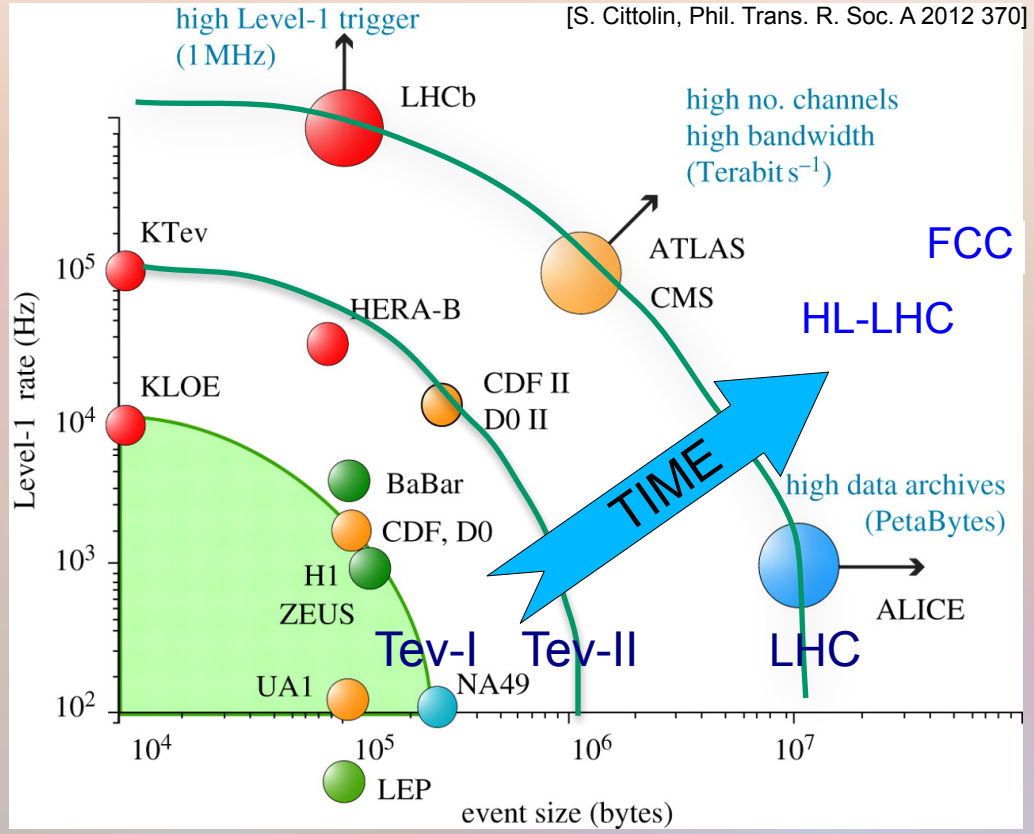
Real-time reconstruction of pixel vertex detectors with FPGAs

Giovanni Punzi
for the LHCb RTA Project
University & INFN-Pisa

VERTEX 2019
Lopud, October 13-18, 2019

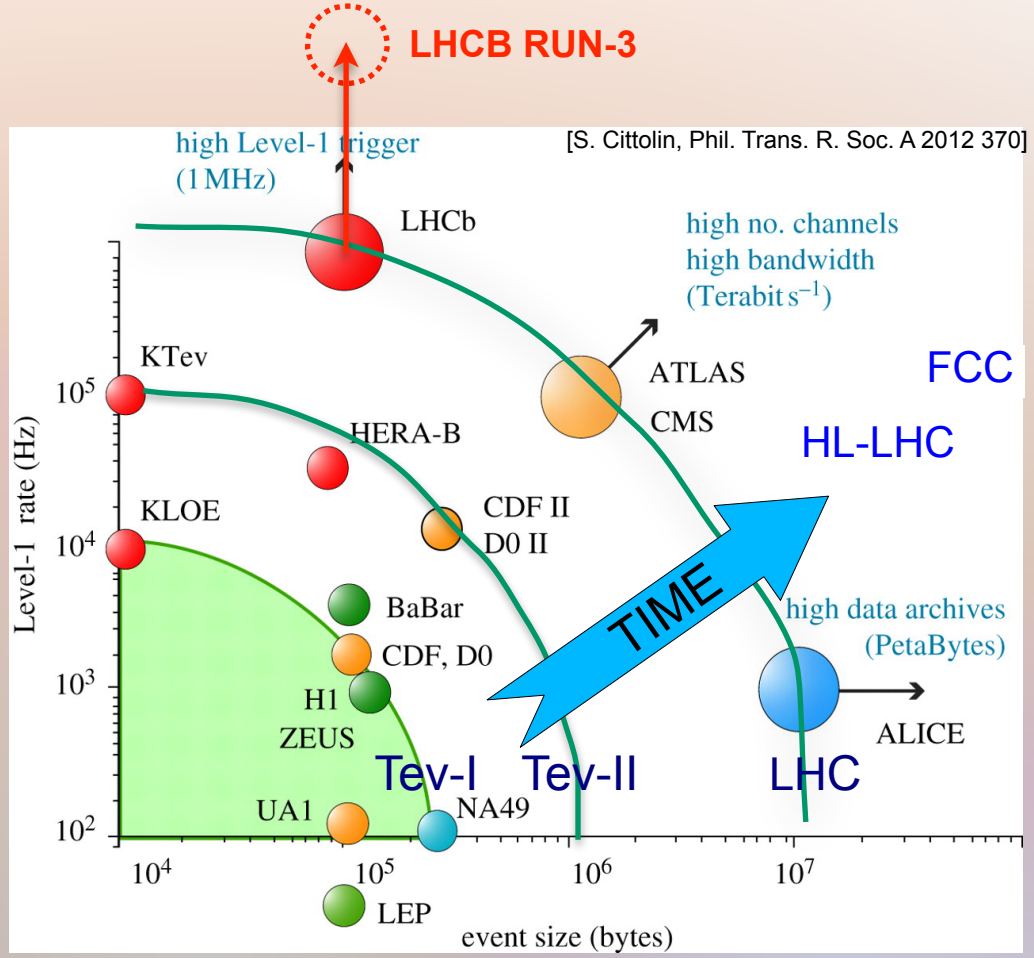
Introduction

- History of HEP data processing has seen a steady increase in sophistication and computing load
- This was greatly helped by strong progress in consumer CPU, providing flexible power at low cost
- With Moore's law **slowing down,** and Physics landscape asking for more ***precision measurements,*** the need is increasing for more effective, specialized solutions.
- Most future high-intensity programs face this challenge, and so does LHCb



