Imperial College London

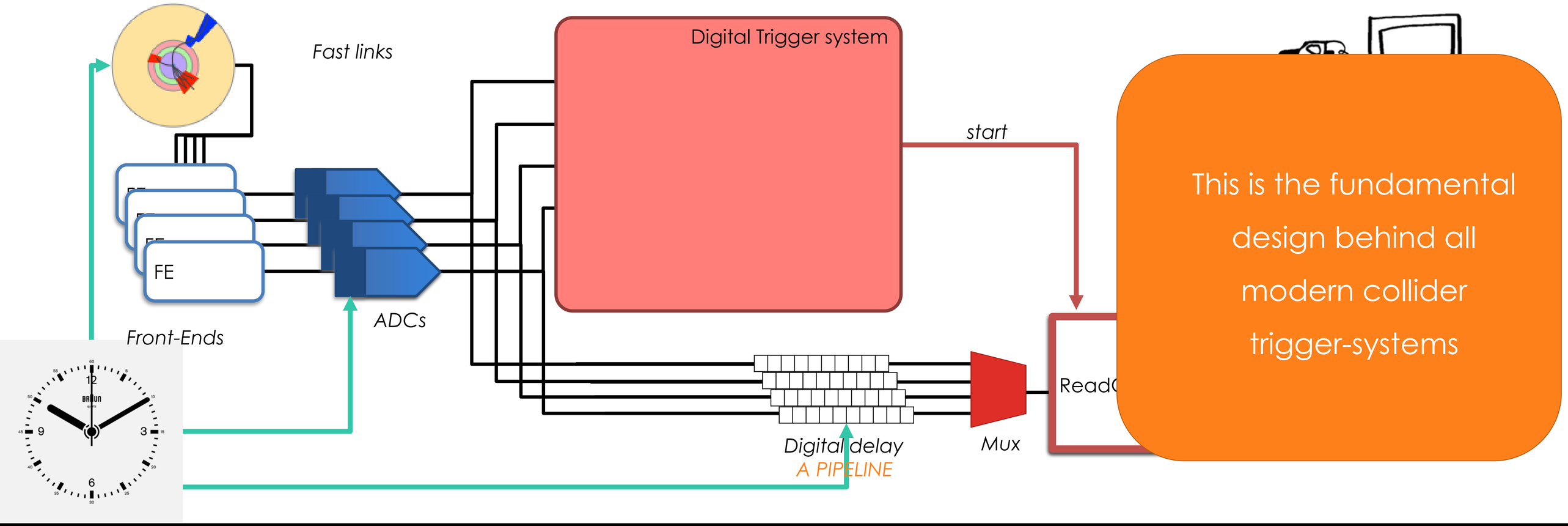
TRIGGER AND DATA-ACQUISITION: PART II

UK Advanced Instrumentation Course 2022

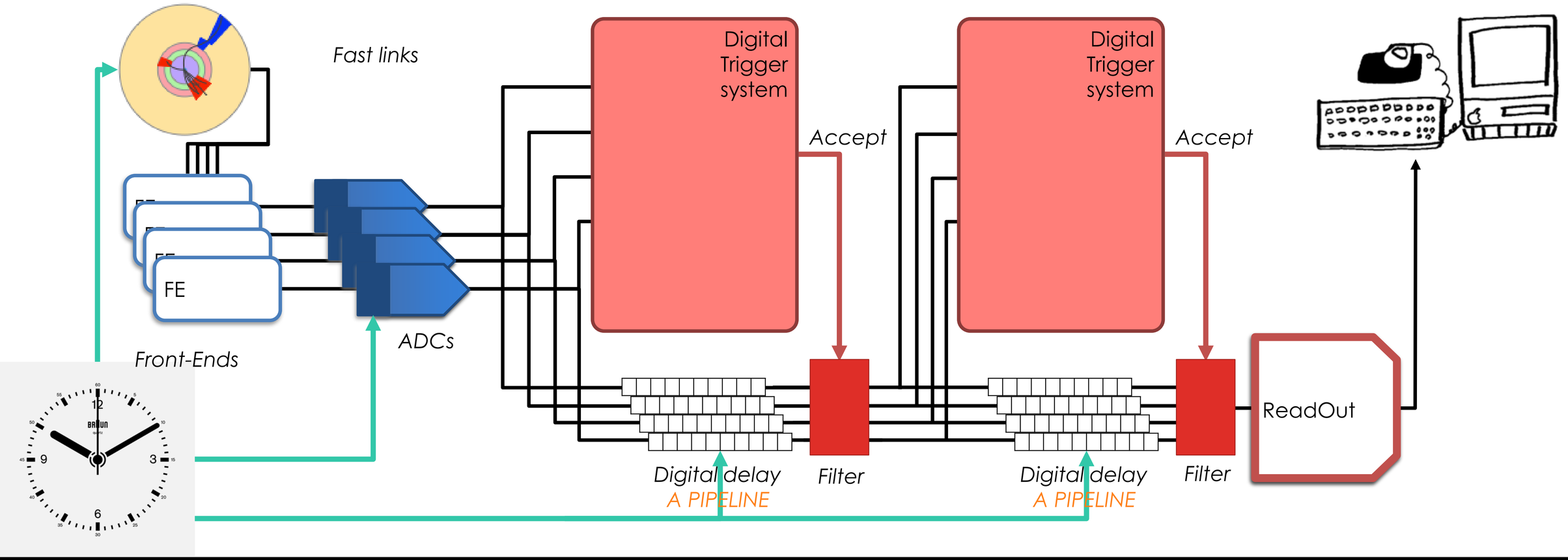
Andrew W. Rose, Imperial College London

awr01@imperial.ac.uk

A SIMPLE TRIGGER SYSTEM: DIGITAL TRIGGERS



A TRIGGER SYSTEM: MULTILAYER TRIGGERS



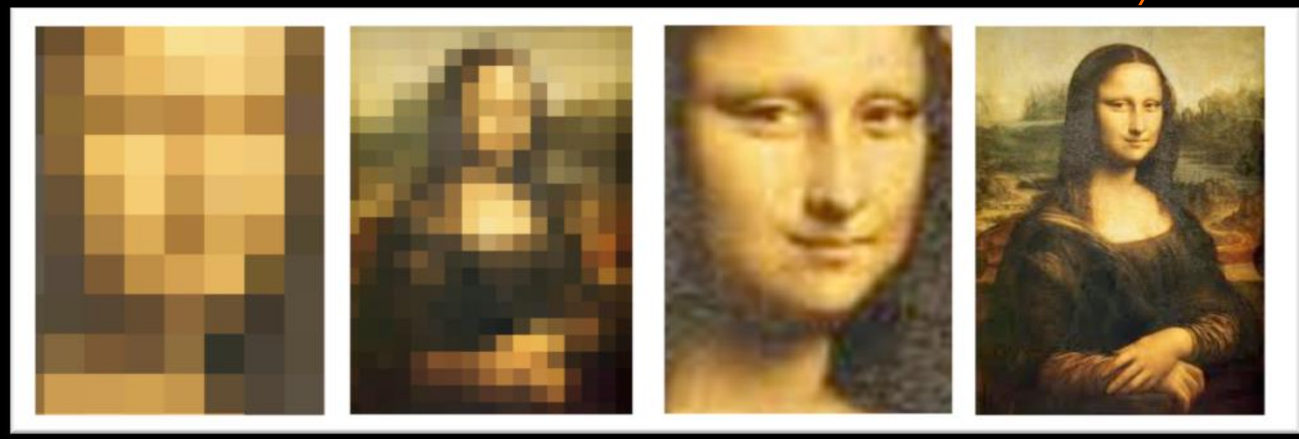
MULTILAYER TRIGGERS

- Each stage reduces the rate, so later stages have longer latency
- Complexity of algorithms increases at each level
- Dead-time is the sum of the trigger dead-time, summed over the trigger levels, and the readout dead-time

MULTILAYER TRIGGERS

- Adopted in large experiments
  - More and more complex algorithms are applied on lower and lower data rates
- Efficiency for the desired physics must be kept high **AT ALL LEVELS**, since rejected events are lost for ever

