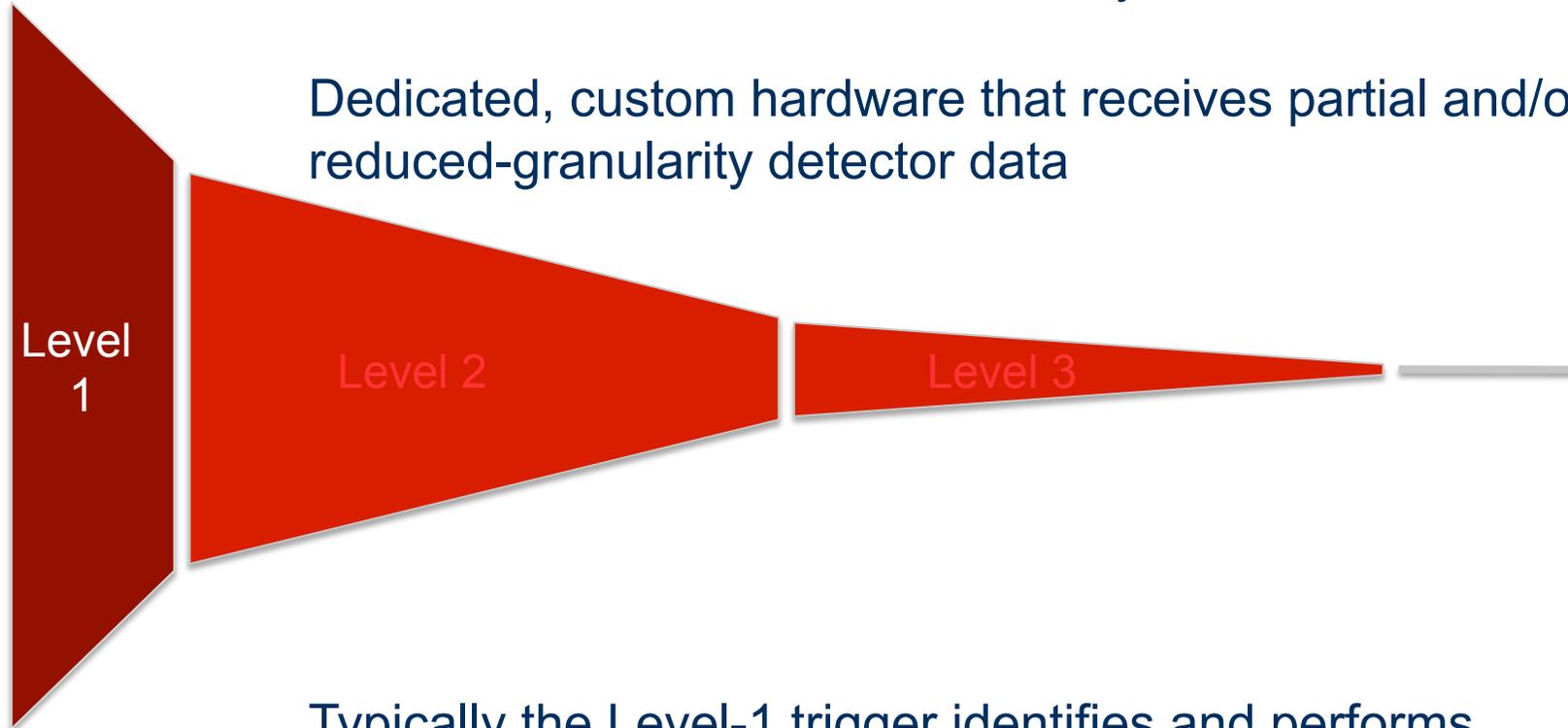
- ➔ More time to process each event
- ➔ Access to more detailed event data



Level 1 trigger

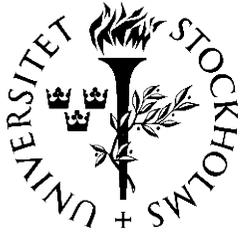
Quick first look while data still locally stored on detector

Dedicated, custom hardware that receives partial and/or reduced-granularity detector data



Typically the Level-1 trigger identifies and performs simple selection algorithms on e.g.:

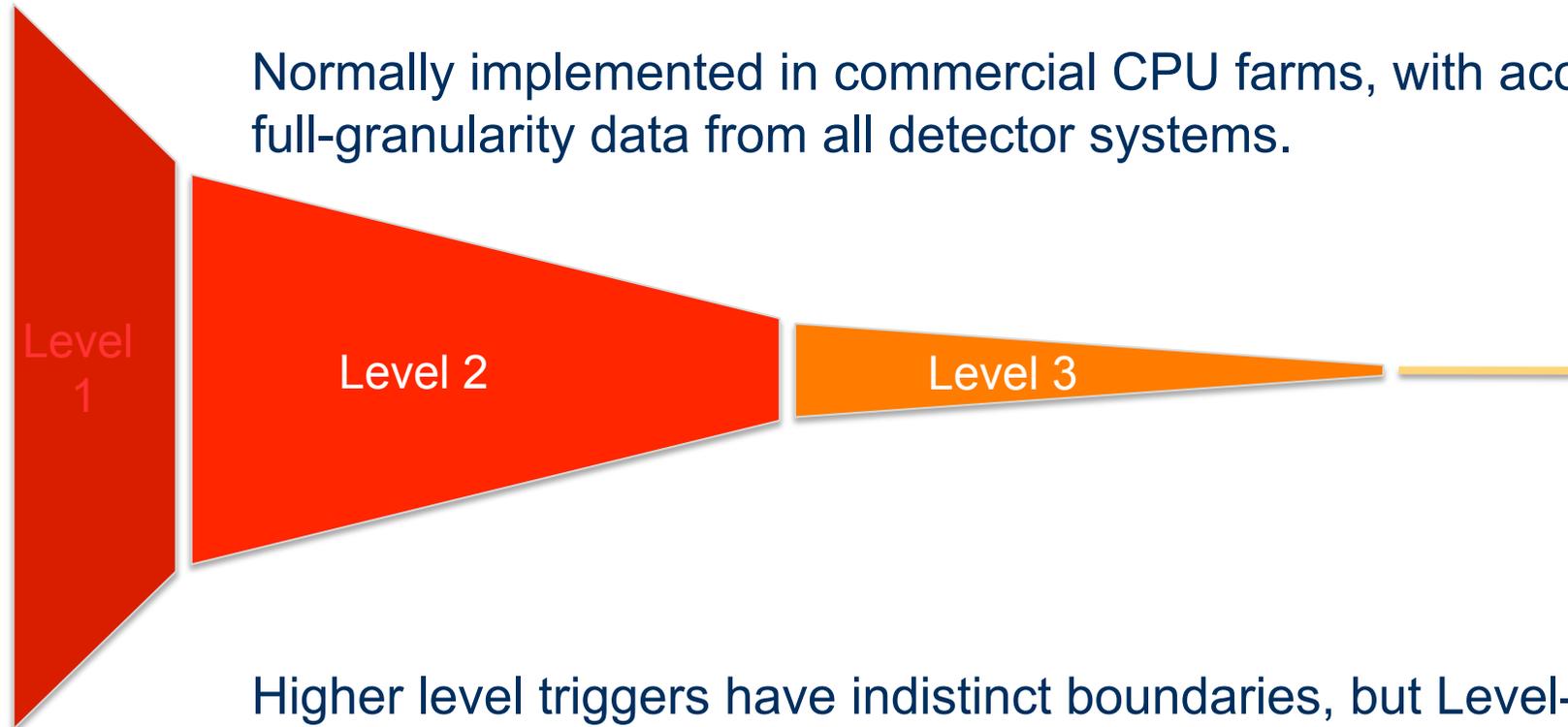
- electron, muon, jet, etc. candidates,
- transverse and missing transverse energy



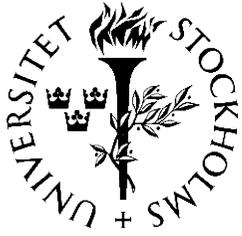
Higher-level triggers

More sophisticated processing of events accepted by Level-1.