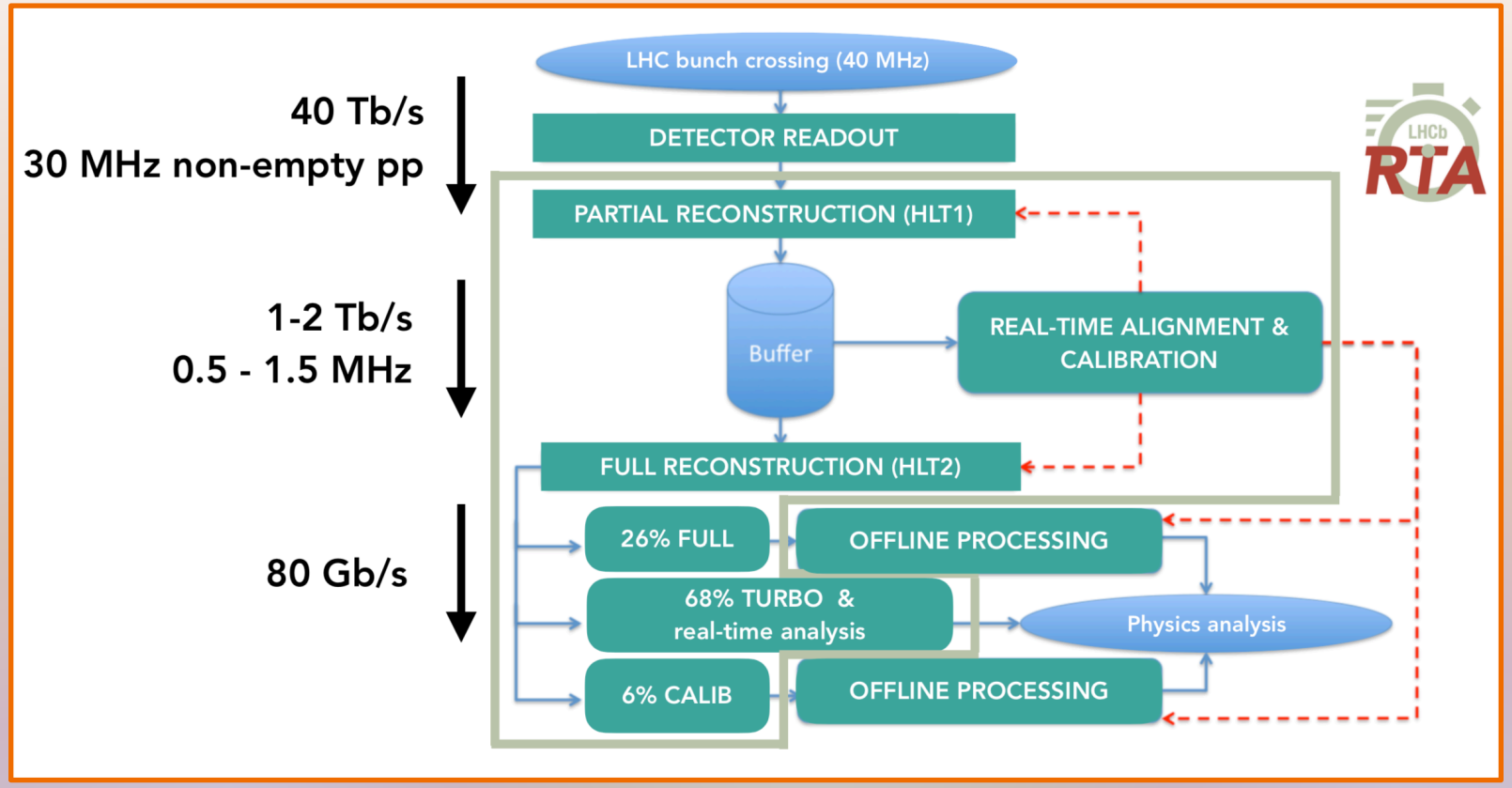
Introduction

- History of HEP data processing has seen a steady increase in sophistication and computing load
- This was greatly helped by strong progress in consumer CPU, providing flexible power at low cost
- With Moore's law **slowing down,** and Physics landscape asking for more ***precision measurements,*** the need is increasing for more effective, specialized solutions.
- Most future high-intensity programs face this challenge, and so does LHCb



- *LHCb, with its large rate of signal events, small but asking for complex analysis, is facing significant challenges already from the upcoming Run-3:* ***running at 40 MHz***

The Upgraded LHCb DAQ model



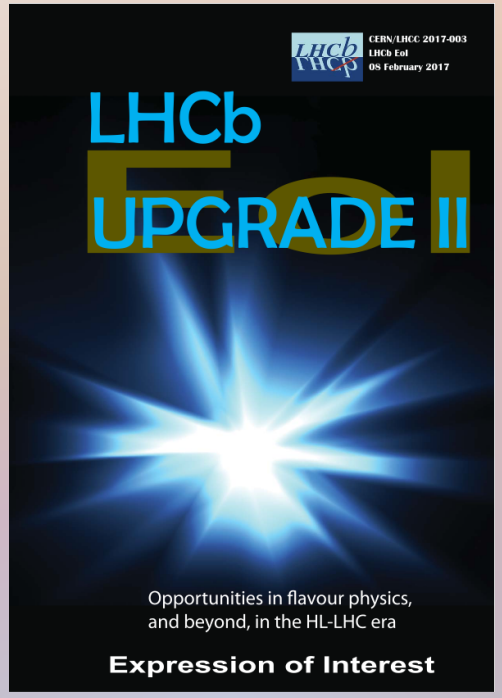
For Run 3, LHCb adopted a **Real-Time Analysis** approach to Data Processing:

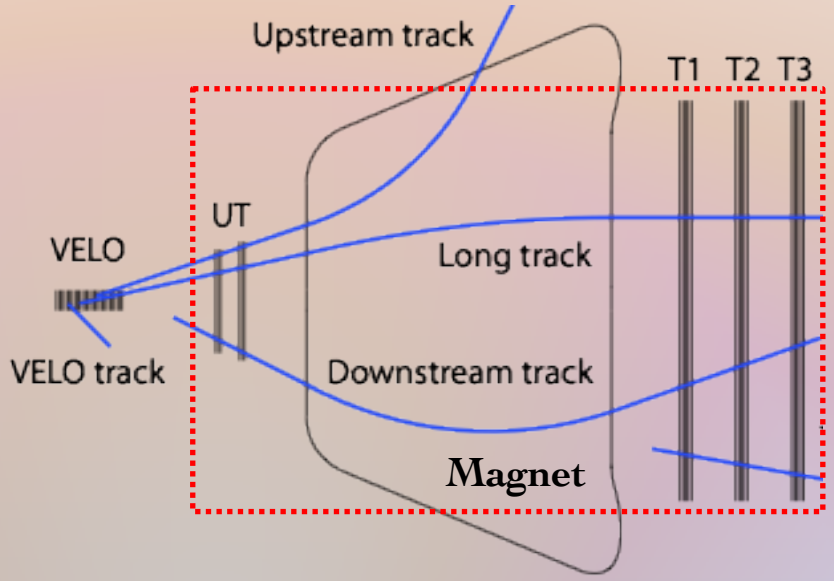
- Large rate of fully reconstructed events written on permanent storage - but with reduced per-event information ("Turbo"). Most is signal. LHCb got *billions* of HF *in Run2 already*
- Requires large amounts of online computing power, split in two-stages, with a large buffer in-between to hold data during calibration/alignment. No offline re-reconstruction.

- Significant challenge already at LHCb target luminosity in Run-3 of $L=2*10^{33}$
- (at Pt_{min}~200 MeV, the track rate is comparable to Pt>2GeV in a central detector $L=5*10^{34}$

A view of the future: LHCb Phase-2 upgrade

- LHCb is planning a Phase-II at L=2*10^{34} to fully exploit the HL-LHC for the purpose of flavor physics [CERN/LHCC 2017-003 (8/2/2017)]
- The EOI (LHCC recommended to turn into TDR) mentions studies towards realization of specialized FPGA processors for track reconstruction