Level-1 *Level-2* *Level-3* *Analysis*

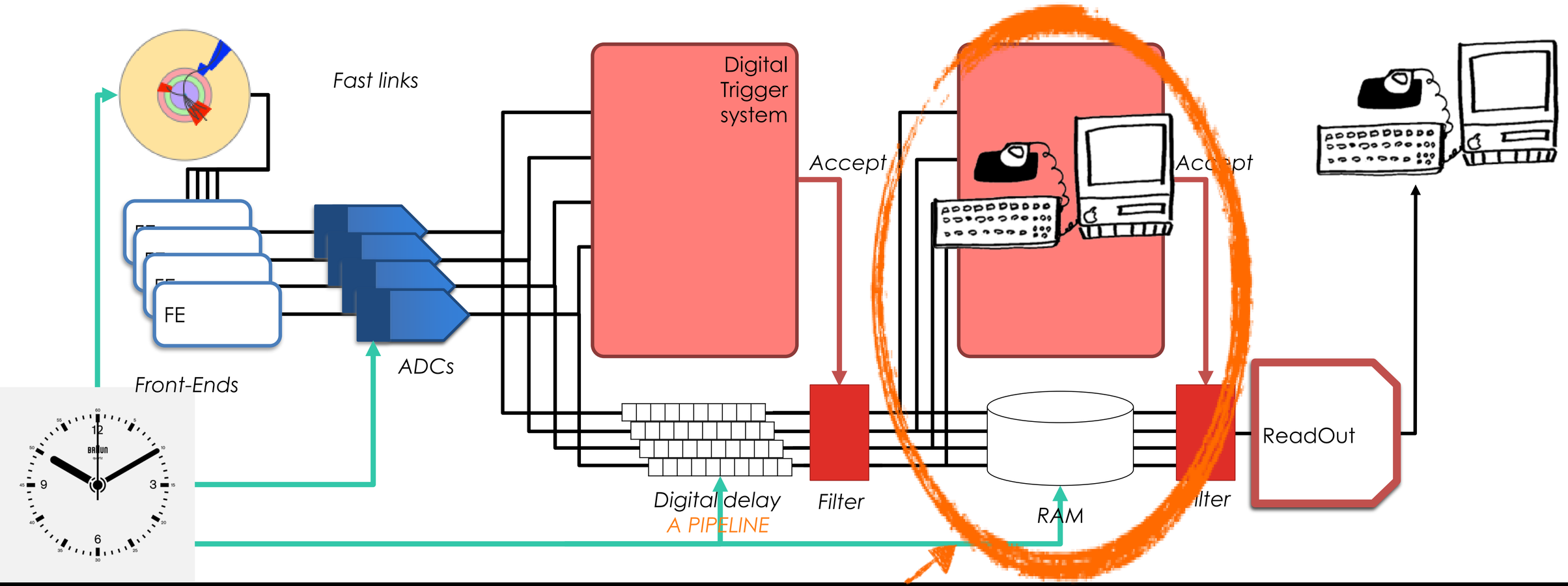


- Low latency
- Full event rate
- Small event fragment size
- Lower algorithmic complexity
- Access to coarse granularity information

- Longer latency
- Lower event rate
- Larger event fragment size
- Higher algorithmic complexity
- Access to higher granularity information

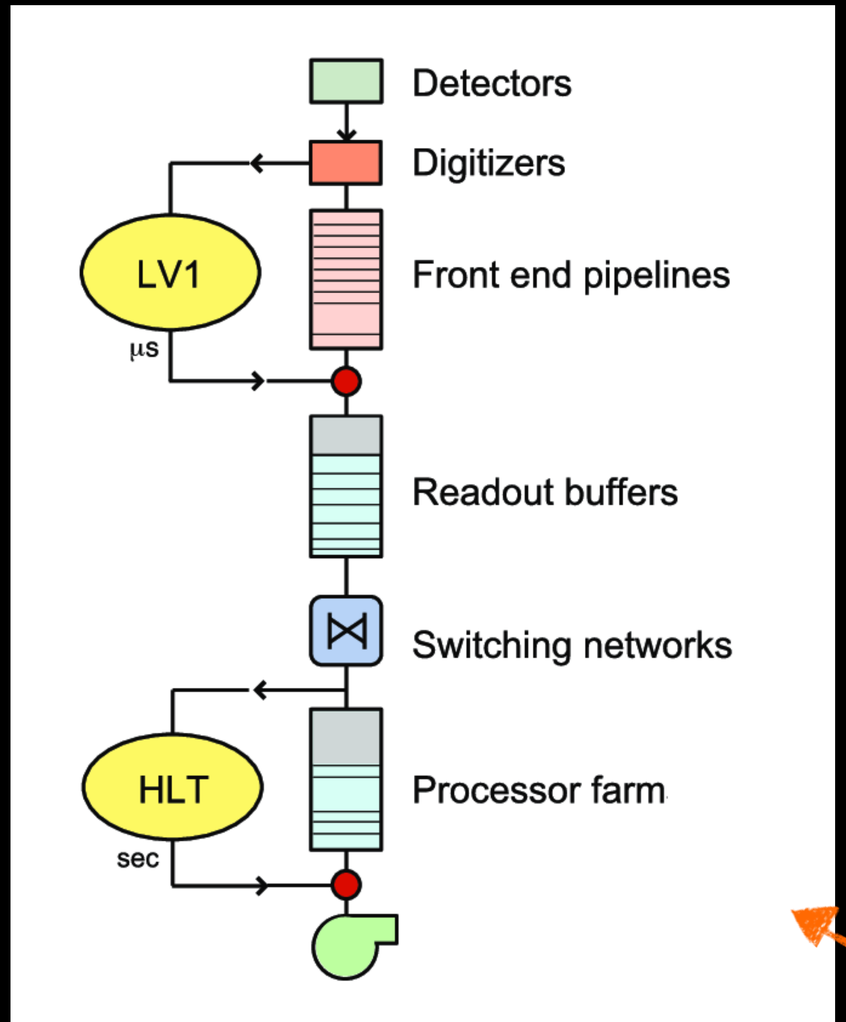
LHC experiments @ Run1	
Experiment	**Number of Levels (excl. analysis)**
ATLAS	**3**
CMS	**2**
LHCB	**3**
ALICE	**4**