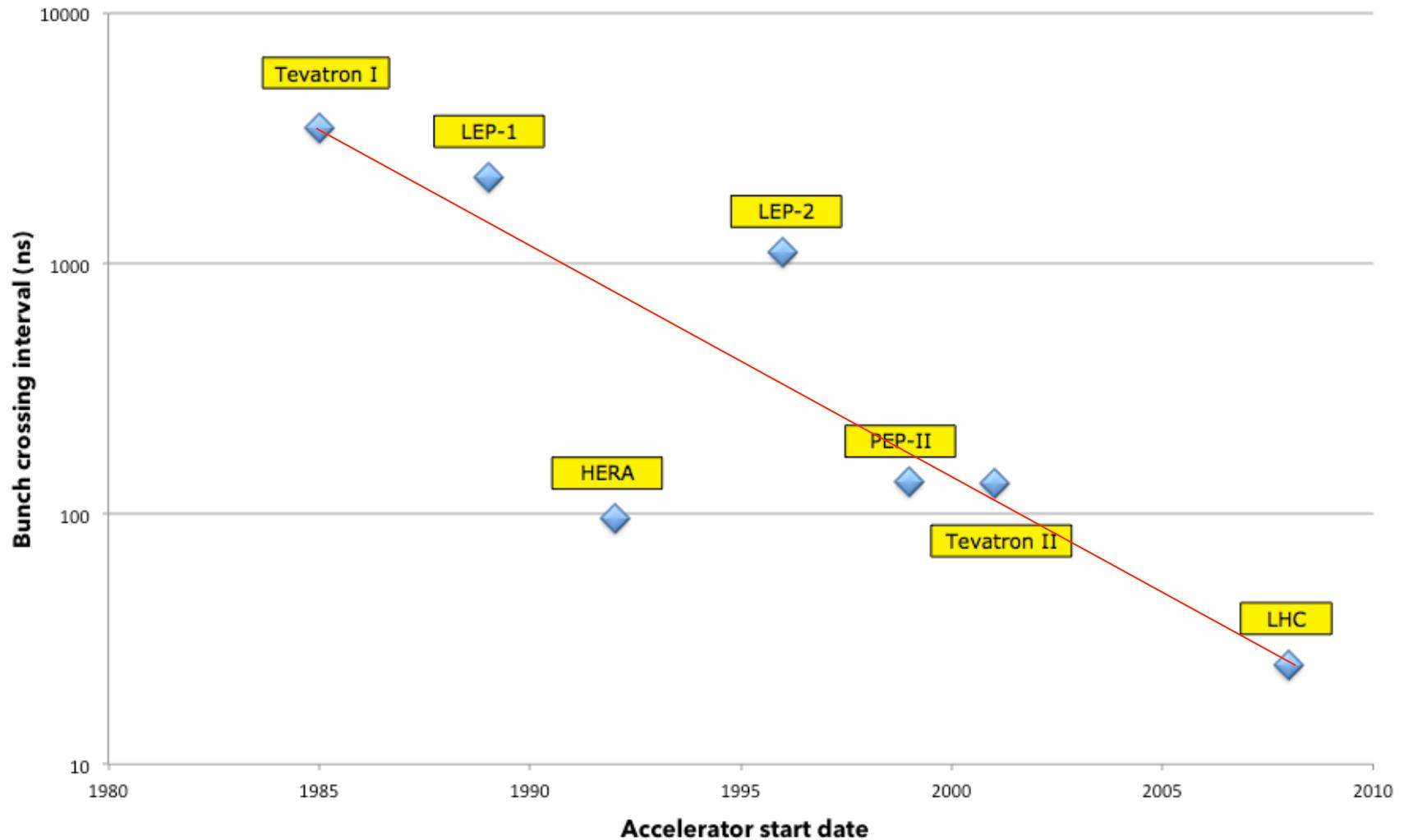
Normally implemented in commercial CPU farms, with access to full-granularity data from all detector systems.

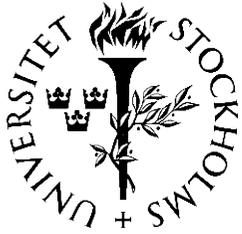


Higher level triggers have indistinct boundaries, but Level-2 typically deals with more detailed examination of trigger objects identified by Level-1, while Level-3 is closer to offline analysis



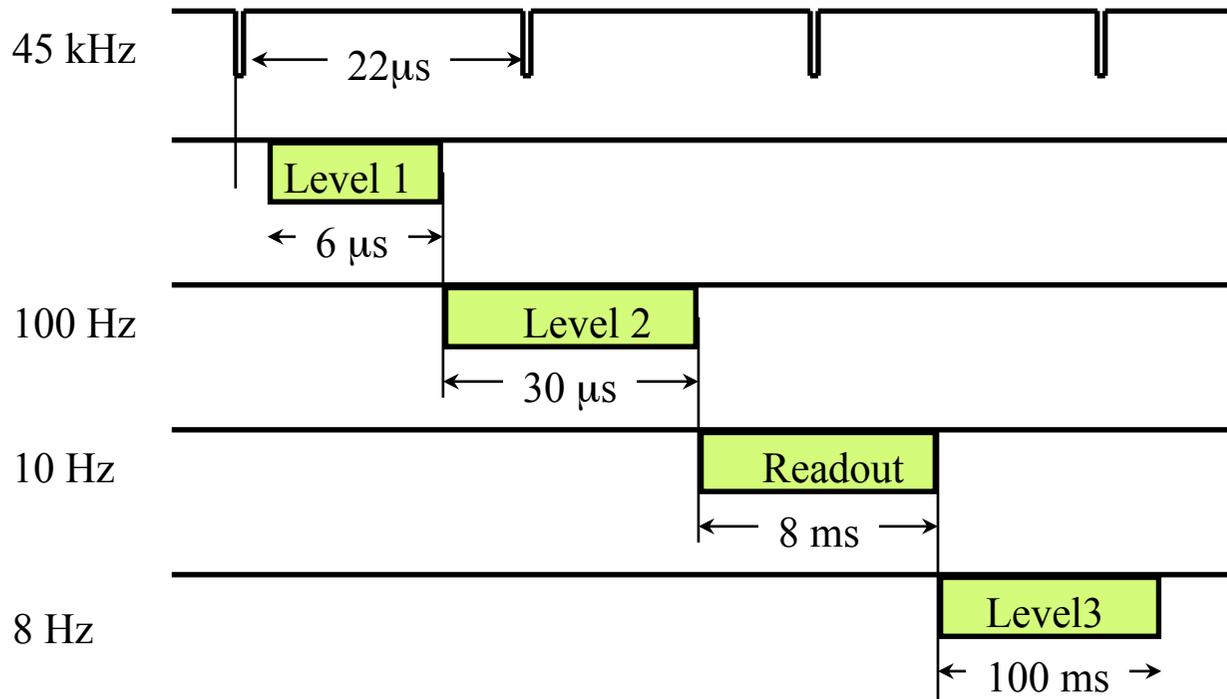
Collider BX rates



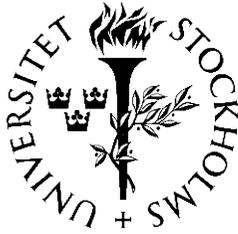


Timing at LEP

e^+e^- crossing rate: 45 kHz (LEP-1)

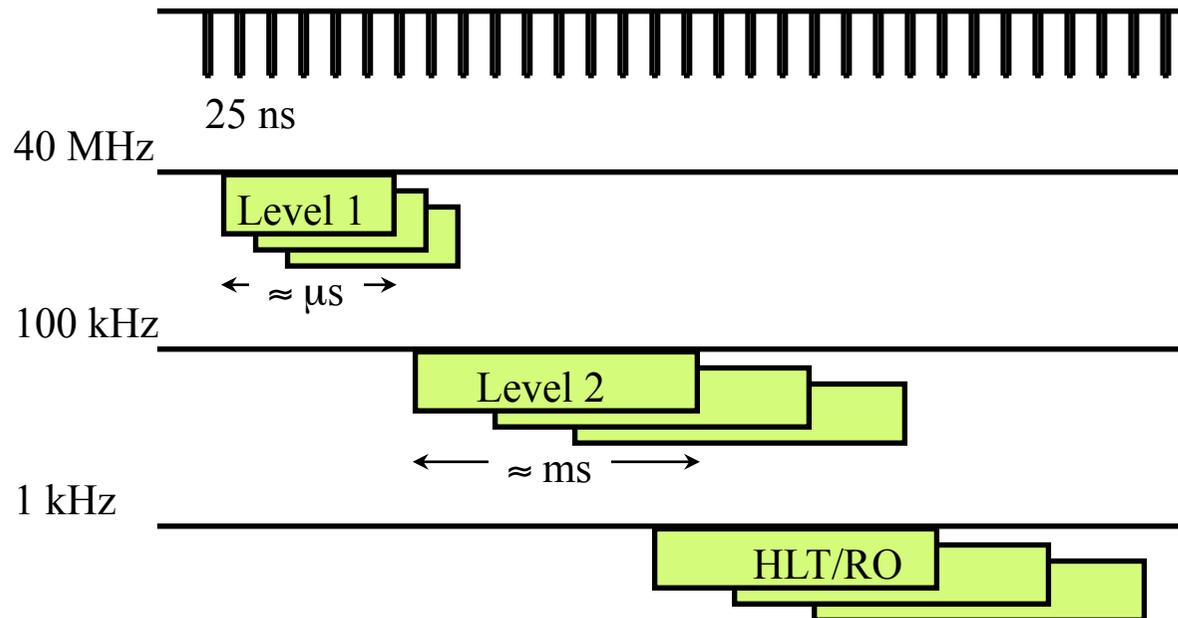


- Level 1 trigger latency less than BX interval → No dead time
- No overlapping events at any trigger level
- Most trigger/DAQ electronics off-detector