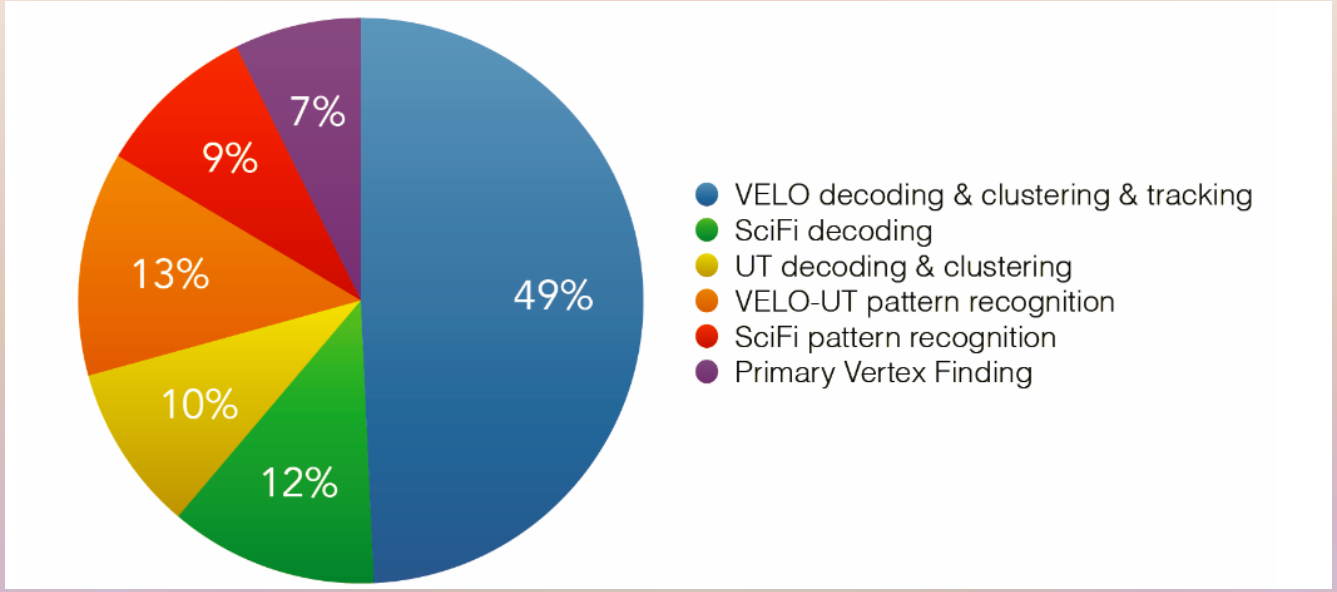


- Possible application: triggering of Long-Lived Particles from *Downstream* tracks (challenging to reconstruct without a specialized processor) [LHCb-II workshop][ACAT-19]

<u>This Talk:</u> Pilot project of a Silicon Pixel Vertex tracker for Run-3

Motivation

- Doing HLT1 at 30 MHz is not an easy business already at $L=2*10^{33}$



HLT1 CPU-time usage (~**35 MHz** total throughput on a **1000-server Farm**)

- VELO is ~10% of data but its reconstruction is ~1/2 of total HLT1 time budget
- Relatively small b/w: can be handled by a modest FPGA system
- Save the flexibility of CPUs for more intelligent jobs if successful.
- A good first test-case for future, larger-scale applications

Ingredients for tracking @30 MHz

With a highly-parallel data-flow architecture, very fast tracking is possible: [NIM A453(2000),425]

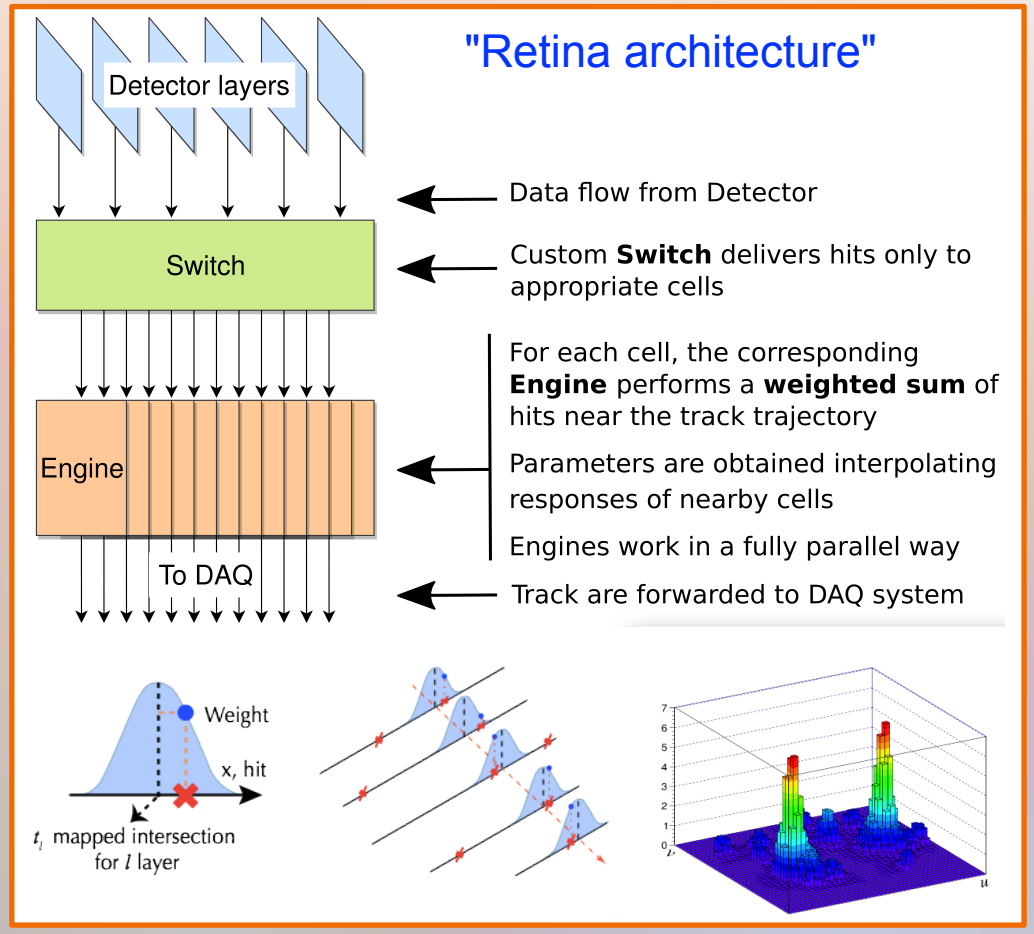
Ingredients:

1) large-bandwidth, fast custom switch (perform similar function to a "*Hough-transform*")

2) an array of cellular processors (engines) working in parallel to perform a vision-like neural algorithm

For a VELO-pixel detector, need to add a **parallel 2D cluster-finder** beforehand

Everything programmed in FPGA at low-level to achieve maximum performance



INFN R&D project ("RETINA") demonstrated feasibility of 30 MHz tracking already with previous-generation FPGAs [PoS(TWEPP-17) 136]. *Application in a real experiment is however a different thing.*

Method of Integration in LHCb DAQ

Plan is to integrate at **Pre-Build** level.

(effectively part of the readout)

Challenges:

Sustain **30 MHz event rate w/o buffering or time-multiplexing**

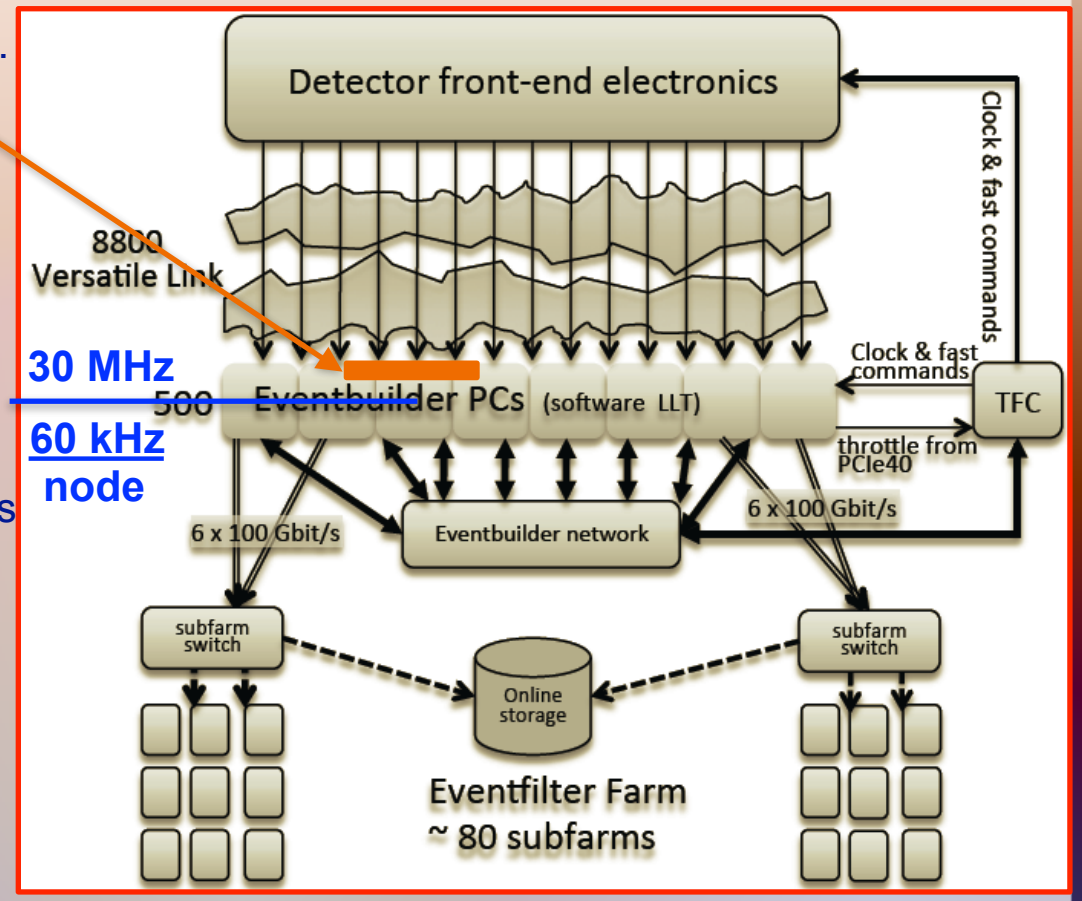
Advantages:

Modularity: use just the number of cards needed for the detector of interest

Unpacking the event not necessary

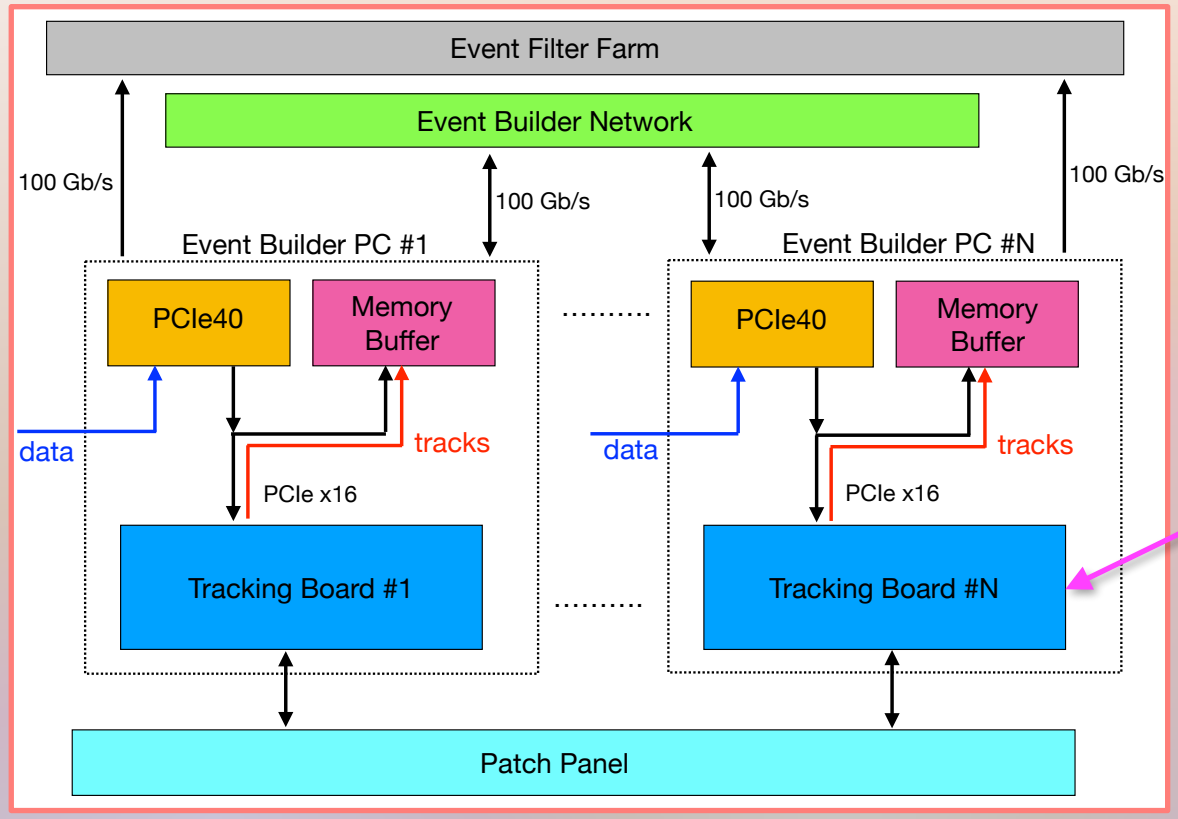
Can reduce data flow INTO the EB

Transparent: Appears as "virtual detector" producing ready-made tracks.



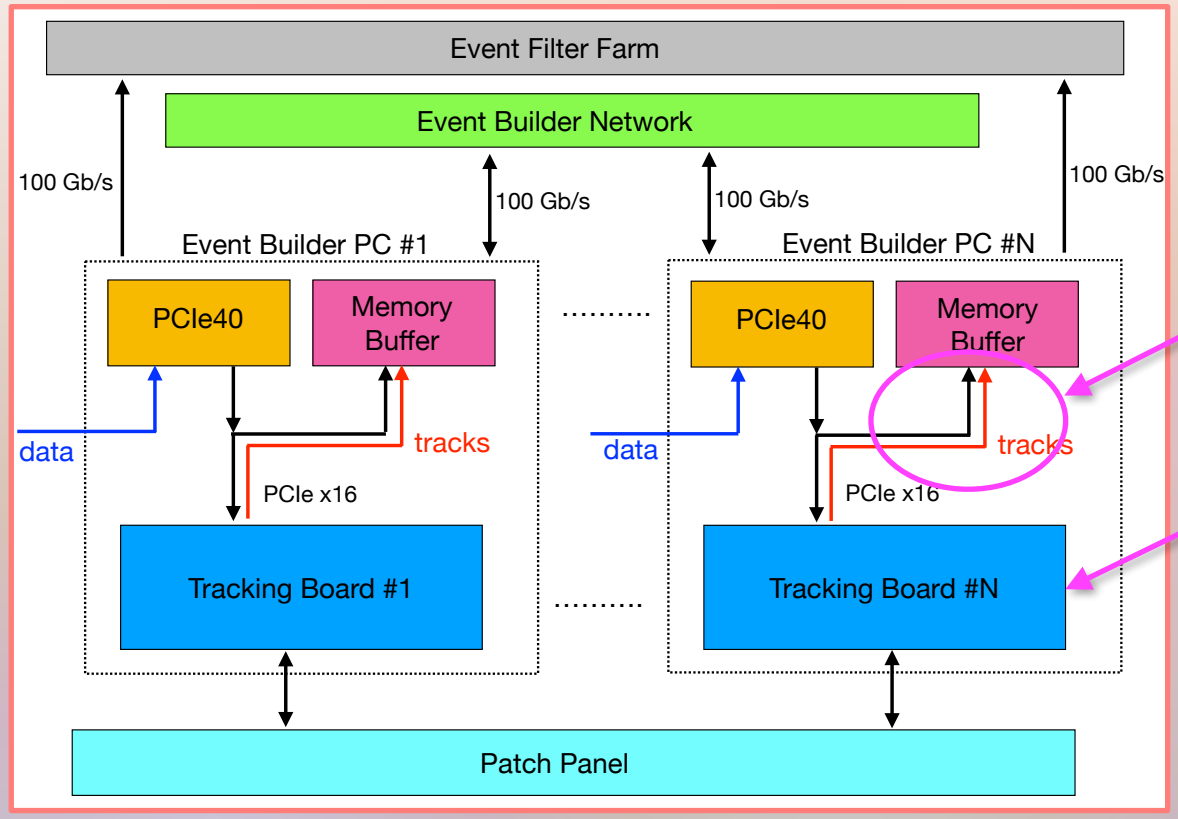
[LHCB-TDR-016(2014)]