A TRIGGER SYSTEM: MULTILAYER TRIGGERS



If your input rate is low enough

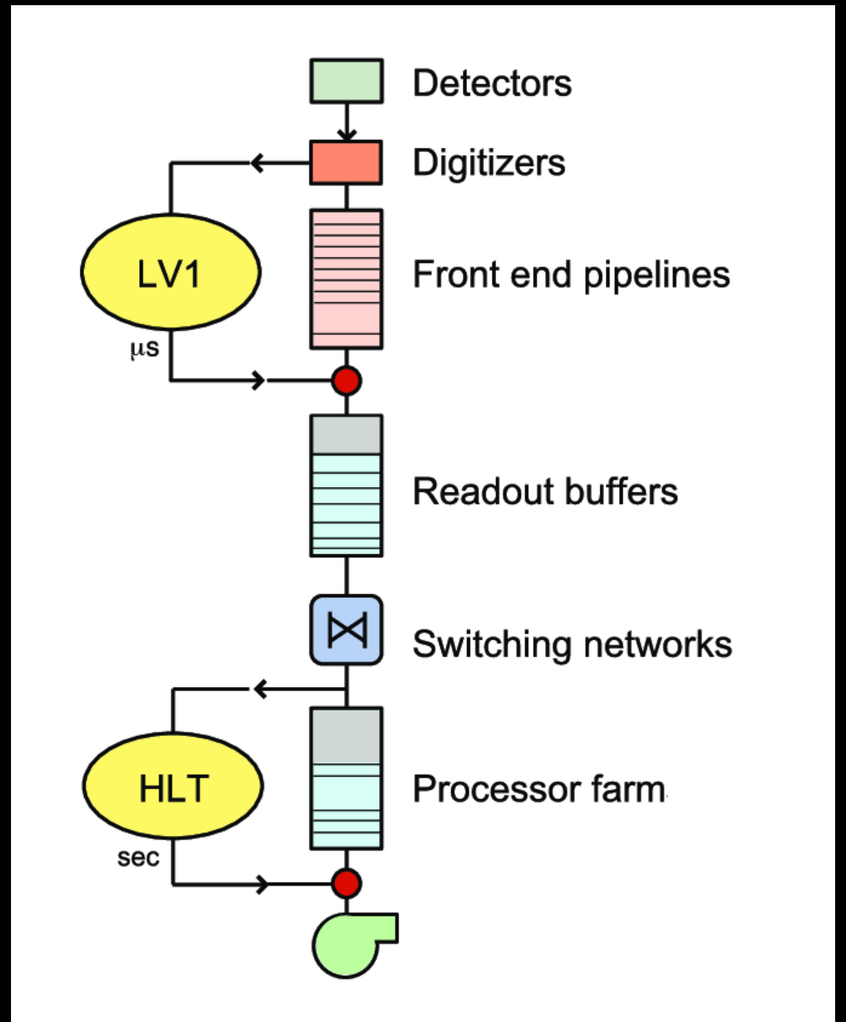
A TRIGGER SYSTEM: MULTILAYER TRIGGERS

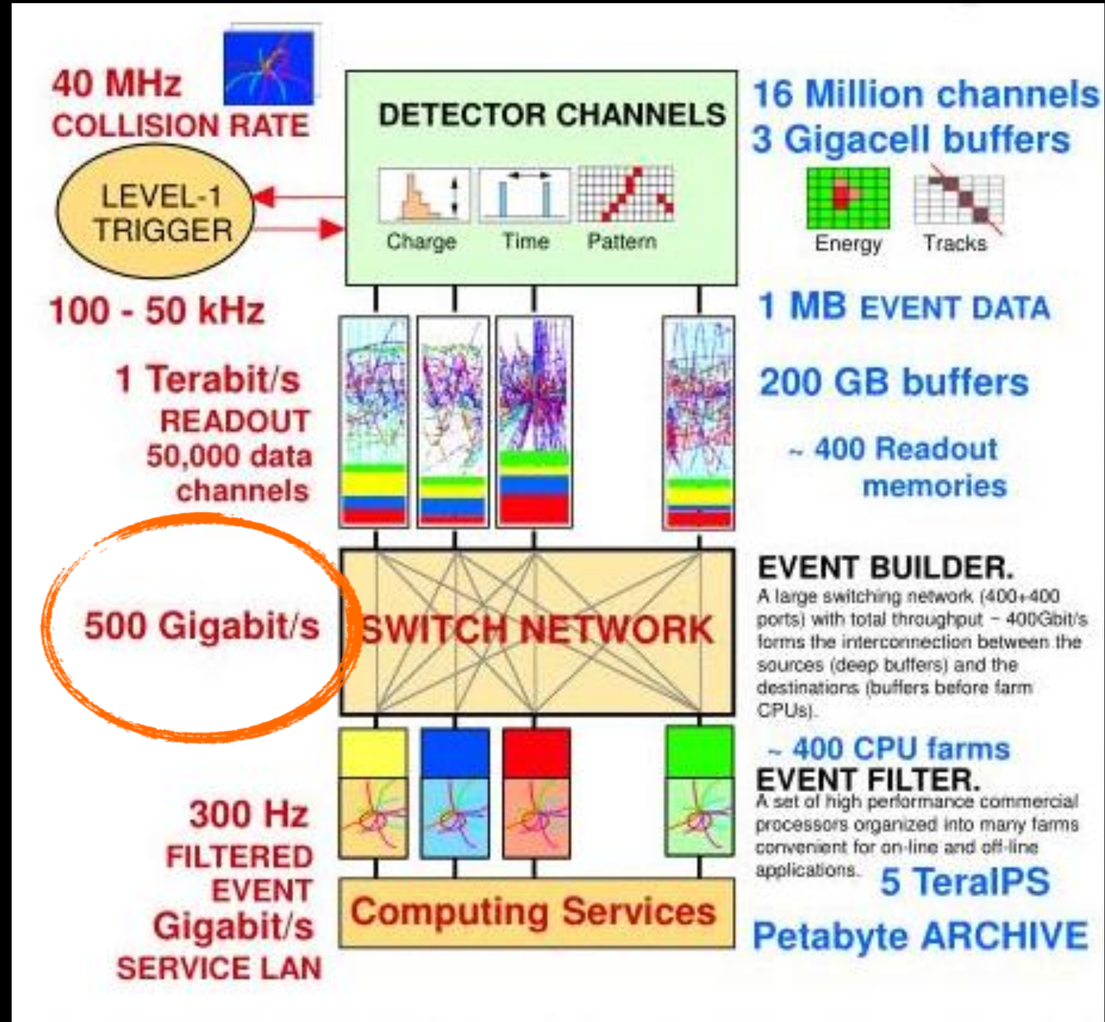


- And this is exactly what the CMS Trigger does

"Standard" figure for the CMS Trigger & DAQ

OF COURSE, "LOW ENOUGH" IS RELATIVE…



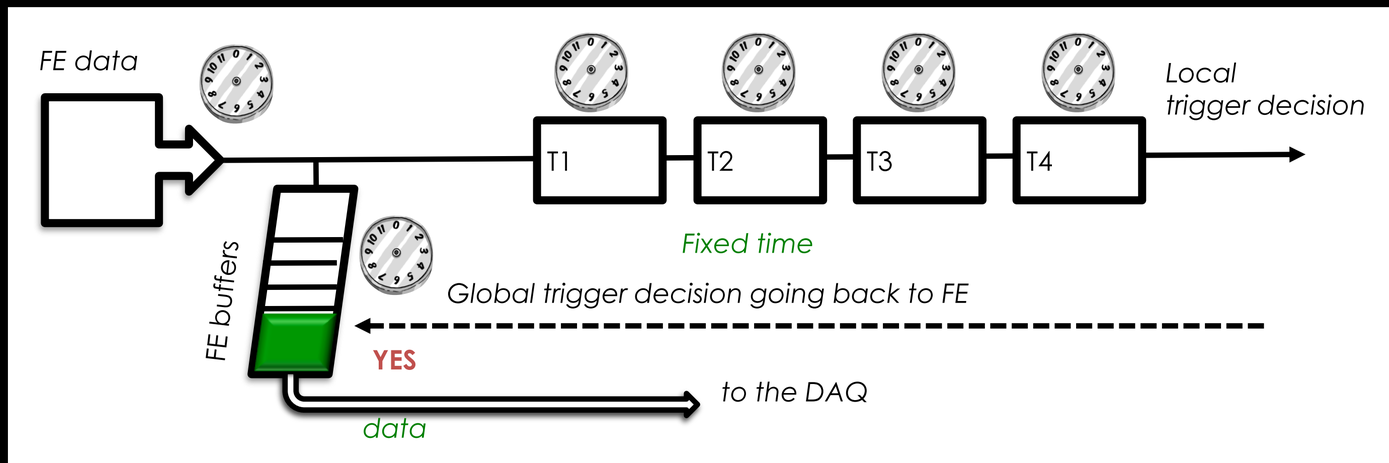


SYNCHRONOUS OR ASYNCHRONOUS?

- Synchronous: operates phase-locked with master clock
  - Data move in lockstep with the clock through the trigger chain
  - Fixed latency
  - The data, held in storage pipelines, are either sent forward or discarded
  - Used for L1 triggers in collider experiments, exploiting the accelerator bunch crossing clock

✓ **Pro's:** dead-time free (just few clock cycles to protect buffers)

✗ **Con's:** cost (high frequency stable electronics, sometimes needs to be custom made); maintain synchronicity throughout the entire system, complicated alignment procedures if the system is large (software, hardware, human…)

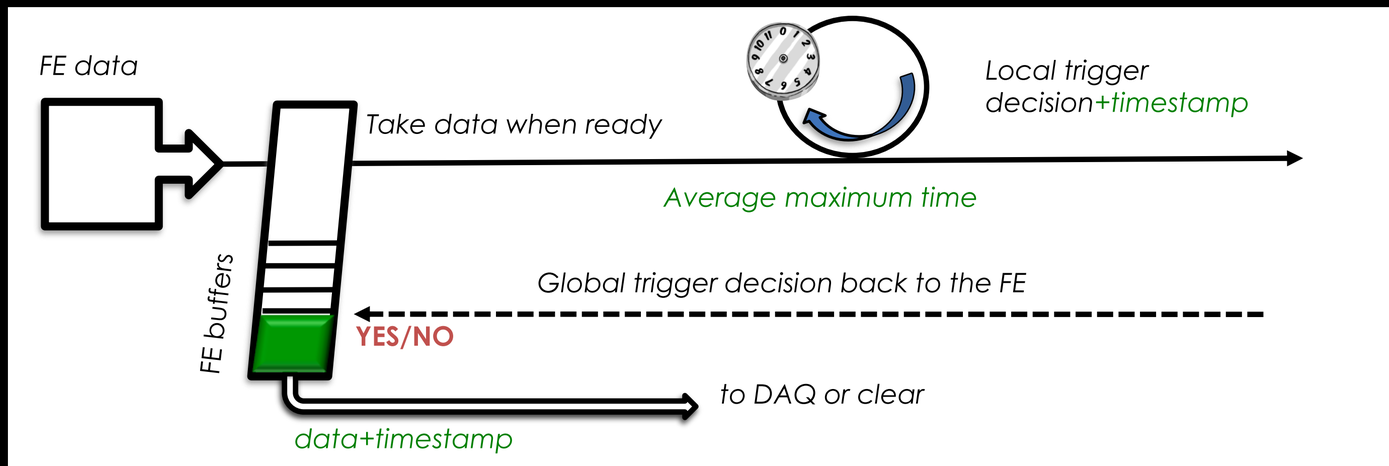


SYNCHRONOUS OR ASYNCHRONOUS?

- Asynchronous: operations start at given conditions (when data ready or last processing is finished)
  - Used for larger time windows
  - Average latency (with large buffers to absorb fluctuations)
  - If buffer size ≠ dead-time → lost events
  - Used for HLT

✓ **Pro's**: more resilient to data burst; running on conventional CPUs

✗ **Con's**: needs a timing signal synchronised to the FE to latch the data, needs time-marker stored in the data, data transfer protocol is more complex)

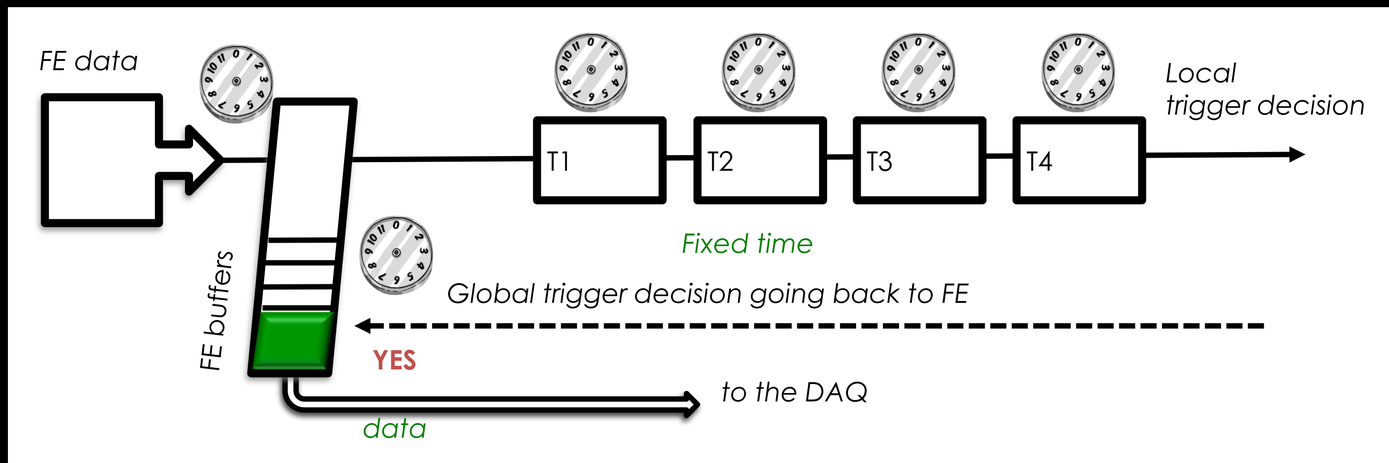


SYNCHRONOUS OR ASYNCHRONOUS? WHY NOT BOTH?