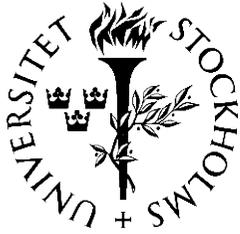


Trigger timing at LHC

p p crossing rate 40 MHz

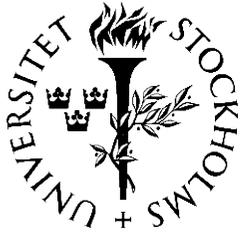


- Level 1 trigger latency \gg bunch interval
- Up to $10 \mu\text{s}$ for complete readout of an event
- Need pipelined trigger and DAQ

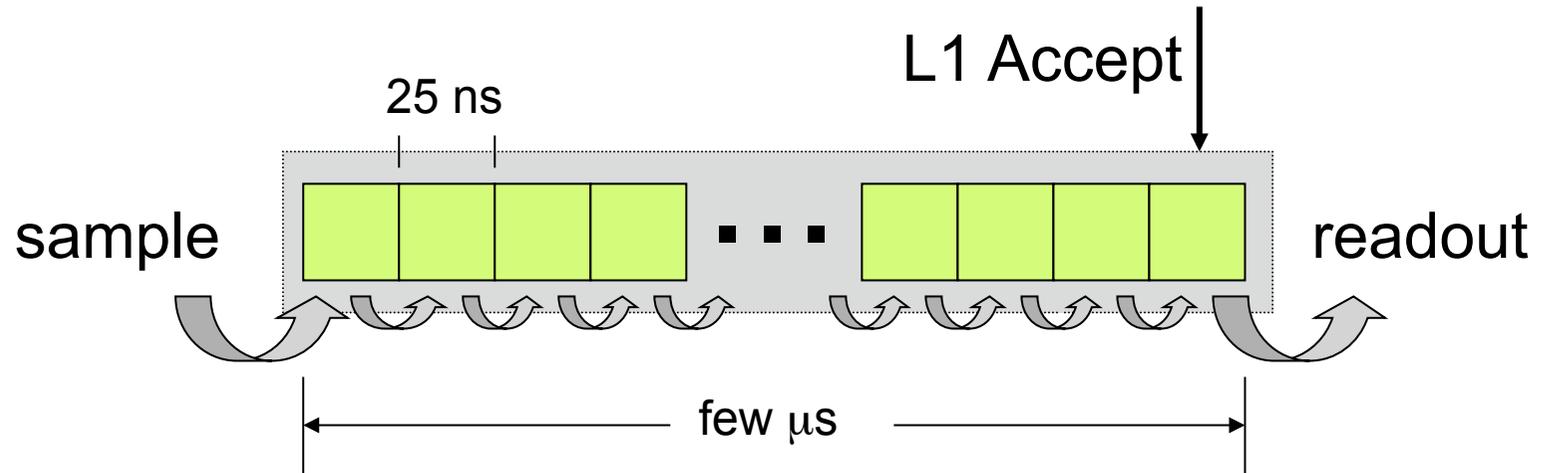


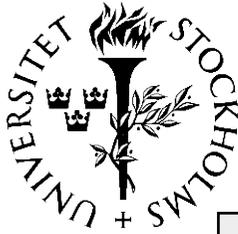
Pipelined Trigger/DAQ

- For every bunch crossing:
 - Sample and store all detector data in fixed-length pipeline buffers (\sim few μ s)
 - Reduced-granularity data sent to Level-1 trigger via low-latency path
 - Level-1 trigger performs pipelined processing, producing a Level-1 accept decision (L1A) for every BX.
- For events with L1A:
 - Data extracted from end of pipeline buffers, read out from the detector