Implementation as array of COTS cards



Match every readout card (PCIe40) with a tracking card

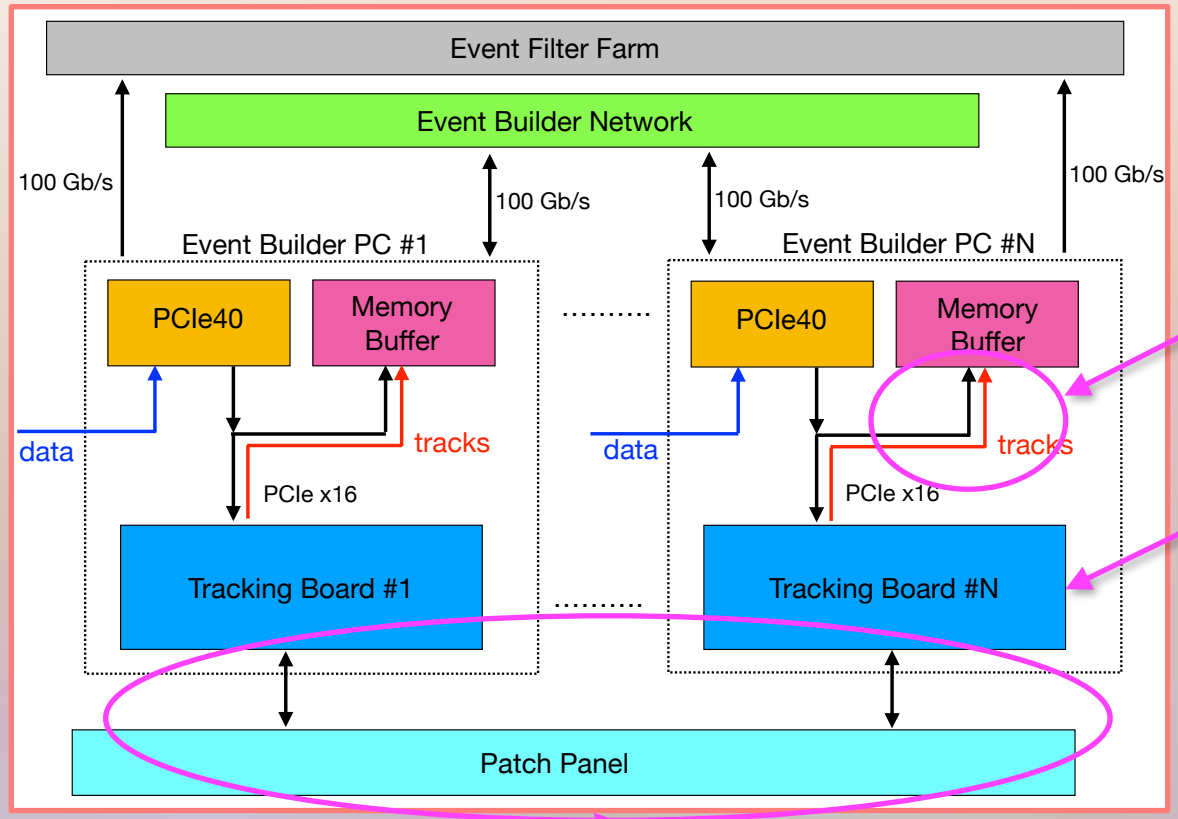
Implementation as array of COTS cards



Tracking cards ADD extra "RAW Data" to event

Match every readout card (PCIe40) with a tracking card

Implementation as array of COTS cards



Tracking cards ADD extra "RAW Data" to event

Match every readout card (PCIe40) with a tracking card

- Tracking cards communicate via fast network (akin to Microsoft's *CATAPULT)*
- Very low-latency (<1μs), makes all boards appear as a single device to the EB
- Bi-directional optical links,10÷28Gb x16

Analogy to Microsoft's CATAPULT architecture

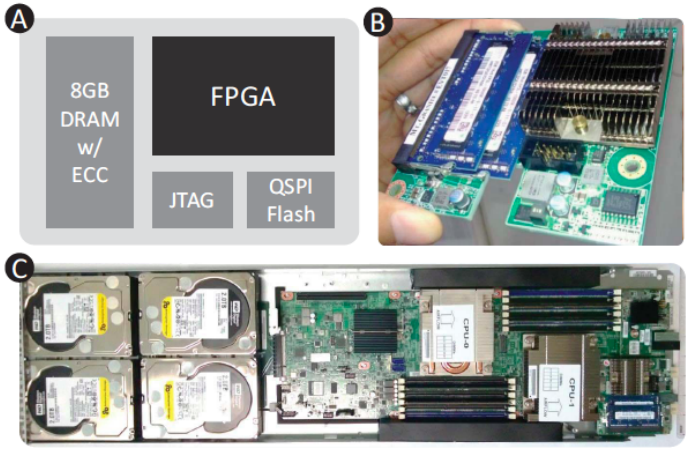


Figure 1: (a) A block diagram of the FPGA board. (b) A picture of the manufactured board. (c) A diagram of the 1 U, half-width server that hosts the FPGA board. The air flows from the left to the right, leaving the FPGA in the exhaust of both CPUs.

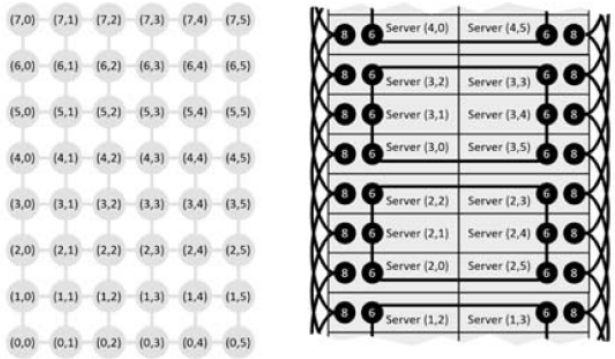


Figure 2: The logical mapping of the torus network, and the physical wiring on a pod of 2 x 24 servers.

Proceeding of the 41st ISCA - ISBN 978-1-4799-4394-4 p. 13-24

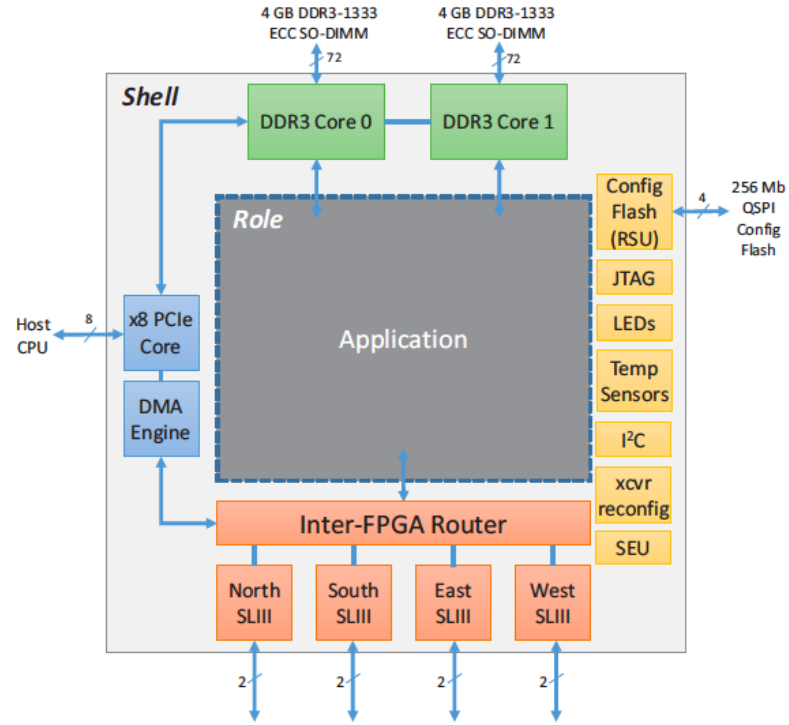


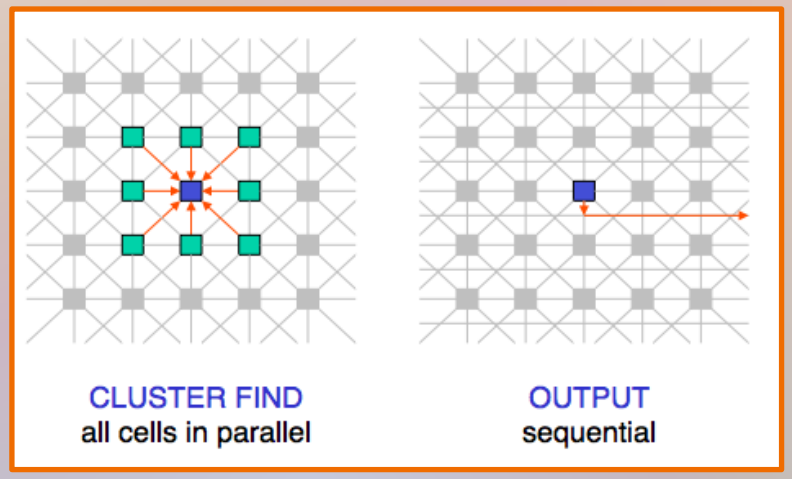
Figure 3: Components of the Shell Architecture.