- Pseudo-synchronous: operates **locally** phase-locked
  - Data move in lockstep through the trigger chain from a set of local clocks
  - Buffering required whenever you move between clocks
  - Clocks run slightly faster than source data to prevent overflow
  - Realignment to global clock only after the final trigger stage
  - Fixed latency

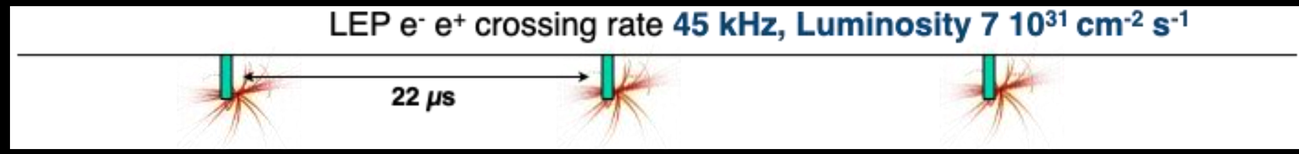
✓ **Pro's:** dead-time free (just few clock cycles to protect buffers), no need for expensive globally-distributed clock, simpler alignment procedure

✗ **Con's:** must propagate timing info with data, buffering required to handle clock-domain change



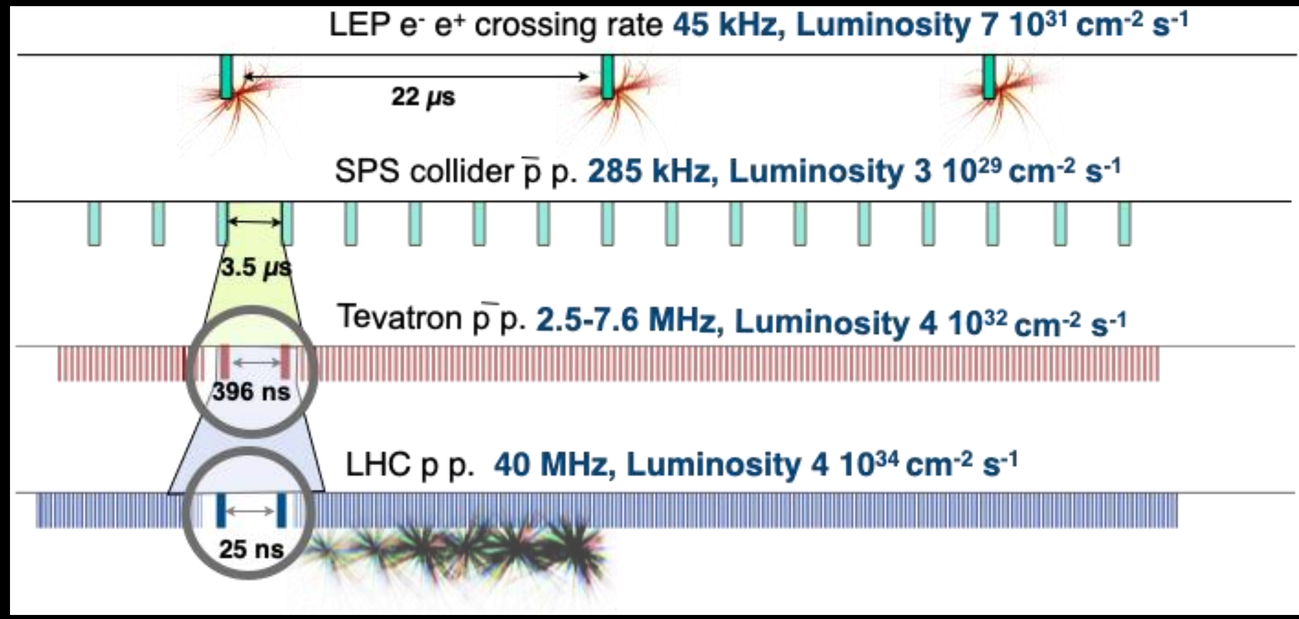
A NOTE ON TIMESCALES

- At LEP, BC interval 22 μs: complex trigger processing was possible between BXs



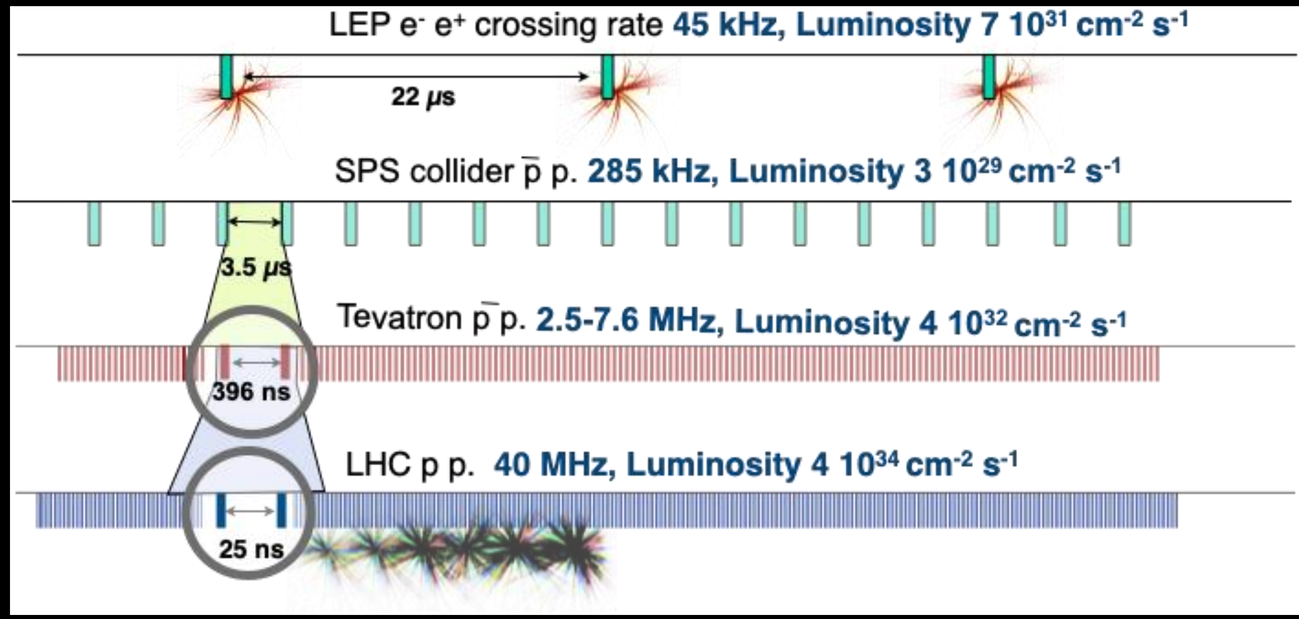
A NOTE ON TIMESCALES

- At LEP, BC interval 22 μs: complex trigger processing was possible between BXs
- Modern colliders chasing statistics
  - High Luminosity by high rate of BX
  - BX spacing too short for final trigger decision!
  - No mechanism to throttle data



A NOTE ON TIMESCALES

- At LEP, BC interval 22 μs: complex trigger processing was possible between BXs
- Modern colliders chasing statistics
  - High Luminosity by high rate of BX
  - BX spacing too short for final trigger decision!
  - No mechanism to throttle data
- Trigger logic must be pipelined



PIPELINED PROCESSING

	6pm	7pm	8pm	9pm	10pm	11pm	12pm	01am	02am	03am
										
										
										
										

PIPELINED PROCESSING

	6pm	7pm	8pm	9pm	10pm	11pm	12pm	01am	02am	03am
										
										
										
										

PIPELINED PROCESSING

	6pm	7pm	8pm	9pm	10pm	11pm	12pm	01am	02am	03am
										
										
										
										