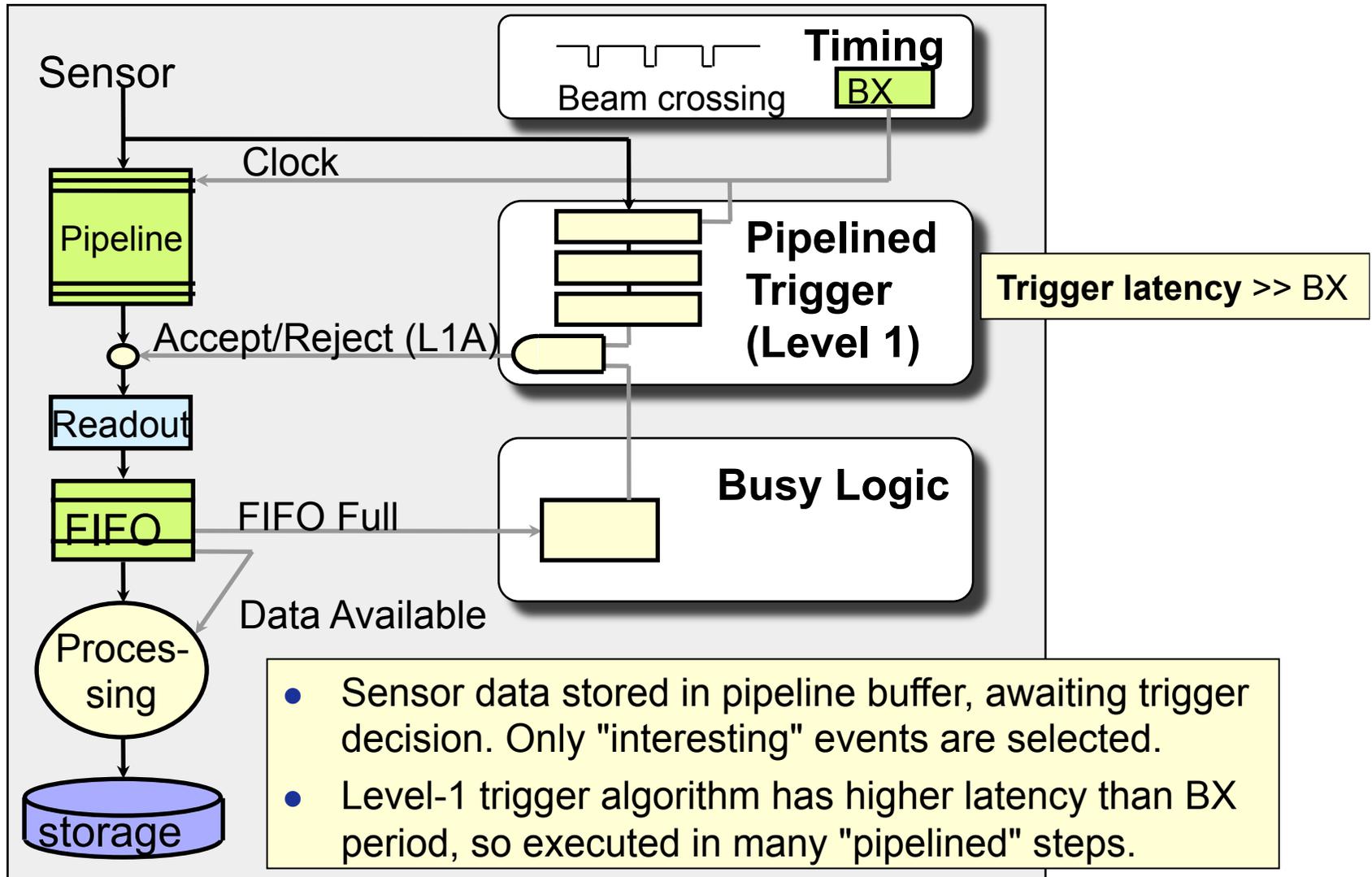


Pipeline buffer



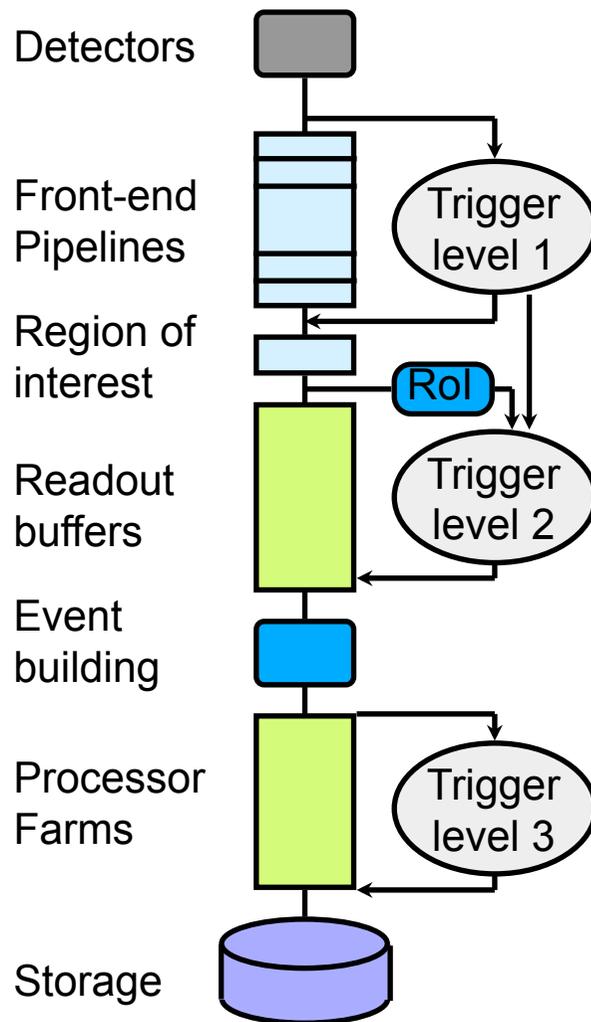


Simple pipelined trigger/DAQ

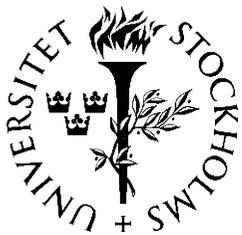




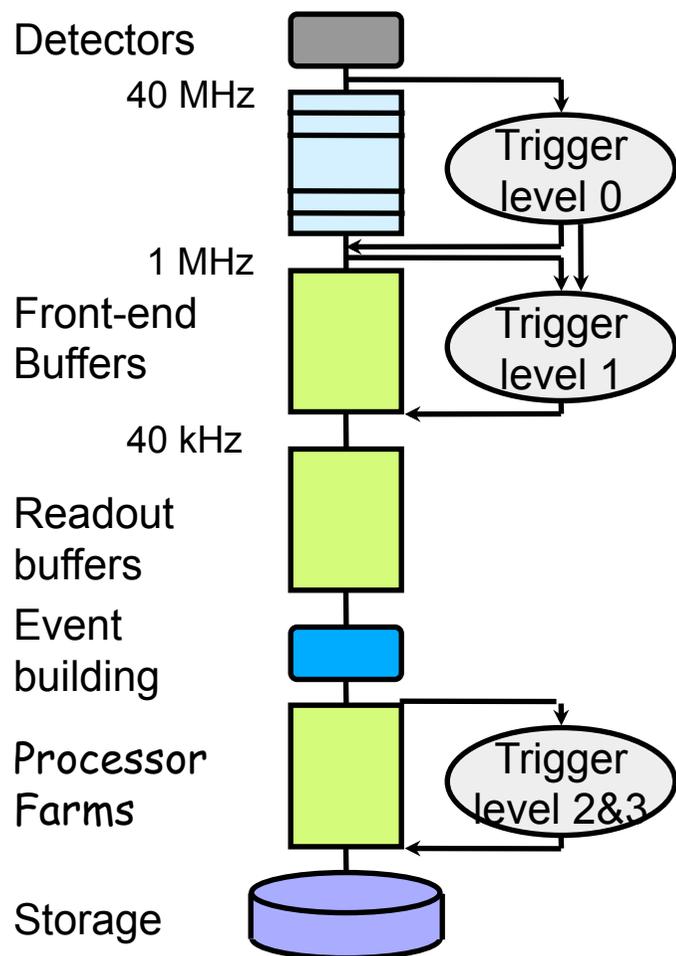
ATLAS trigger system (LHC)



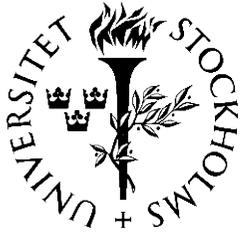
- **Level-1 (3.5 μ s) (custom processors)**
 - Isolated clusters, jets, ET in calorimeters
 - Muon trigger: tracking coincidence matrix.
- **Level-2 (100 μ s) (processor farm)**
 - Guided by Regions Of Interest (RoI) identified by Level-1
 - Select detector data routed to CPUs by routers and switches
 - Feature extractors (DSP or specialized) perform refined object ID algorithms
 - Staged local and global processors
- **Level-3 (\approx ms) (commercial processors)**
 - Reconstruct the event using all data
 - Select of interesting physics channels



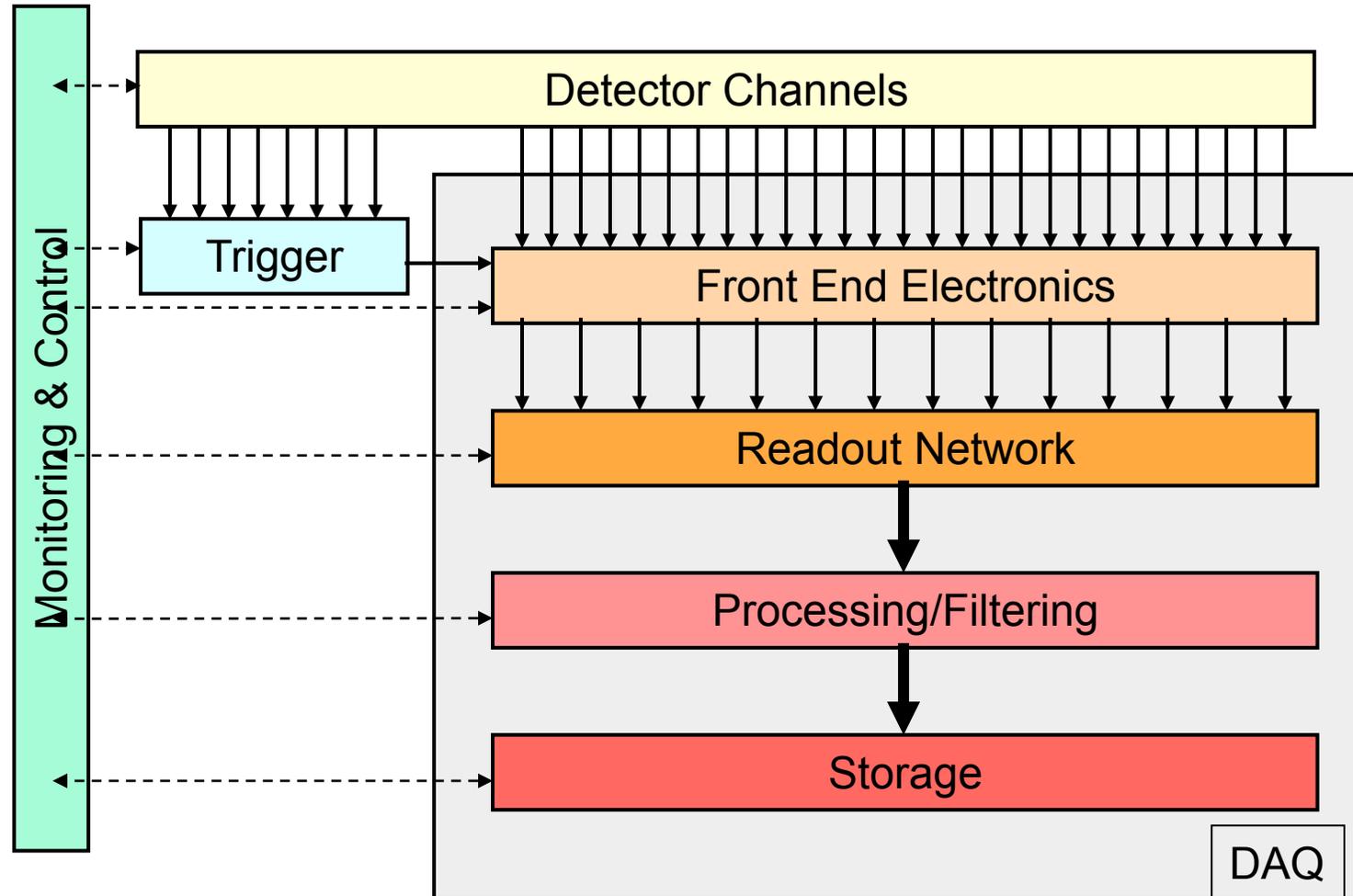
LHCb trigger system

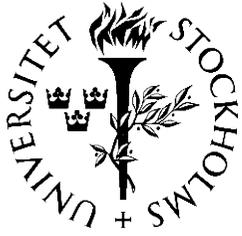


- **Level-0 (4 μ s) (custom hardware)**
 - High p_T electrons, muons, hadrons
 - Pile-up veto.
- **Level-1 (1000 μ s) (specialized processors)**
 - Vertex topology (primary & secondary vertices)
 - Tracking (connecting calorimeter clusters with tracks)
- **Level-2 (\approx ms) (commercial processors)**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



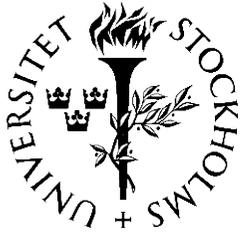
TDAQ and control



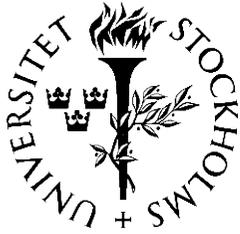


Or for real:





Level-1 Algorithms



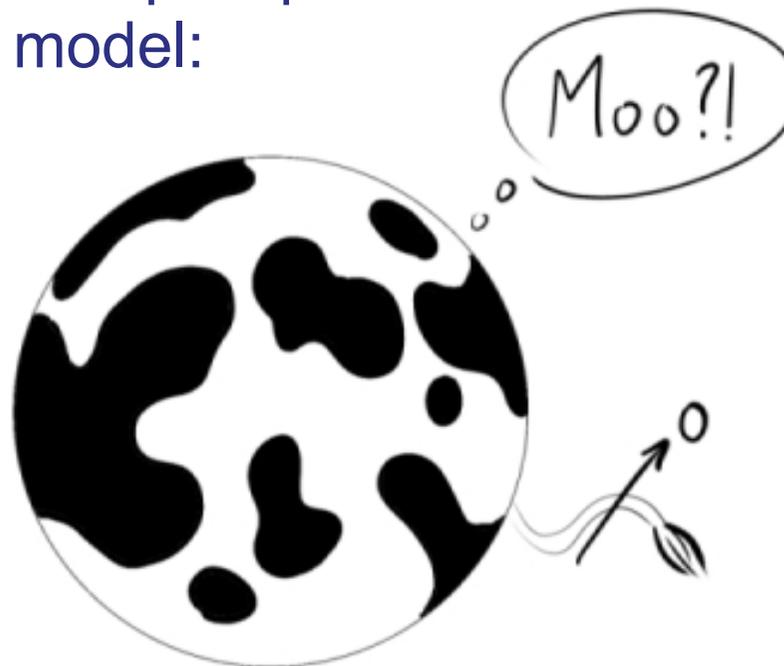
The challenge:

- Need to identify and trigger on events with “interesting” features:
 - Isolated electrons and taus, jets, missing transverse momentum, etc...
- HLT (CPUs) perform sophisticated identification algorithms at full granularity
 - But Level-1 only has low-granularity data
 - Limited time for execution
- So how do we identify these features at Level-1?



Start with a spherical cow....

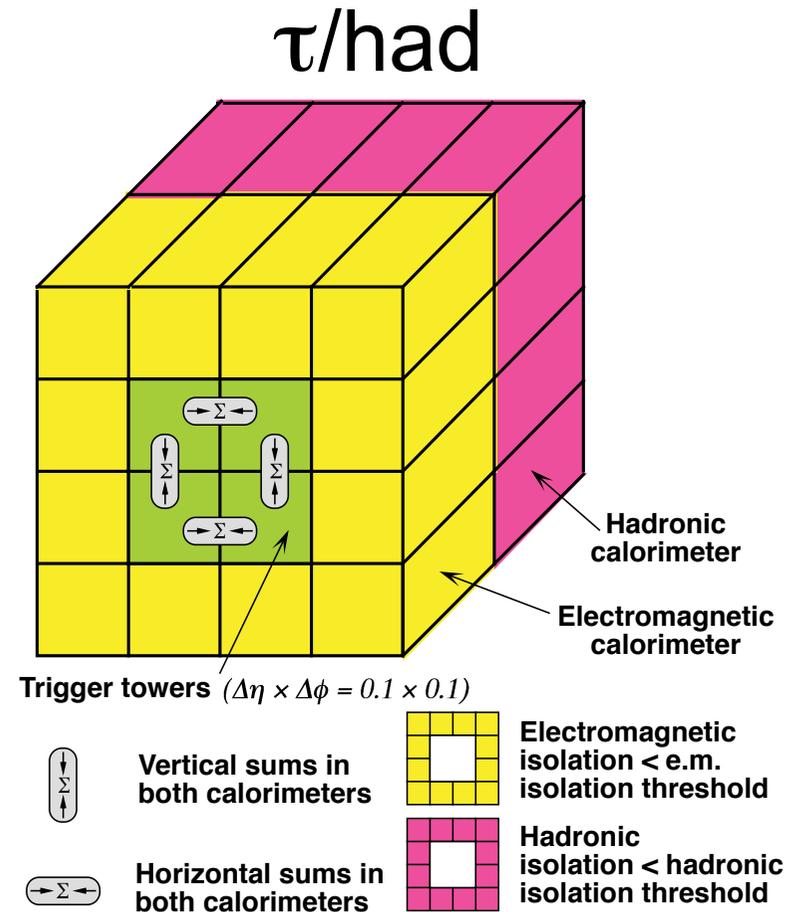
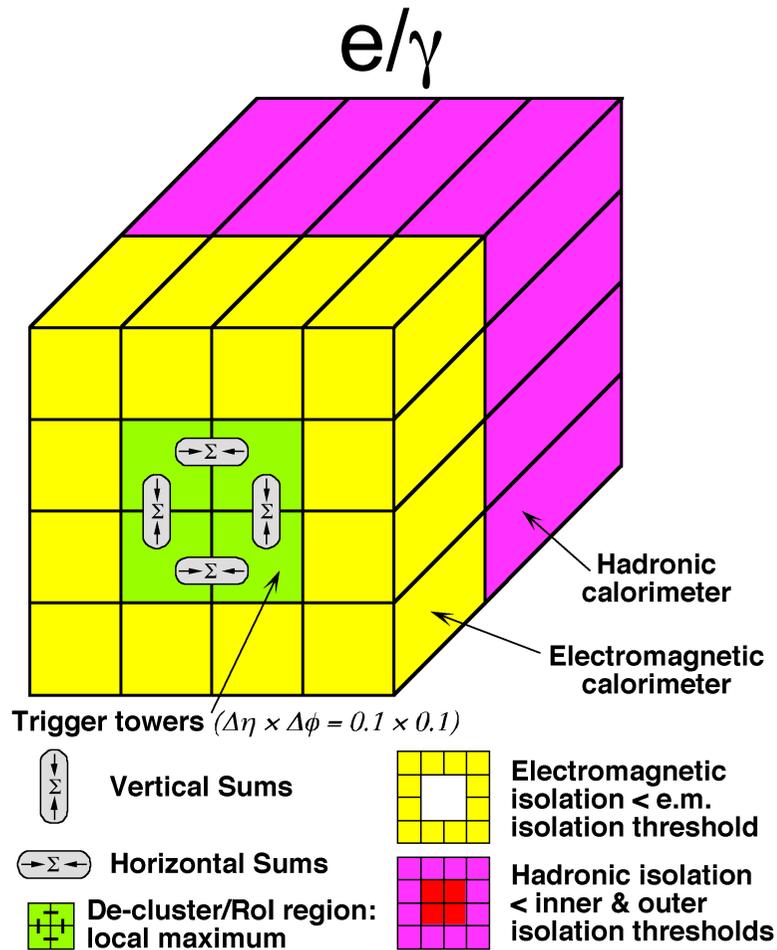
Approximating a complex problem
with a simplified model:

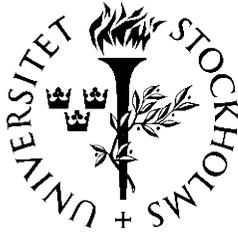


Consider a spherical cow
of radius R ... \square

Level-1 triggers start with e.g. rectangular electrons...

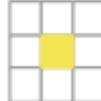
ATLAS e/ γ and τ algorithm



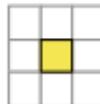
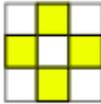


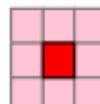
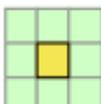
CMS e/ γ algorithm

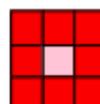
Trigger Primitive Generator

Fine grain Flag Max of (, , , ) & Sum ET 

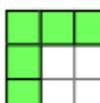
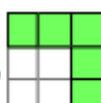
Regional Calorimeter Trigger