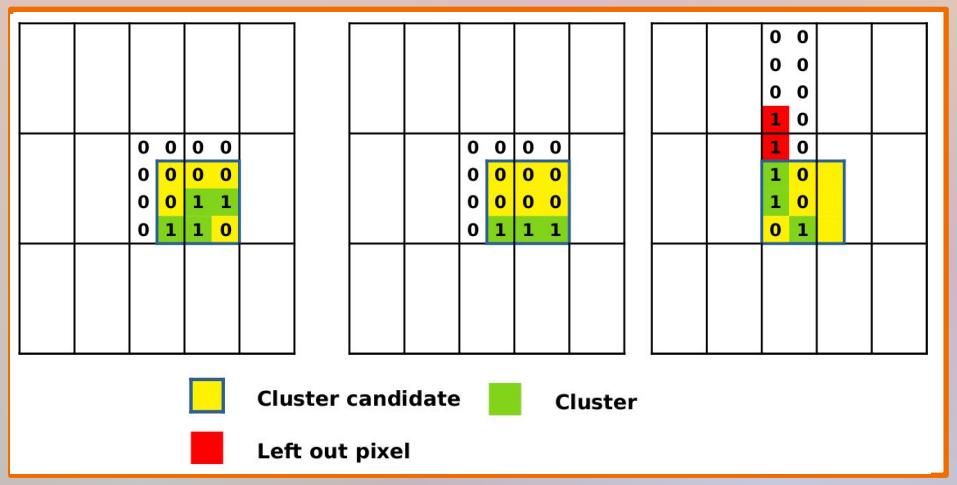
- Interesting structural analogy with independently-developed 'Catapult' system
- Distributed, inter-connected FPGA boards (powering *Bing* in clouds)
- Larger latencies, but similar issues

Step1: Cluster finding

- VELO data read out as 2x4 pixel arrays ("SuperPixels")
- Designed parallel FPGA algorithm for finding clusters of ***on*** pixels
- A matrix of active, interconnected elements finds meaningful clusters

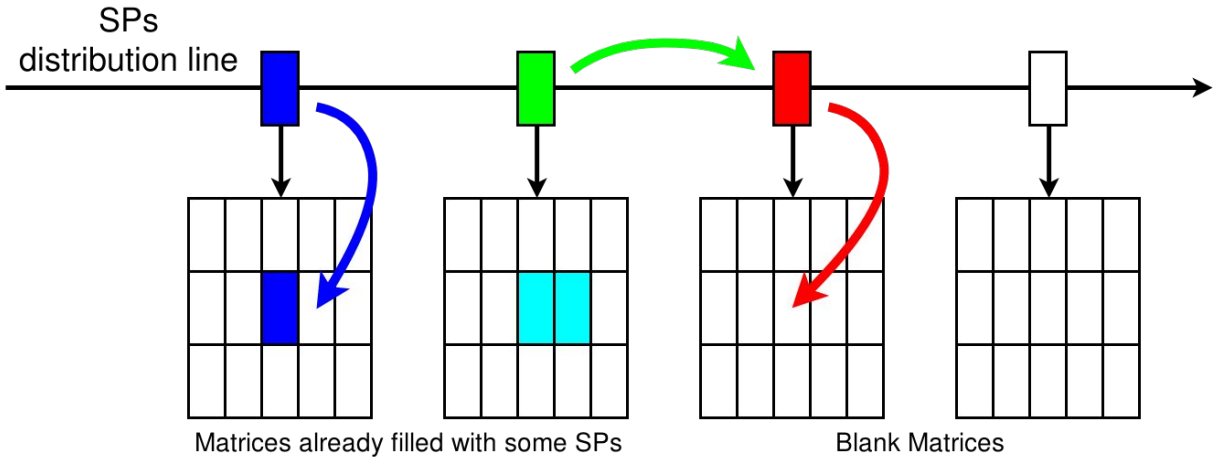
  (same principle of 'retina' algorithm for track finding)





Data distribution for Clustering

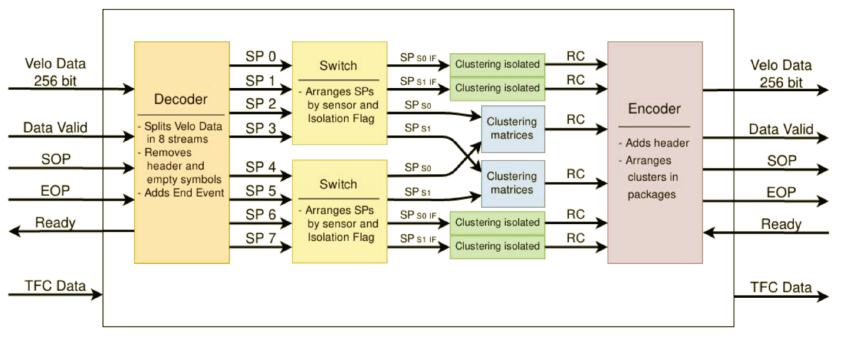
- To save FPGA logic, use sparse-matrix, instantiated on-demand as SP data flows in
- Isolated SP's are easier to handle and are given separate treatment (still in parallel)

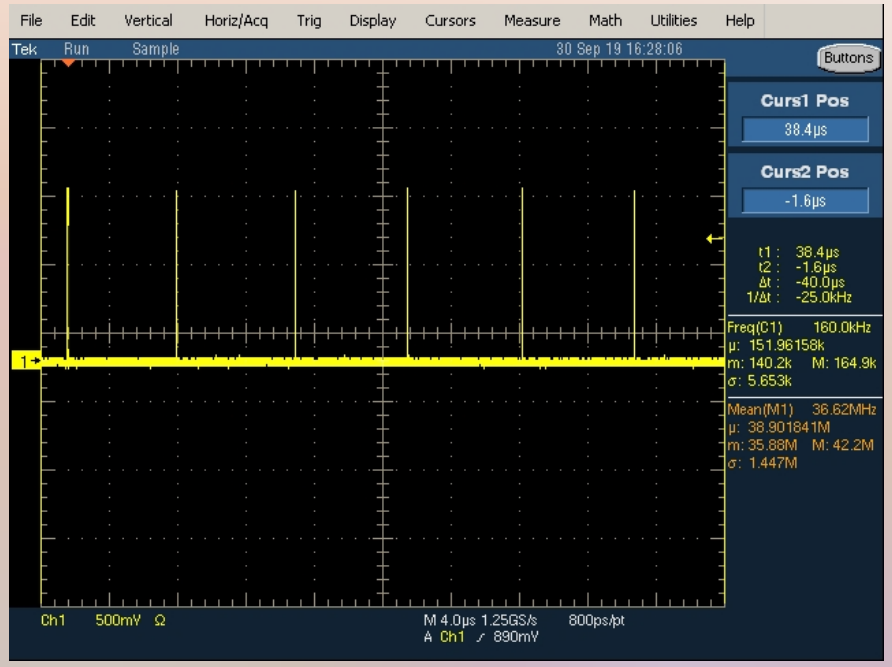


- **BLUE** SP belongs to the matrix, it fills the matrix
- **GREEN** SP doesn't belong to the matrix, it moves forward
- **RED** SP has reached a blank matrix because it doesn't belong to any previously filled matrices, it fills the blank matrix