That would just be stupid

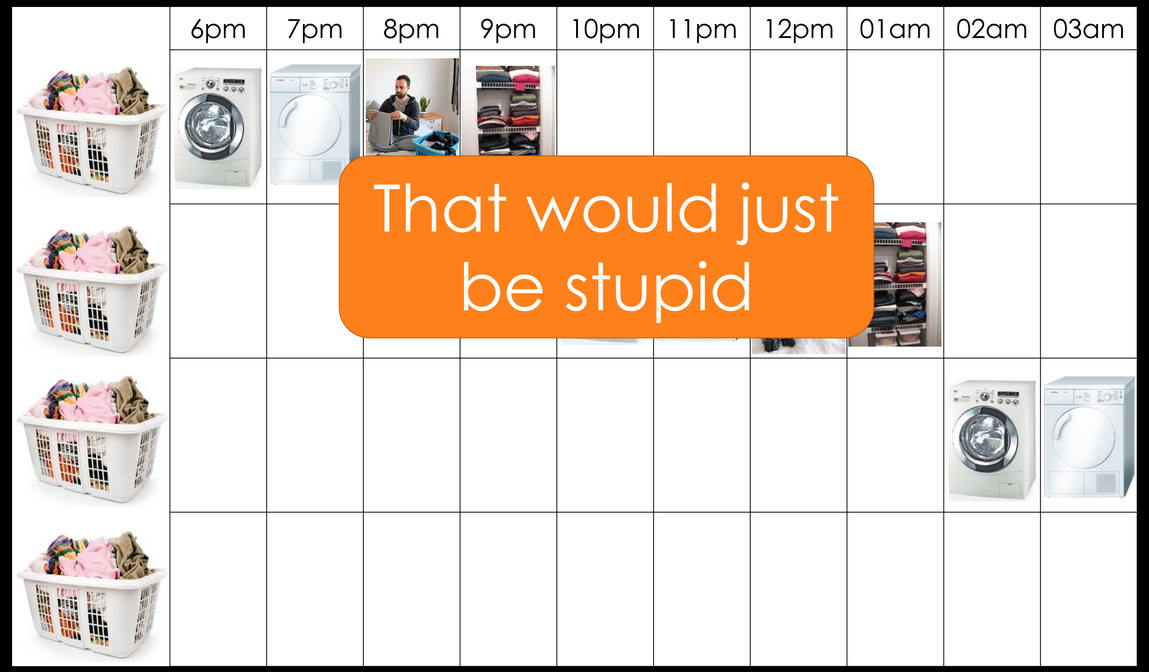
PIPELINED PROCESSING

	6pm	7pm	8pm	9pm	10pm	11pm	12pm	01am	02am	03am
										
										
										
										

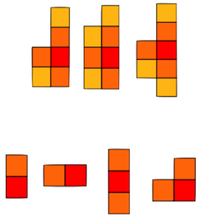
BUT THIS IS PRECISELY WHAT A CPU DOES…

- To first order, the ALU of a CPU handles one instruction at a time

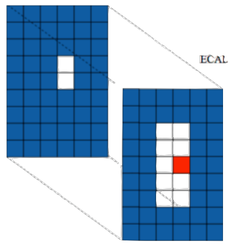
Shameless advertising for my FPGA lecture