E_T cut  + Max () > Threshold

Longitudinal cut (H/E)  AND /  < 0.05

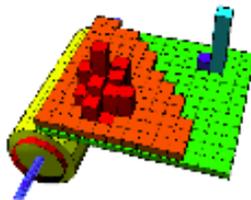
Isolation, Hadronic & EM  < 2 GeV

AND

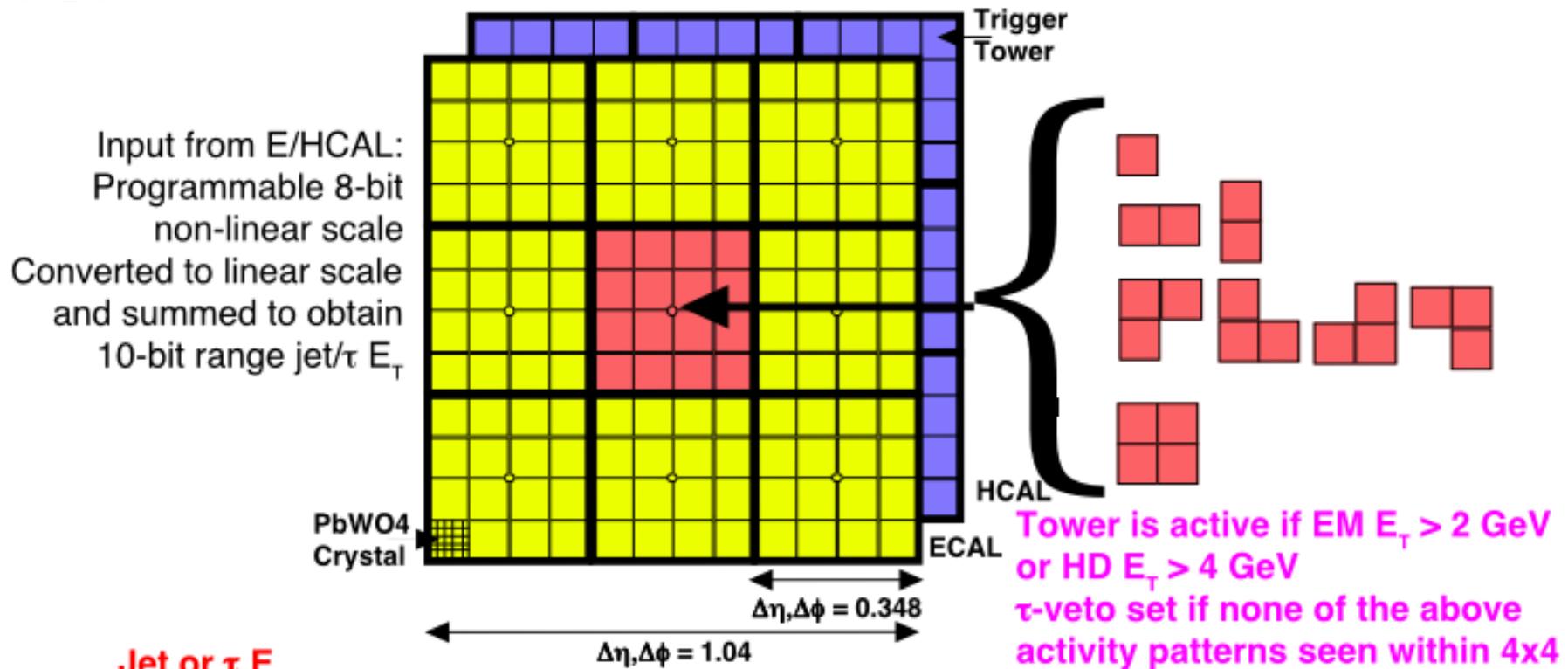
One of (, , , ) < 1 GeV



ELECTRON or PHOTON



CMS τ / Jet algorithm



Jet or τ E_T

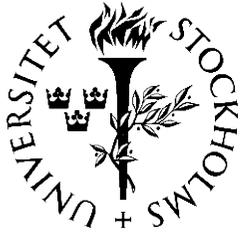
- 12x12 trigger tower E_T sums in 4x4 region steps with central region $>$ others
- Larger trigger towers in HF but \sim same jet region size, $1.5 \eta \times 1.0 \phi$

τ algorithm (isolated narrow energy deposits), within $-2.5 < \eta < 2.5$

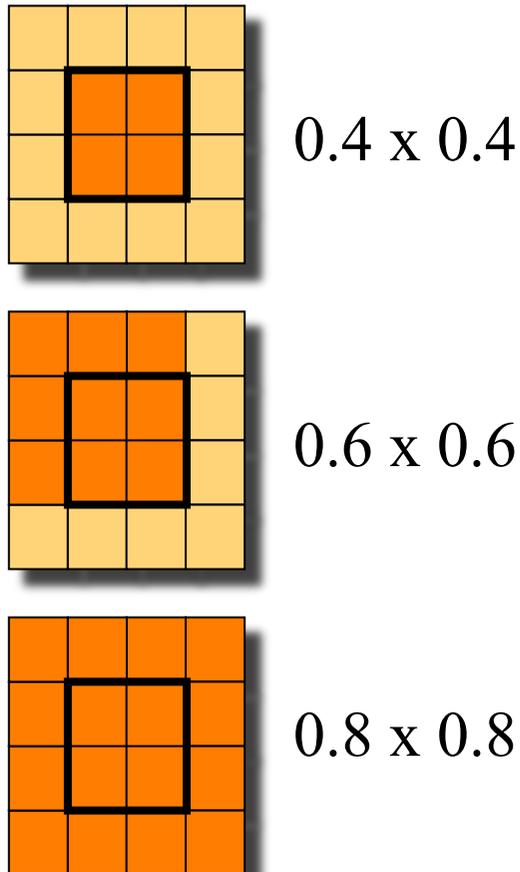
- Redefine jet as τ jet if none of the nine 4x4 region τ -veto bits are on

Output

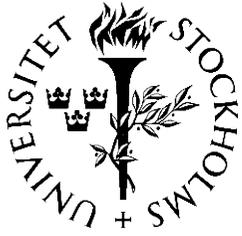
- Top 4 τ -jets and top 4 jets in central rapidity, and top 4 jets in forward rapidity



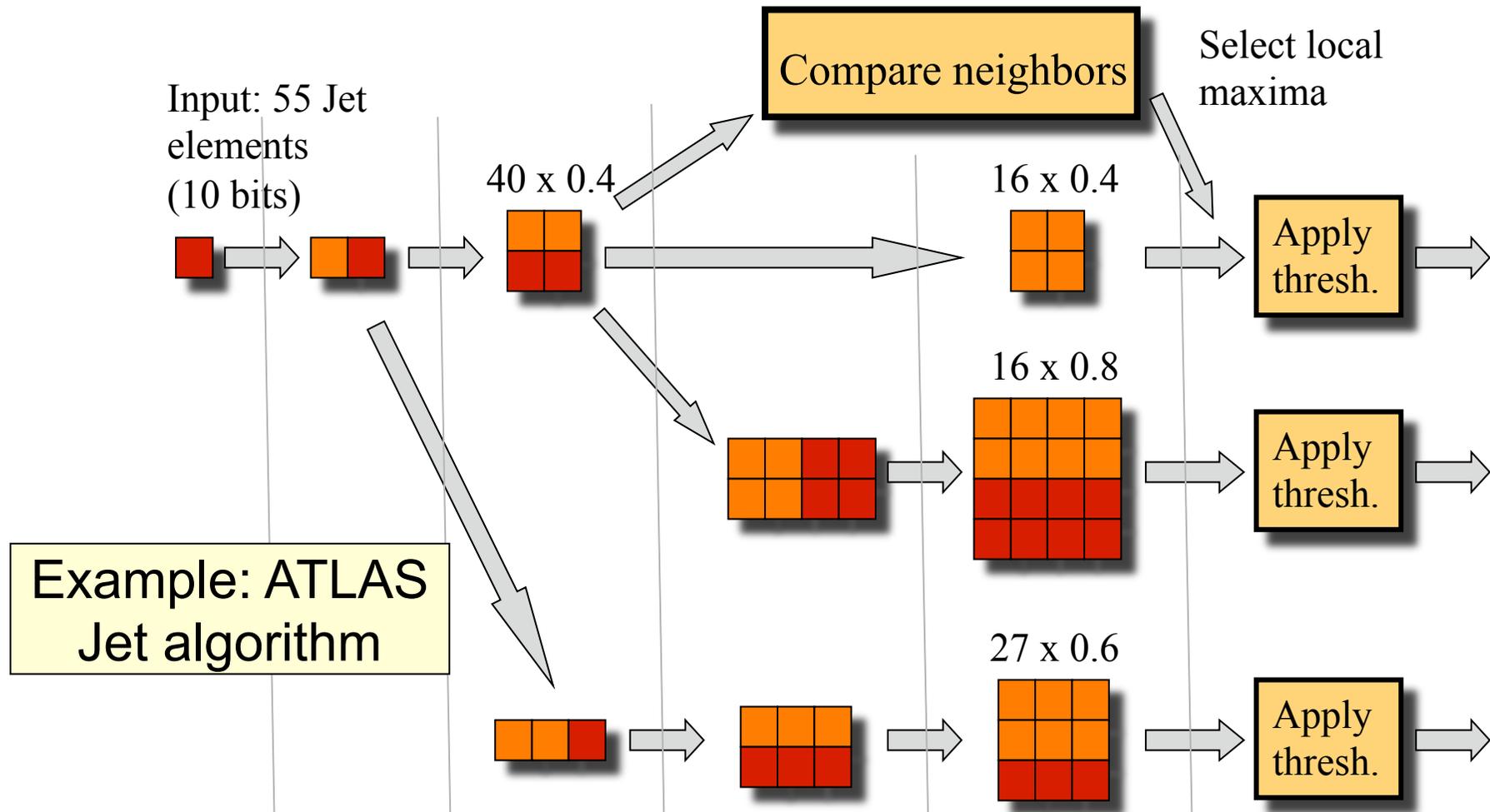
ATLAS Jet algorithm

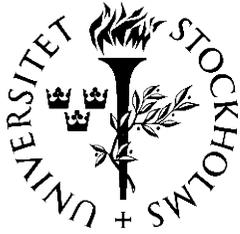


- Local maximum in a 0.4 x 0.4 window sliding by 0.2
- Compare transverse energy sum against thresholds for Jet window(s) surrounding the local maximum
 - Jet sizes: 0.4, 0.6, 0.8
 - 8 Jet definitions available:
 - Jet size
 - Et threshold