- Ad end-of event, data pushed out in parallel to a network of mergers/switches, and serialized for next stage.
- Next event follows in immediately without any wait state [ACAT-19]

Test of Hardware implementation





- Firmware coded in VHDL, implemented in ALTERA Stratix-V FPGA
- Running on a 350 MHz clock
- VELO data from full LHCb simulation of the most crowded sensor in Minimum Bias sample

<u>Free running at 39 MHz@L=2*10^{33} without errors.</u>

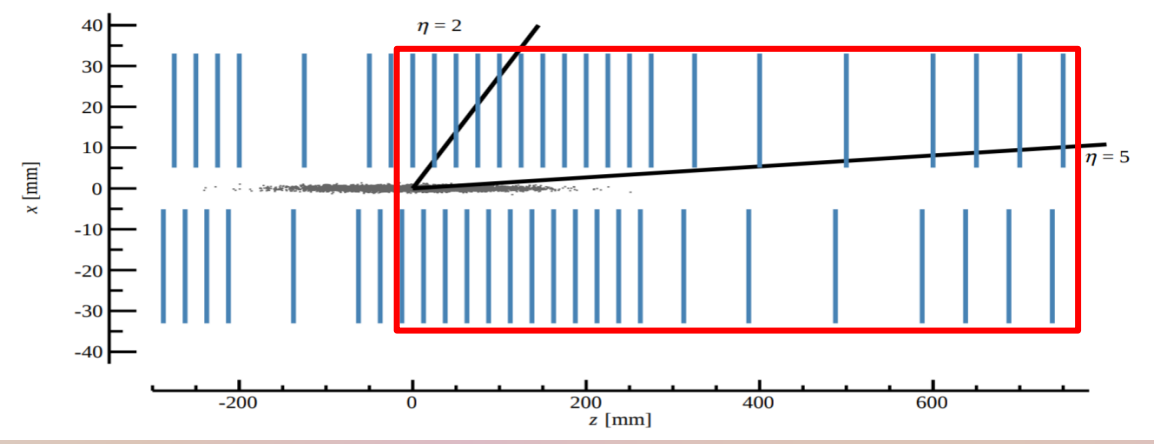
Effect on tracking performance

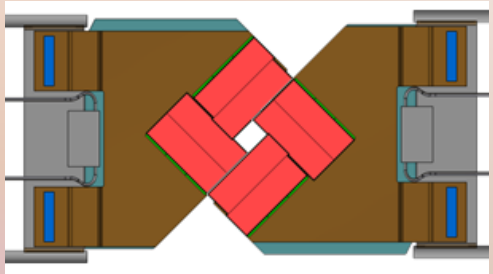
- Simulator matching hardware behavior bit-by-bit
- Run on $B \to \phi\phi$ samples (full LHCb simulation)
- Result: tracking performance closely matches CPU algorithm
  - Including check on the innermost 3 hits (critical for resolution)

		VeloClusterTracking (CPU)	FPGA + VeloClusterTracking
Velo tracks	**efficiency**	95,519% ± 0,014%	95,258% ± 0,014%
	clone	1,365% ± 0,008%	1,421% ± 0,008%
	hitEffFirst3	95,21%	94,51%
Long tracks	**efficiency**	97,705% ± 0,013%	97,493% ± 0,014%
	clone	1,291% ± 0,010%	1,361% ± 0,010%
	hitEffFirst3	95,70%	94,96%
Ghost tracks		0,8745% ± 0,0041%	1,0217% ± 0,0045%

- *Final implementability and physics performance checks ongoing, ahead of a decision whether to adopt this as the new Run-3 baseline.*
- *Implementation within VELO-readout card to maximize use of existing FPGA boards*

Step 2: Track reconstruction

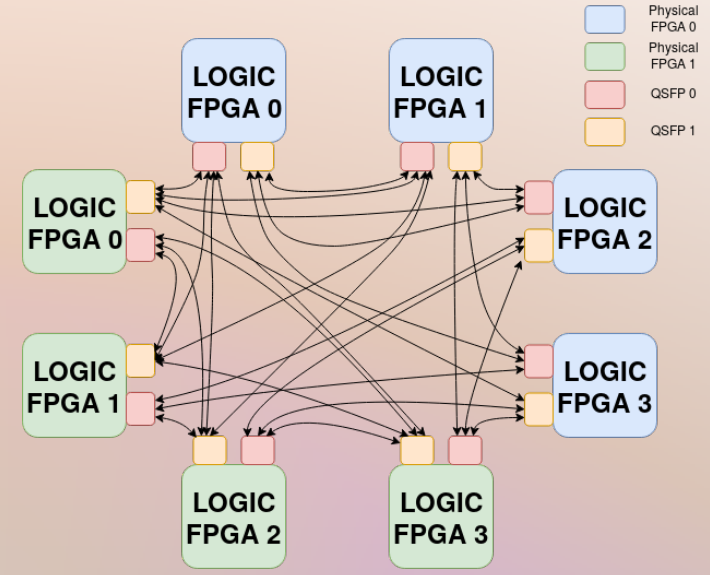




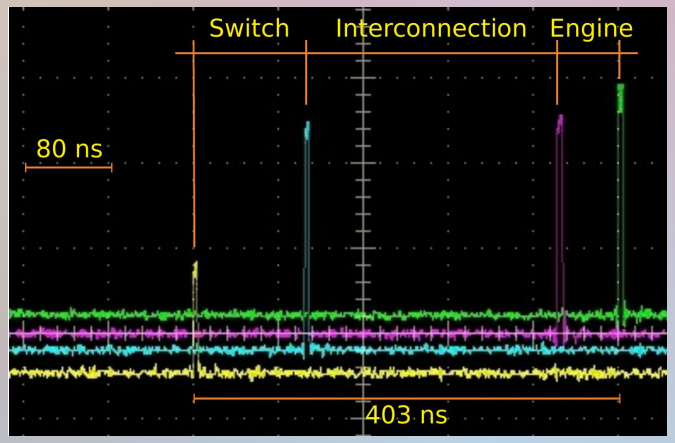
- VELO composed by 52 silicon pixel modules arranged in 26 layers
  - Each module read out by its own readout card in the EB
  - Plan to use data from layers 8 to 26 → **38 readout units**
  - Aim at forward-going tracks only. Other layers used mainly to optimize primary vertex precision - out of our current scope.

Connection Network

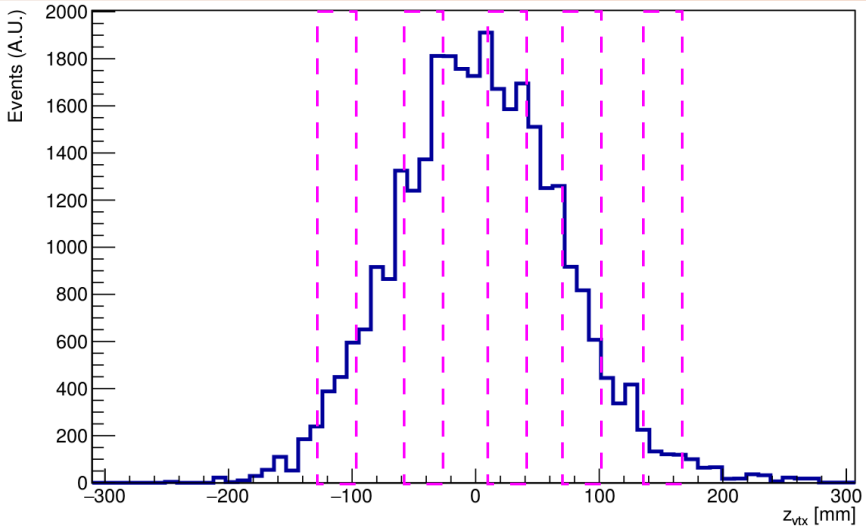
- Connection network an important part of system
- Plan to connect 38 cards with 4-5 full-mesh fast bi-directional optical link networks, 12-28Gbs
- Each subnet will have bridges to others via small number of extra links
- Bench test successfully performed on 8-node full-mesh network of 12Gbs links (~1/4 total)
- Full occupancy of >1 Tb/s b/w, no slowdown of throughput. Latency 250ns out of 400ns total