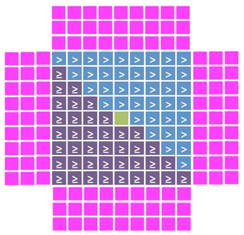
THE CMS CALORIMETER TRIGGER



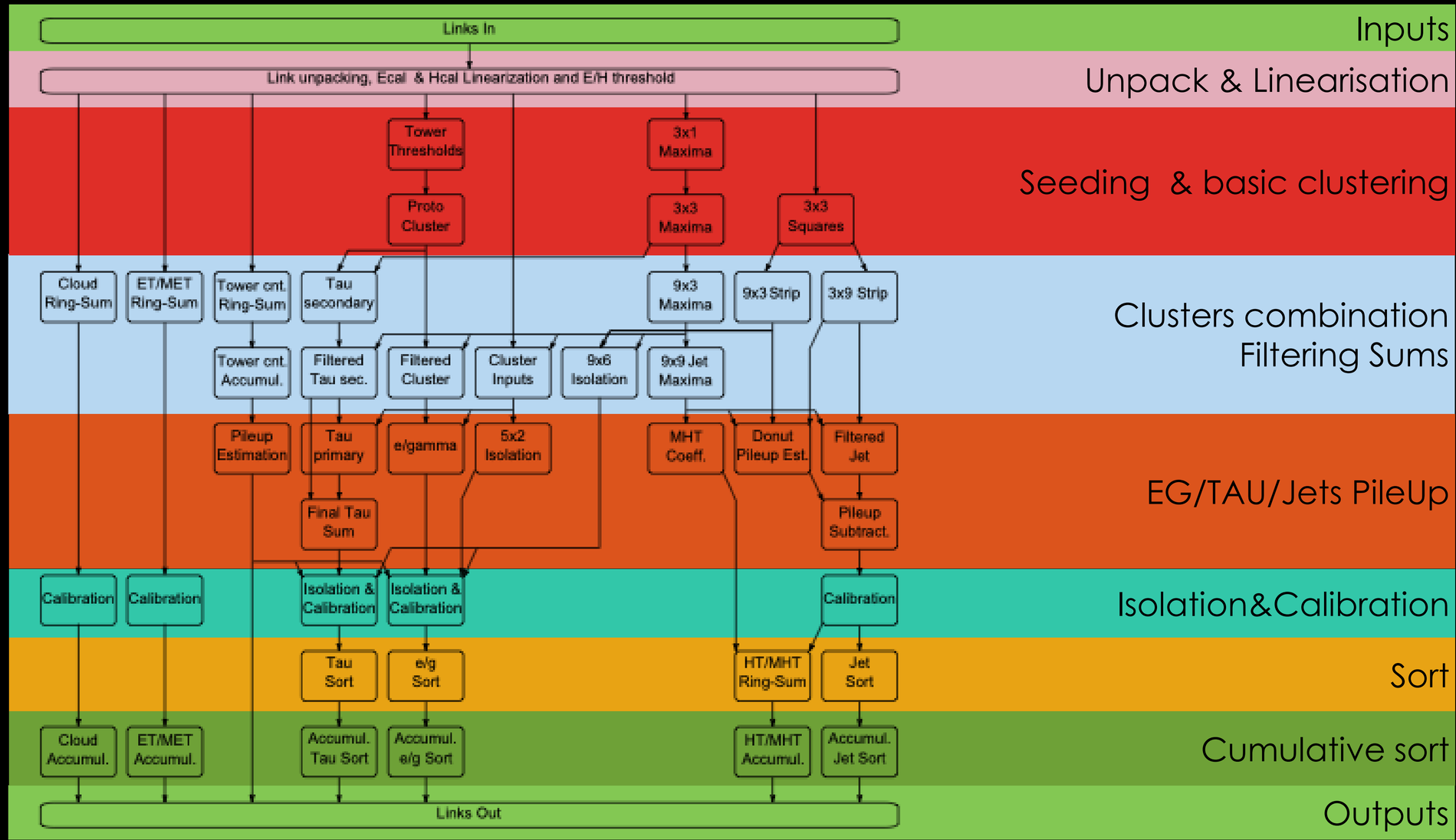
Dynamic clustering



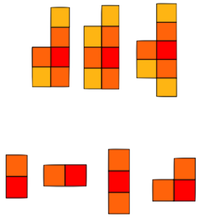
Jet building with pileup subtraction



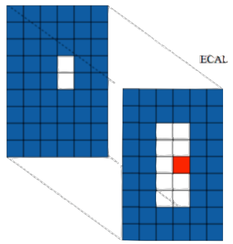
Shape veto, H/E, isolation, calibration



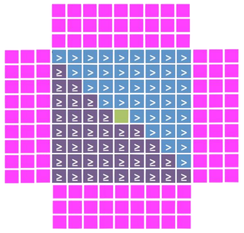
THE CMS CALORIMETER TRIGGER



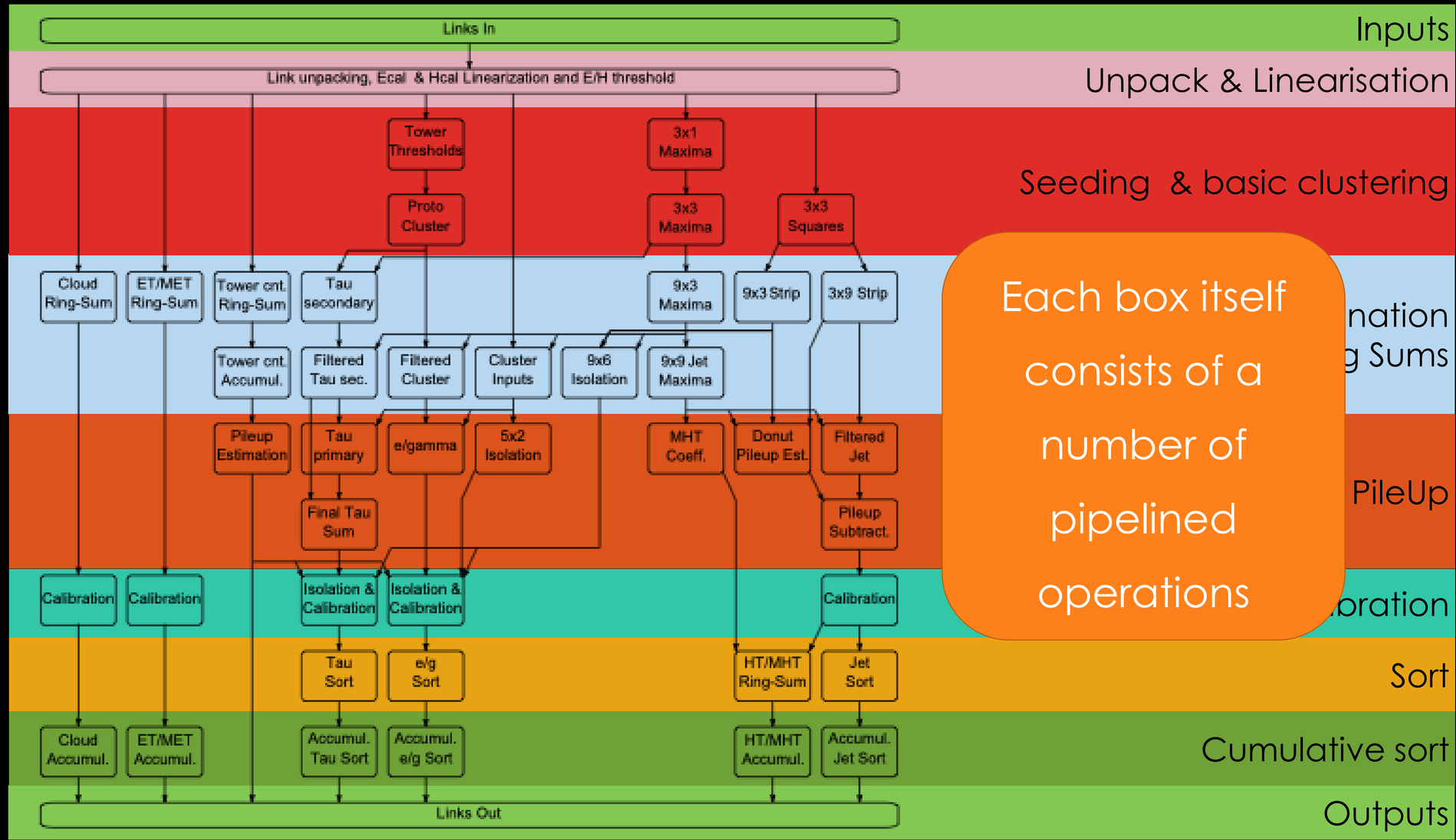
Dynamic clustering



Jet building with pileup subtraction

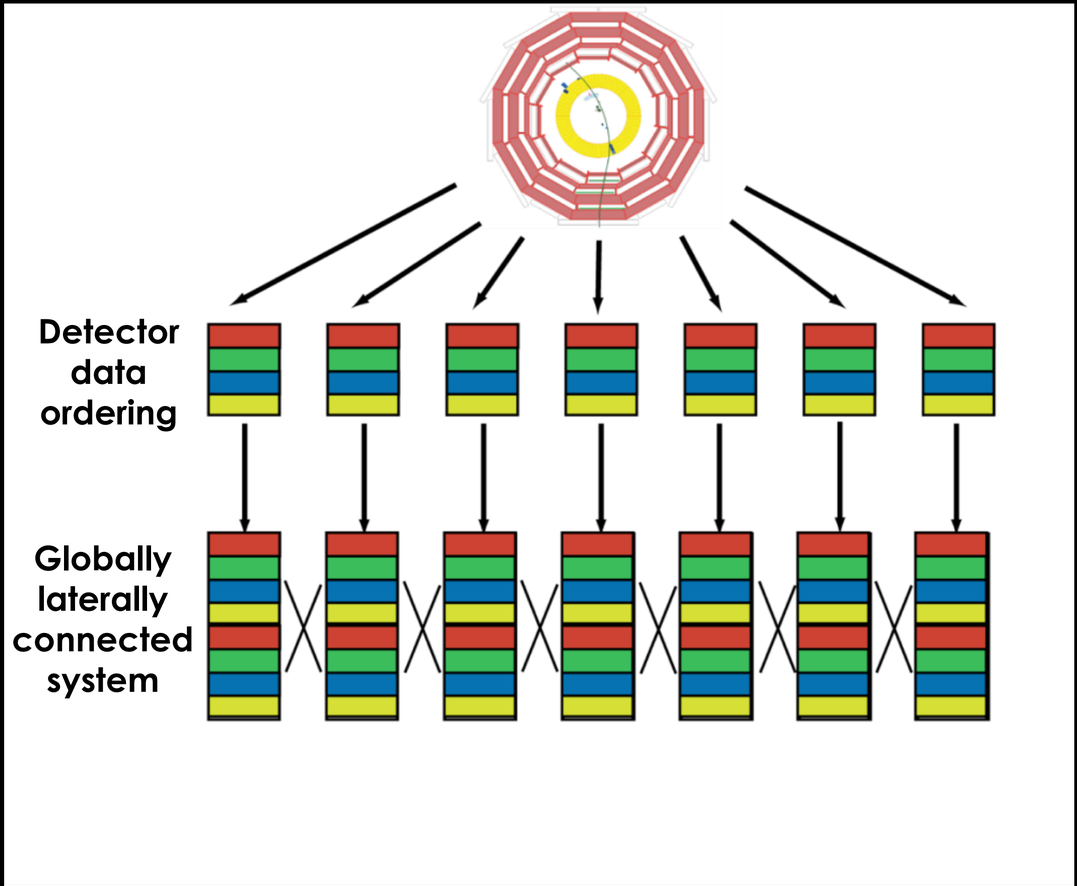


Shape veto, H/E, isolation, calibration



Each box itself consists of a number of pipelined operations

CONVENTIONAL ARCHITECTURE

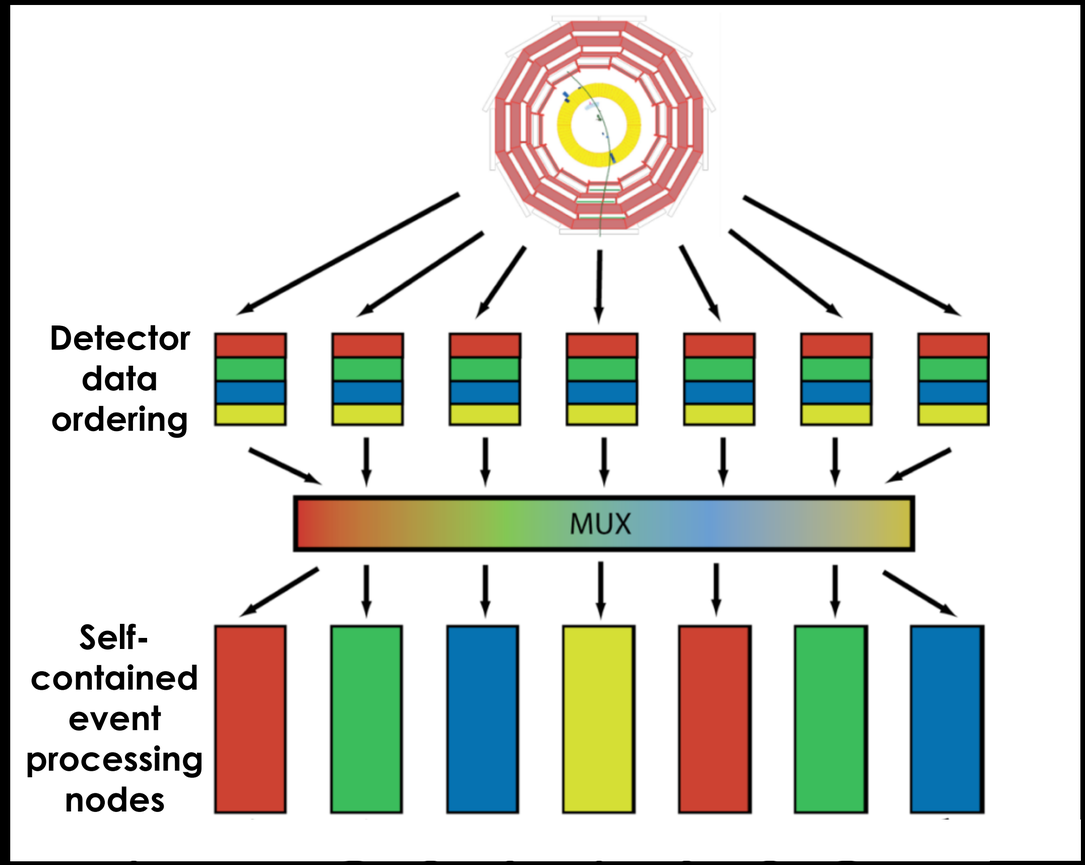


- Each subsystem is regionally segmented
- Each region must talk to its neighbour
  - This is the root cause of requiring specialized boards for a given task!
- Each region of each processing layer compresses, suppresses, summarizes or otherwise reduces its data and passes it on to the next level which is less regionally segmented

Many, many details on time-multiplexing and conventional architectures in sections 1-3 of https://cds.cern.ch/record/1421552/files/IN2011_022.pdf (although please note that the systems proposed in section 4-9 are very outdated and should be ignored)

TIME-MULTIPLEXED ARCHITECTURE

- Buffer data and stream it out optimized for processing
- Spread processing over time
  - Stream-processing rather than combinatorial-logic
  - Maximise reuse of logic resources
  - Easiest for FPGA design tools to route and meet timing
- Costs you latency, bought back by more efficient processing



Many, many details on time-multiplexing and conventional architectures in sections 1-3 of https://cds.cern.ch/record/1421552/files/IN2011_022.pdf (although please note that the systems proposed in section 4-9 are very outdated and should be ignored)