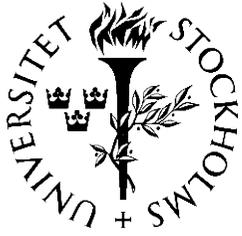
Pipelined algorithm design



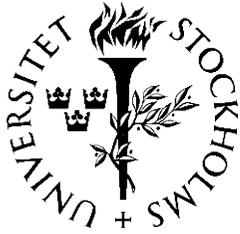


Triggering involves trade-offs

- Real electrons and jets are not rectangular!
 - But it's easier and faster this way in hardware
 - HLT can do a more realistic job later
 - If the event makes it that far...
- Reality:
 - Over 99% of what Level-1 accepts is junk!
 - The job of Level-1 is to reduce rates
 - While not throwing away too much good physics
 - Latency/bandwidth/cost are major constraints
 - Do the “best we can” within these constraints

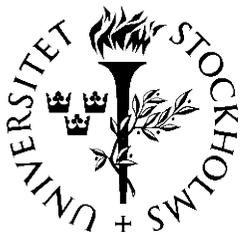


Dead Time



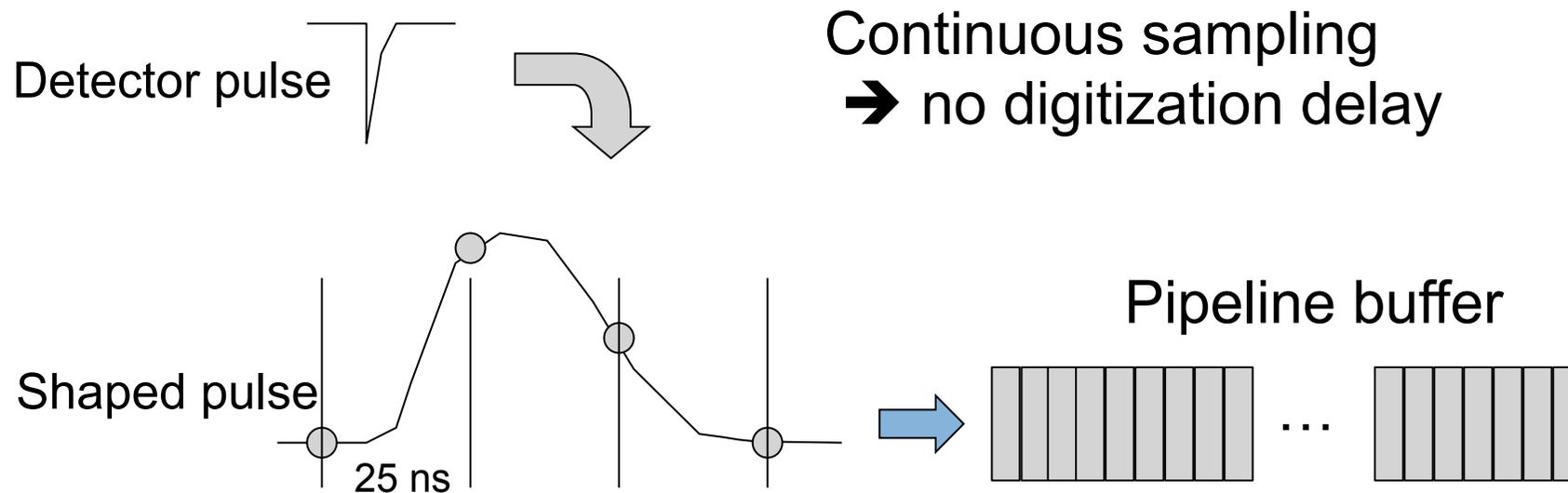
Dead Time

- Definition: The fraction of time during which the system cannot record data
- Sources of dead time:
 - ADC conversion time
 - ADCs take a finite time to sample and digitize a single event. Can't digitize new data until done.
 - Readout dead time
 - Finite time to read out multiple ADC samples for a triggered event
 - “Busy” condition in derandomizing buffers
 - Inhibit triggering new events to prevent overflow
 - Operational dead time
 - Detector readiness, etc...



ADC conversion time

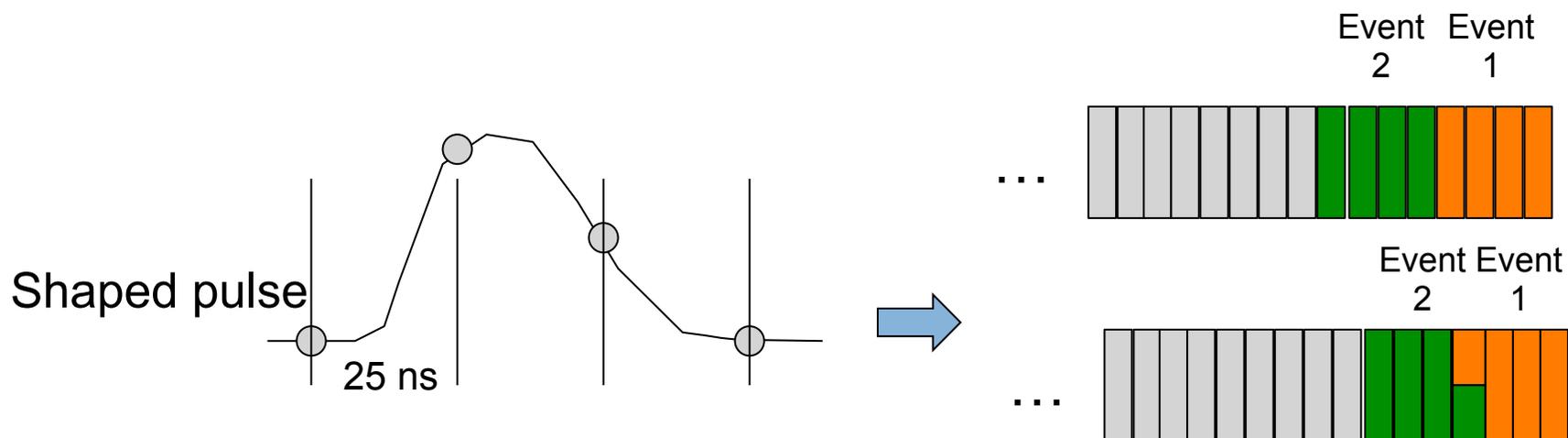
- Collider experiments use fast, pipelined ADCs
 - Digitize every BX, store in pipeline buffer
 - ADC normally not a source of dead time

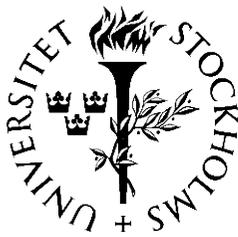




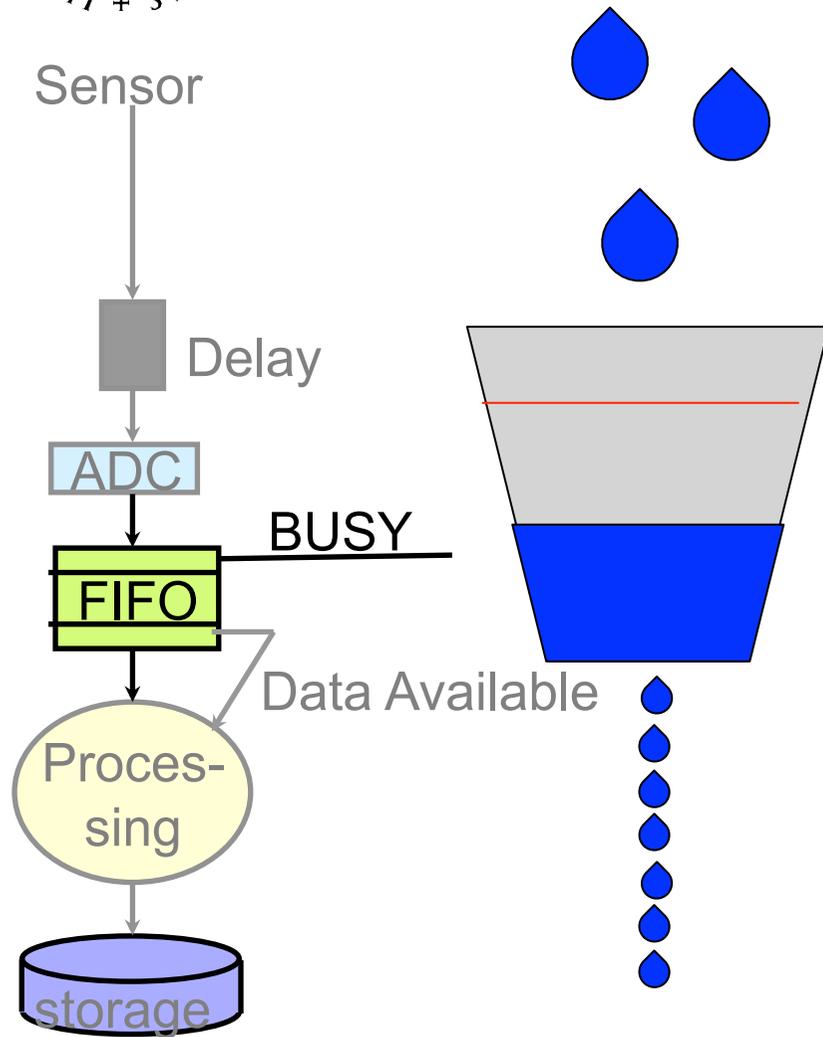
Readout dead time

- Read out multiple samples per event
 - Simpler systems can have problems reading out overlapping events
 - Result: cannot trigger events too close in time
 - Can avoid this with digital readout