Test board, 2 FPGAs
96 total optical link

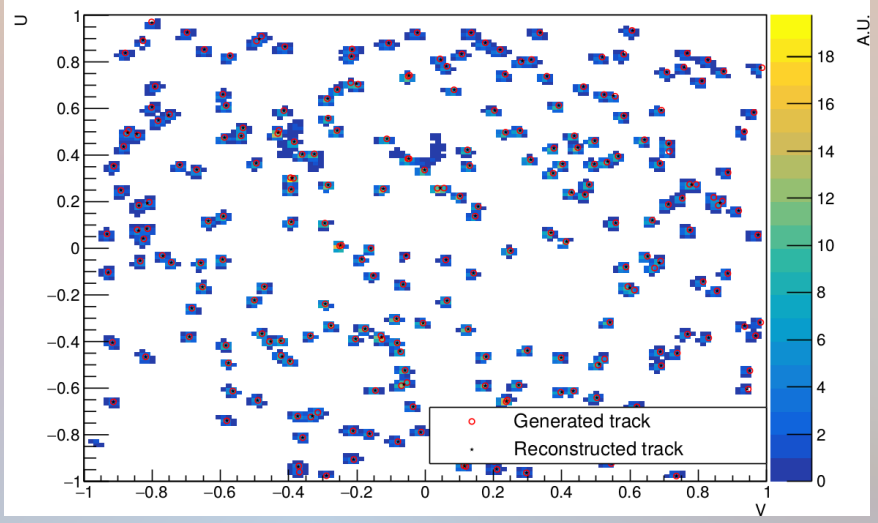


Segmentation of parameter space



Retina algorithm based on a 2D-ish space ⟹ segment track space in 10 z slices * 10000 cells each (100k total cells)

→ Fits FPGA logic available in current COTS PCIe cards.



Plot of cell excitation levels of a single 100*100 retina matrix, shows reliable track recognition

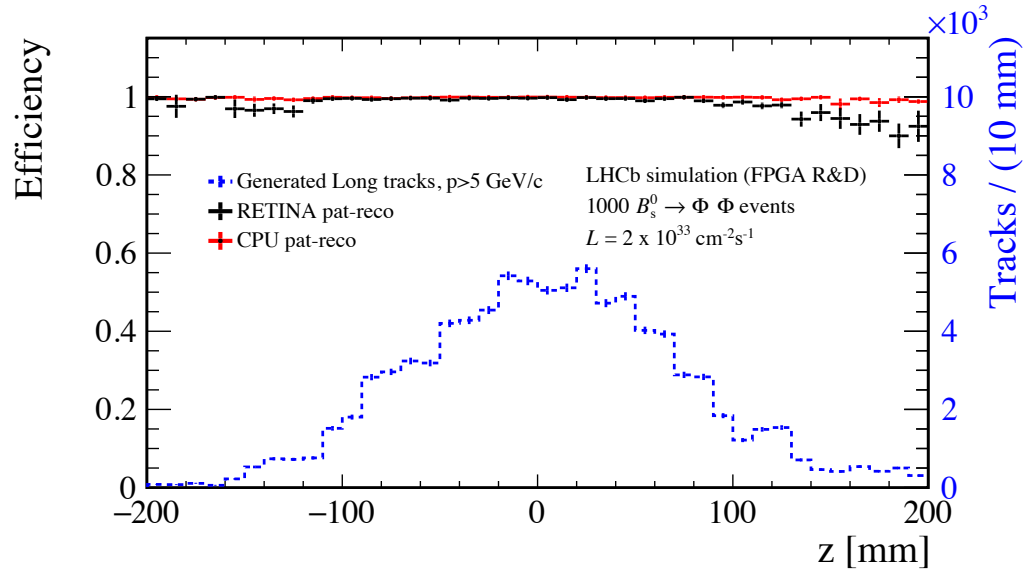
Local algorithm filters out noise from nearby z slices before output

Output lists all hits associated to each found track candidate.

Prepares packet for delivery to the HLT1 via the Event Builder - allows final fit to be performed in the HLT with use of best alignment/corrections

Tracking Performance from Simulation

- Use $B \rightarrow \phi\phi$ sample from official LHCb full simulation
- Run through bit-level simulation of tracking algorithm
- Inject result into LHCb's track performance benchmark
- Comparison with standard CPU algorithm shows ***very close efficiency performance*** on fiducial tracks
- Work still needed in merging back tracks split over multiple retina matrices and finalizing firmware



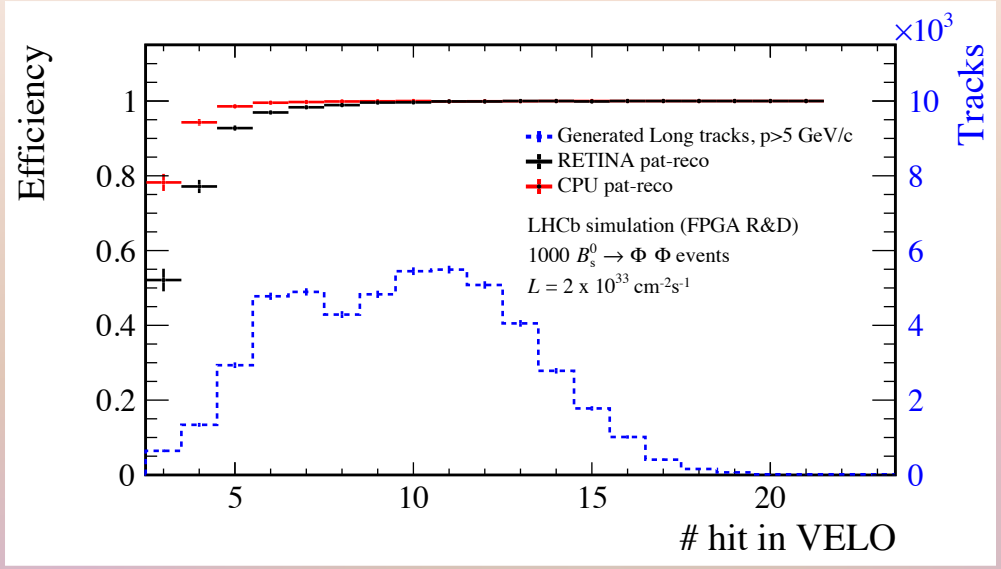
Track type	ε CPU pat-reco (%)	ε FPGA pat-reco (%) all z	ε FPGA pat-reco (%) fiducial z-region
Long tracks with $p > 5$ GeV/c and hits in VELO> 5	99.84 ± 0.02	99.27 ± 0.06	99.45 ± 0.05
Long tracks from b with $p > 5$ GeV/c and hits in VELO> 5	99.61 ± 0.13	99.24 ± 0.21	99.41 ± 0.18
Long tracks from c with $p > 5$ GeV/c and hits in VELO> 5	99.89 ± 0.12	98.50 ± 0.53	98.62 ± 0.53

Conclusions

- After a few years of R&D, now getting close to life-size demonstrator of FPGA-based 30-MHz tracking
- Detailed simulation studies show fairly good tracking performance for such fast tracking, based on extreme-parellization "retina" approach, and practical feasibility of implementation in current COTS FPGA systems
- Good hopes for future applications in Phase-II LHCb high-rate real-time reconstruction needs
- First DAQ integration tests in the LHCb Vertical-Slice setup planned in timescales of months.

Backup

Other tracking performance plots



Track type	ε CPU pat-reco (%)	ε FPGA pat-reco (%) all z	ε FPGA pat-reco (%) fiducial z-region
Long tracks with $p > 5$ GeV/c	99.56 ± 0.03	97.94 ± 0.09	99.15 ± 0.02
Long tracks from b with $p > 5$ GeV/c	98.94 ± 0.21	97.40 ± 0.38	98.88 ± 0.25
Long tracks from c with $p > 5$ GeV/c	99.45 ± 0.19	96.56 ± 0.78	97.97 ± 0.61