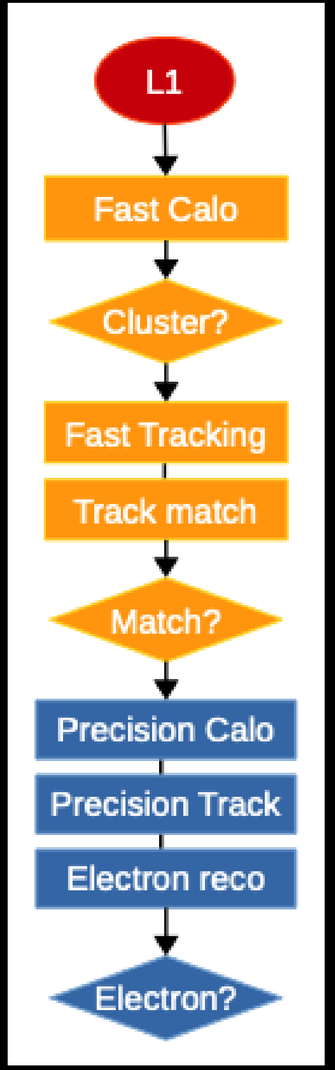
HIGH LEVEL TRIGGER ARCHITECTURE

- LEP: 40 Mbyte/s
  - VME bus sufficient for bandwidth needs
- LHC: cutting-edge processors, high-speed network interfaces, high speed optical links
- Different approaches possible
  - Network-based event building (CMS)
  - Seeded reconstruction (ATLAS)

	Levels	L1 rate	Event size	Readout bandwidth	HLT rate
LEP	2/3	1 kHz	100 kB	few 100 kB/s	~5 Hz
ATLAS	2/3	100 kHz (L2: 10 kHz)	1.5 MB	30 GB/s (Incremental Event Building)	~1 kHz
CMS	2	100 kHz	1.5 MB	100 GB/s	~1 kHz

HIGH LEVEL TRIGGER DESIGN PRINCIPLES

- Offline reconstruction too slow to be used directly
  - Takes >10s per event
  - HLT usually needs << 1s
- Instead, step-wise processing with early rejection
  - Stop processing as soon as one step fails
  - Event accepted if any of the trigger passes
  - Add a time-out to kill the Poisson tail!
- Fast reconstruction & L1-guided regional reconstruction first
- Precision reconstruction as full detector data becomes available

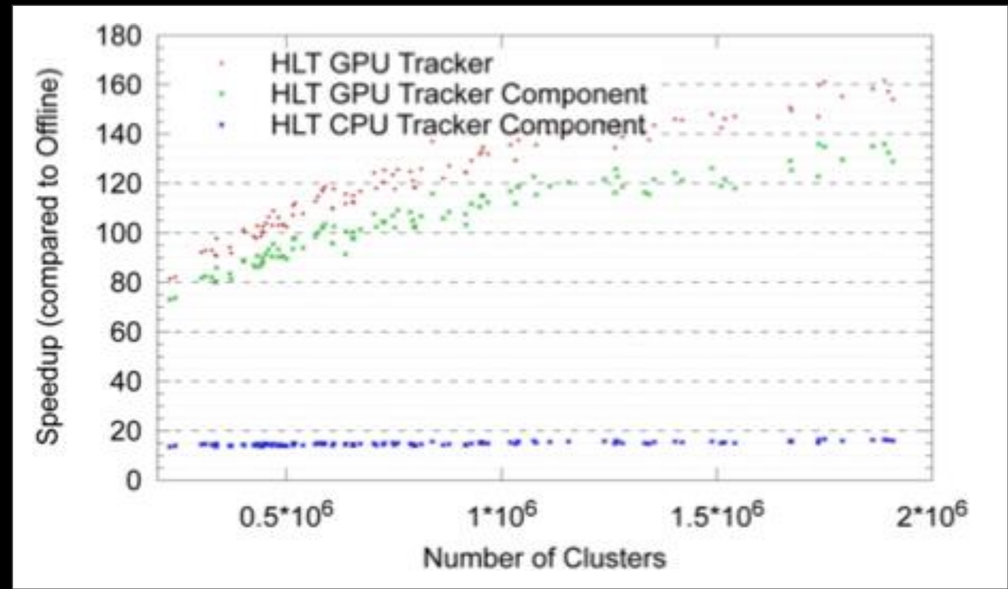


HIGH LEVEL TRIGGER DESIGN PRINCIPLES

- Event-level parallelism
  - Process more events in parallel
  - Multi-processing or/and multi-threading
- Algorithm-level parallelism
  - **GPUs** effective whenever large amount of data can be processed concurrently (although bandwidth can be a limiting factor)

- Algorithms developed and optimized offline
- Common HLT-reconstruction software framework

reduces maintenance and increases reliability

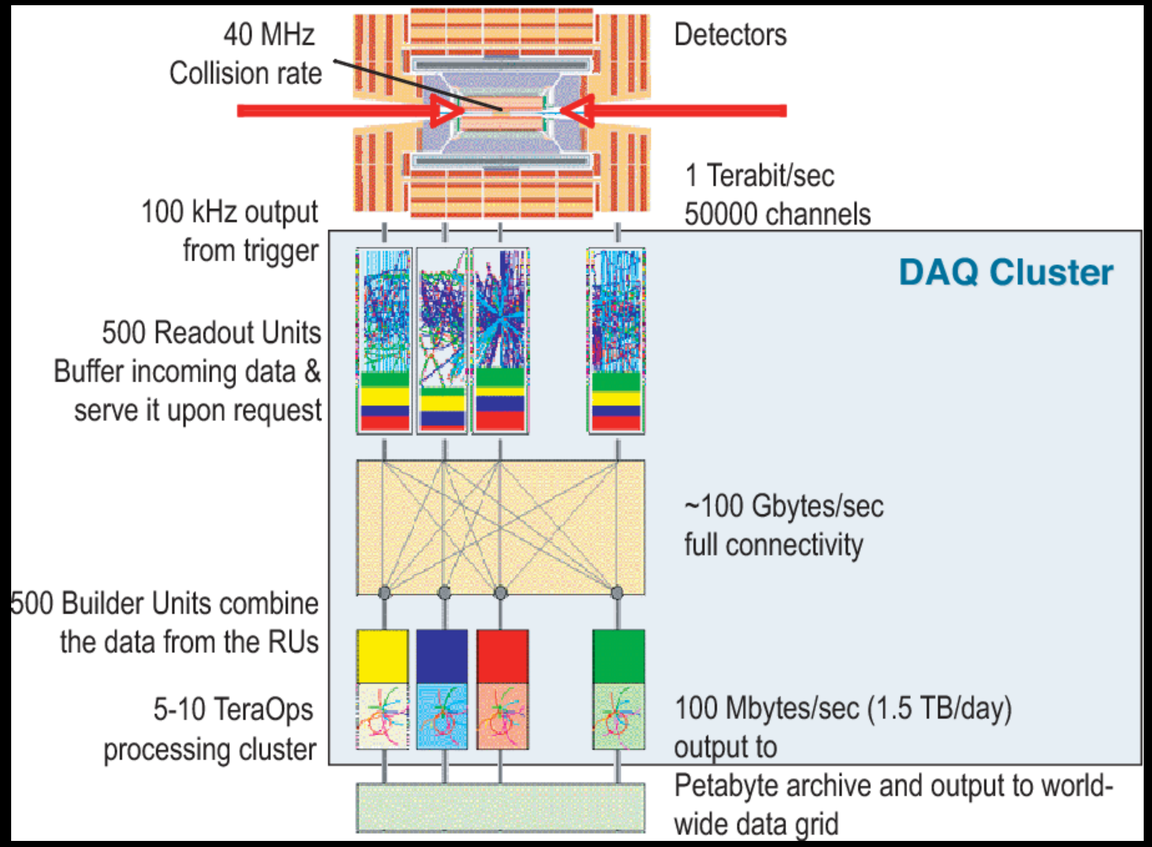


EXAMPLE: CMS HLT

- Approximately 38,000 cores
  - An equal mix of Haswell, Broadwell and Skylake
- Multithreading allowing the cores to share non-event data
  - Reduced memory footprint → can process more events: ~20% higher performance
- Upgrades to add a GPU in every filter farm node is ruled out by cost and power
  - More likely a dedicated server sub-farm which does heavy tasks on demand
  - FPGAs acceleration also a (possibly better) option
- Boundary between trigger and DAQ is fuzzy, they are closely related
  - At CMS the “High Level Trigger” is part of the DAQ

CMS - EVENT BUILDING

- At the detector readout, data is fragmented
  - Readout PCs access data from some local detector region
  - Each PC buffers data from multiple events
- Software triggering & storage need all data for one event
- High-throughput network to reorganize data
  - Using standard networking technology as much as possible

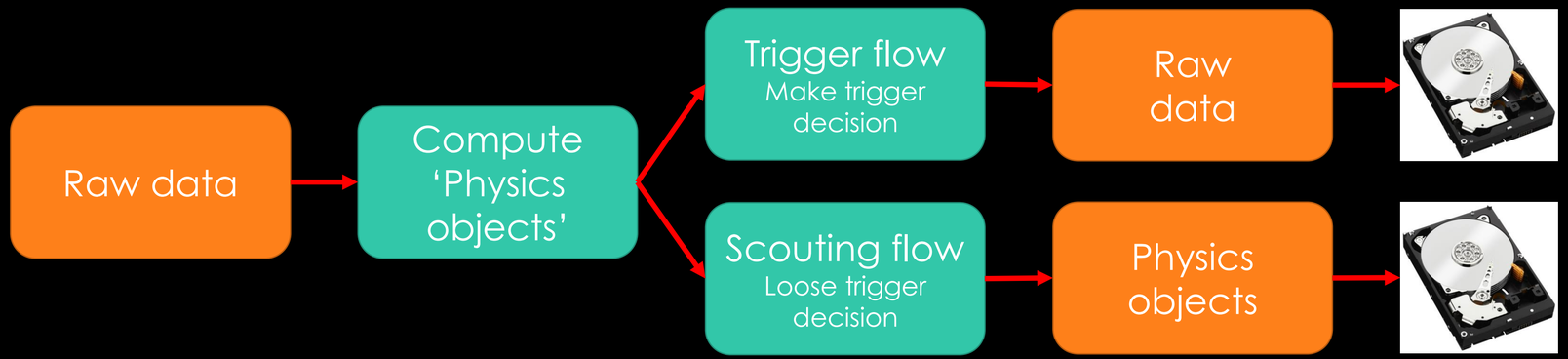


Absolute numbers here are out of date!