“Busy” condition



- “Leaky bucket” paradigm for derandomizing buffers
- Input at random intervals
- ~ Constant output rate
- If “high- water mark” passed, block trigger
- Maximum trigger rate:
 - $R_{\max} = 1/T_{\text{readout}}$
- Use sufficiently long buffers to minimize dead time below R_{\max} .



Calculating the dead time

Analytic solution* assuming a constant time to read out each event:

$$D_N = 1 - \frac{S_N}{1 + \rho S_N} \text{ where } \rho = R\tau \text{ and}$$

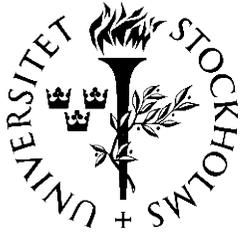
$$S_N = e^{N\rho} \sum_{j=0}^{N-j} \frac{(N-j)^j (-\rho e^{-\rho})^j}{j!}$$

Variables:

- $N-1$ buffers
- R : Input rate (trigger)
- τ : Readout time

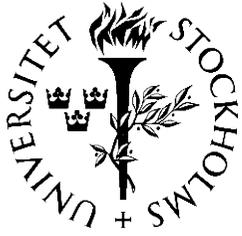
Reality: τ is not always constant, and the dead time model is often more complicated. Monte Carlo simulation provides better accuracy

* Source: G.P. Heath, "Dead time due to trigger processing in a data acquisition system with multiple event buffering", Nuclear Instruments and Methods in Physics Research, A278 (1989) 431-435.

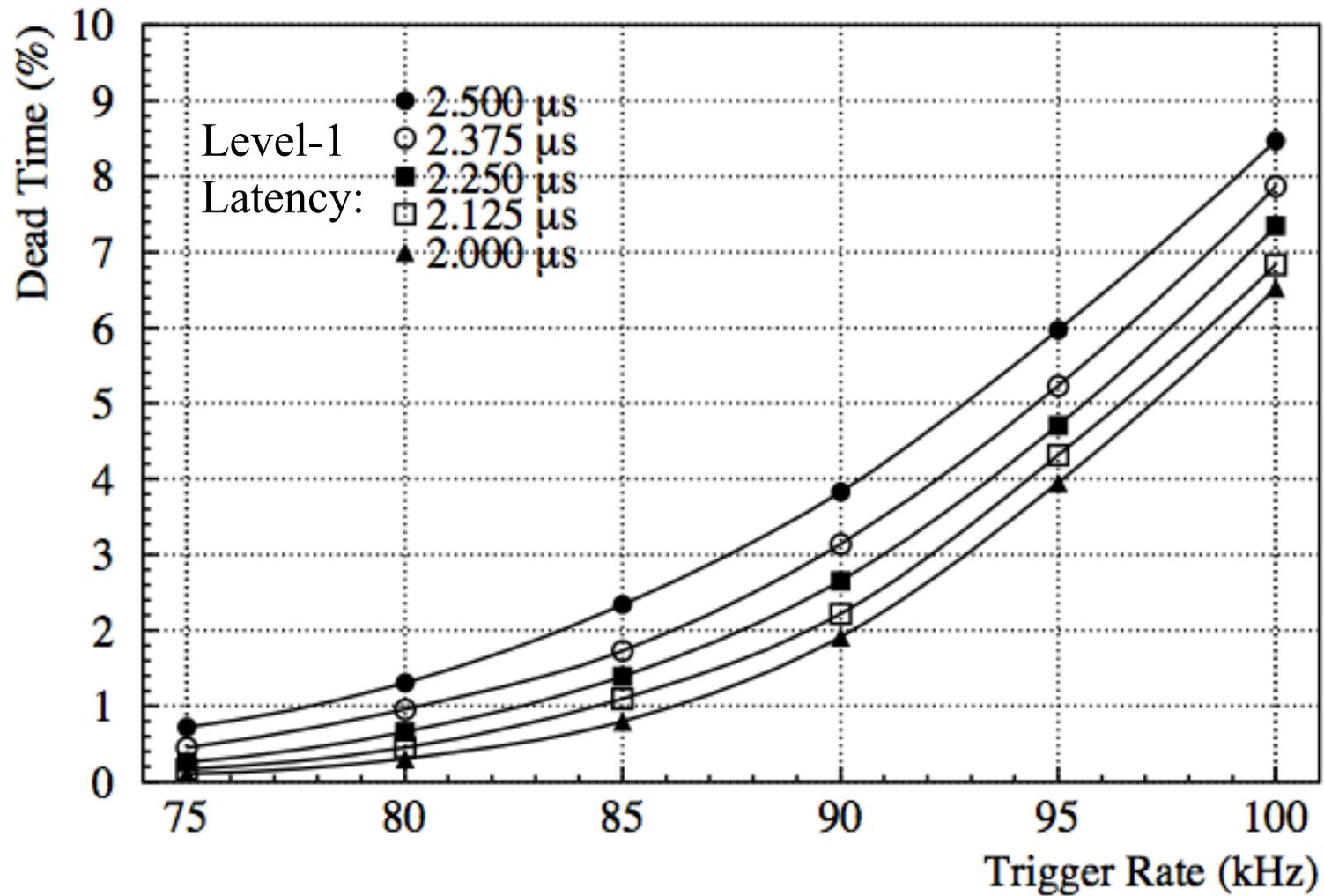


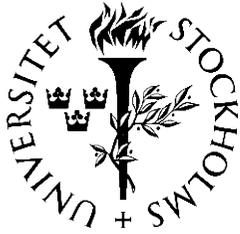
Example: ATLAS Lar calorimeter

- Basic parameters:
 - BX rate: 40.08 MHz (25 ns)
 - Max trigger rate: 75 kHz (upgrade to 100)
 - Readout time (5 samples/event): 10.6 μ s
- Other complicating factors include:
 - At least five BXs (125 ns) between two Level-1 accepts (L1A) (readout dead time)
 - Not all bunches in LHC filled
 - Nominally 2808 of 3564 (trains of 72 bunches)
 - No L1A expected for empty bunches
 - Analog pipeline buffers are 144 samples long
 - Divided between pipeline and derandomizer FIFO
 - So trigger latency affects FIFO length (& dead time)



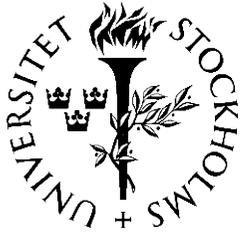
Simulated LAr dead time





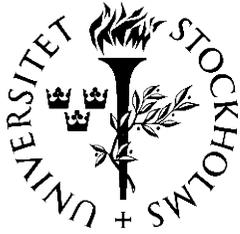
In my next two lectures...

- Algorithms and architectures
 - More detailed examples of trigger systems, and how they are built
 - The “tools” available for collider detector triggers, and how they can be used
- New directions: SLHC
 - SLHC planning and implications
 - New technologies and architectures
 - TDAQ upgrades for SLHC



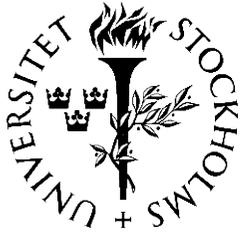
TDAQ lab exercise

- Field Programmable Gate Array (FPGA) tutorial
- Specify in VHDL language:
 - Simple Boolean logic gate (e.g. AND)
 - A coincidence counter
- Implement and test in real hardware



Homework

- Write a Monte Carlo dead time simulation of a semi-realistic LHC readout
 - (Based on ATLAS LAr)
 - Use 'leaking bucket' algorithm
- Parameters:
 - 40 MHz bunch crossing rate (25 ns)
 - L1 accept rate (random): 75 kHz
 - Readout time per event: 10.6 μ s
 - Minimum time between L1 accepts: 125 ns (5 bunch crossings)
- Investigate the questions:
 - How deep must the derandomiser buffer be in order to keep dead time below 1%?



Questions?