J. Gutleber, Data Acquisition in High Energy Physics

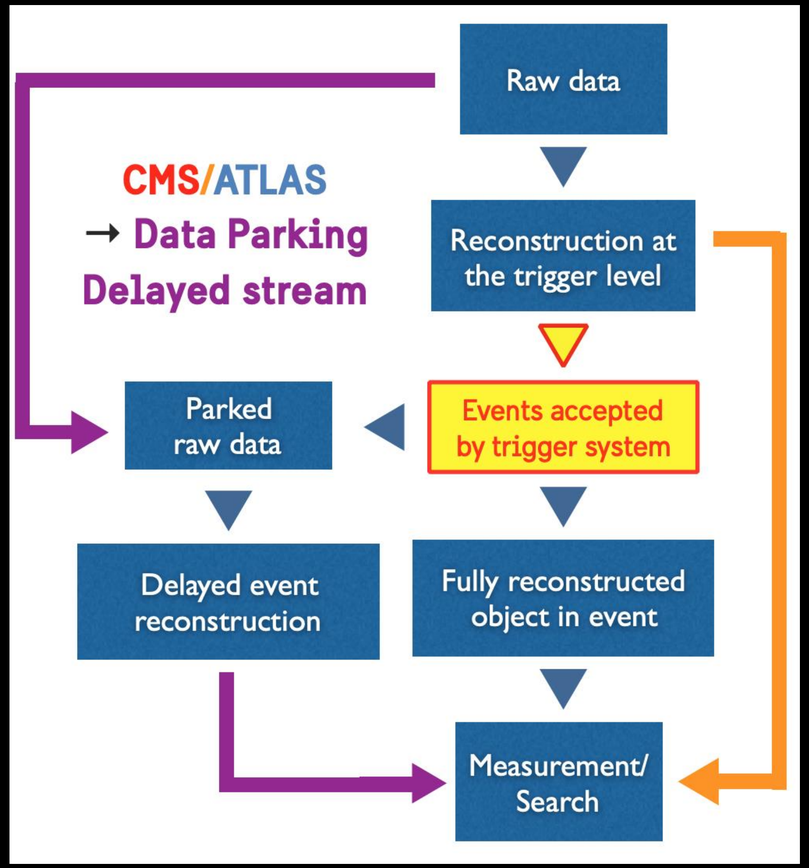
REAL-TIME ANALYSIS / SCOUTING

- We have discussed the typical trigger & DAQ paradigm
  - Fast & coarse processing of raw data -> decide what events to keep -> store raw event data
- In CMS we have "scouting" - today at HLT, at L1T also for Phase 2
  - Same concepts exist at LHCb (Turbo Stream) and ATLAS (Trigger-object-level analysis)
- Store objects computed by the trigger (L1T or HLT) for *more events* for later analysis
  - More events, smaller event content (don't keep raw detector data)



DATA PARKING

- Based on the fact that HLT trigger rate was a bit lower than what the DAQ could handle
- Add some new, loose, trigger paths for specific analyses
- ‘Park’ the raw data -
  - Don’t run full reconstruction on accepted events immediately, store the raw data
  - Process later when no triggers are arriving - e.g. in between runs
- CMS, LHCb, ATLAS all use this

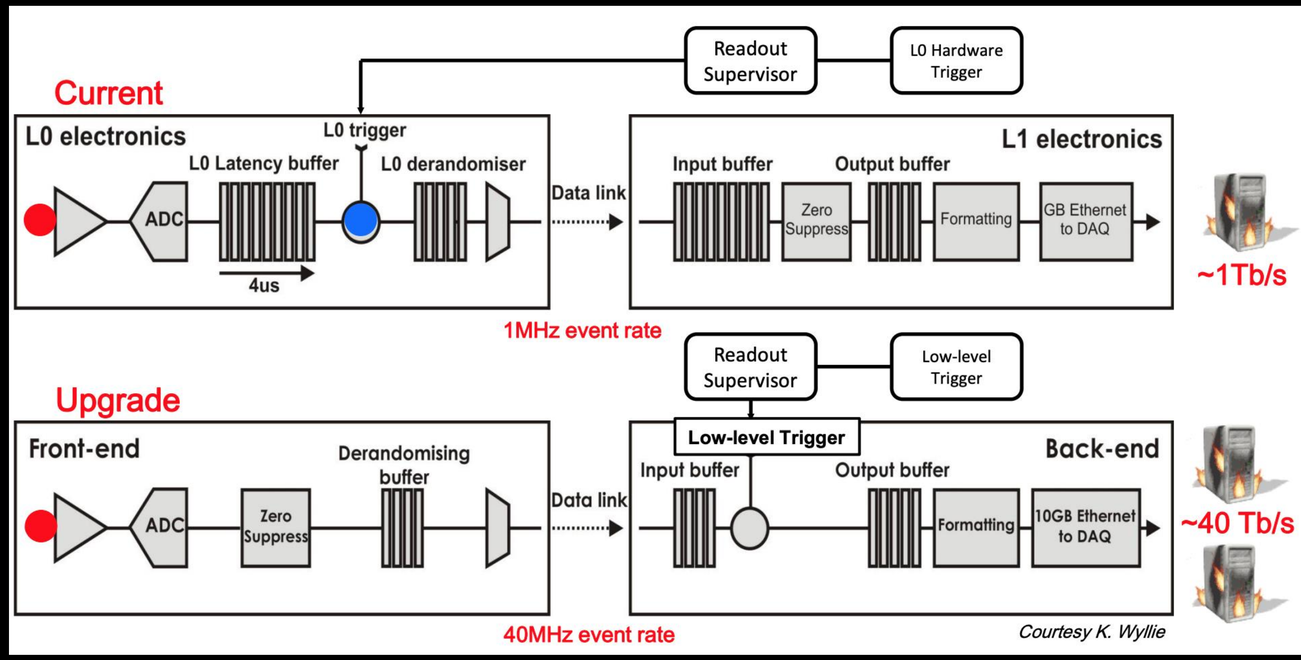


Right orange arrow is scouting

Talk on ‘real time analysis’ - C. Doglioni

THE FUTURE: TRIGGERLESS READOUT?

- LHCb started with a hardware trigger
- Then decided they could get rid of that step as L0 trigger was introducing bias
- Back-end electronics and software filter see 40x higher rate



DAQ MINI-SUMMARY

- DAQ should aim to minimize dead time and keep up with incoming rate
- Many choices when designing DAQ
  - e.g. zero-suppression on or off detector? Simple front-end with high output rate, or complicated front-end with lower output rate?
- Modern experiments are large detectors with many channels
  - DAQ systems are complicated
  - Many strategies for enhancing existing DAQ strategies - scouting, parking, etc.
- Brute-force computing power can be the simplest and “cleanest” strategy

TRIGGERING: PRACTICAL ADVICE

- You might well have to design a trigger for some physics channel you are interested in
- Not as unusual as you might imagine!
- Some things to remember....

TRIGGERING: PRACTICAL ADVICE

- Keep it as simple as possible
  - Easy to commission
  - Easy to debug
  - Easy to understand

TRIGGERING: PRACTICAL ADVICE

- Be as inclusive as possible
  - One trigger for several similar analyses
  - Your trigger should be able to discover the unexpected as well as the signal you intended it for!

TRIGGERING: PRACTICAL ADVICE

- Make sure your trigger is robust
  - Triggers run tens of millions of times a second so ANY STRANGE CONDITION WILL OCCUR, make sure you are prepared for it
  - Detectors don't work perfectly EVER! Make sure your trigger is immune to detector problems
  - Beam conditions change - be prepared

TRIGGERING: PRACTICAL ADVICE

- Build in redundancy
  - Make sure your signal can be selected by more than one trigger
  - Helps to understand biases and measure efficiencies
  - Also for safety, if rates are too high or there's some problem you still get your events

TRIGGERING: PRACTICAL ADVICE

- Finally…Taking your signal events is only part of the game
  - You might well also need background samples
  - You will need to measure the efficiency of your trigger using a redundant trigger path
  - You will need to know if it works! Monitoring!

TRIGGERING: PRACTICAL ADVICE

- And remember…

The goal is not to perform the analysis online – it is just to get the events written to tape at a manageable rate

TRIGGERING: CONCLUSION

- Triggers are not new
  - but they are constantly evolving as the accelerators and detectors do
- The design of how you structure the transfer of data around your system is the most important decision you will make
- Heterogeneous computing farms look likely to feature at HL-LHC
  - but it is a brave new world!

TRIGGERING: CONCLUSION

- Triggers are not new
  - but they
- The design
  s the most important decision you will make
- Heterogeneous computing farms look likely to feature at HL-LHC
  - but it is a brave new world!

Oh, and be very suspicious if your supervisor plies you with strong coffee and gets you to look for scintillation light

THANK YOU

Any questions?

Another shameless advert for my FPGA lecture on Friday!