Demonstration of track reconstruction with FPGAs on live data at LHCb

Francesco Terzuoli *et al.* on behalf of the LHCb-RTA project

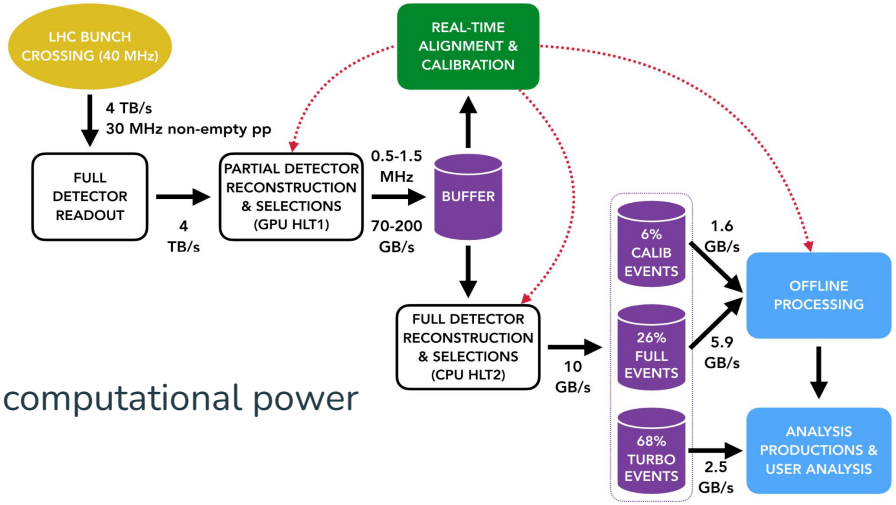
8th International Connecting The Dots Workshop

Mini-workshop on Real-time tracking: triggering events with tracks – Oct 10th-13th 2023

Event reconstruction at LHCb

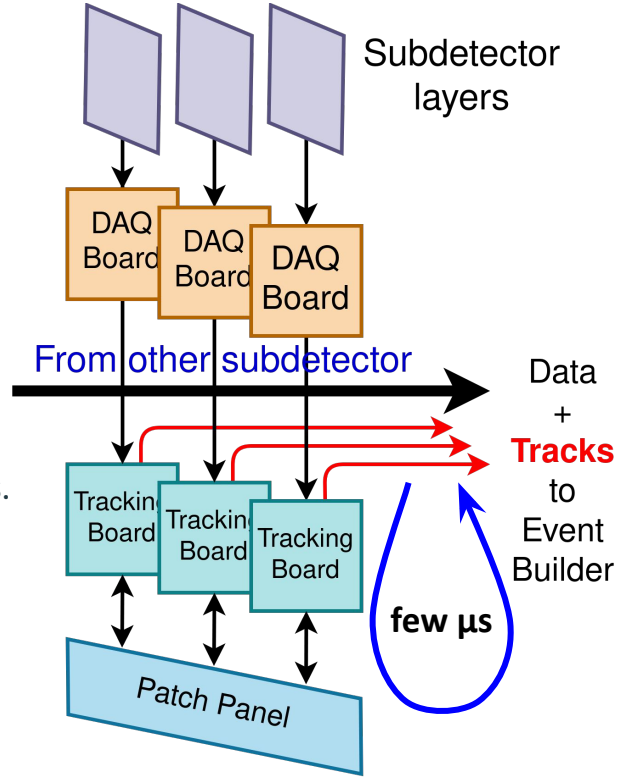
- LHCb is a forward spectrometer tailored for charm and beauty physics studies
- High cross section of interesting events [1]:
  → triggering on simple quantities is not possible
  → event fully reconstructed online by LHCb at the LHC average rate (~30 MHz)
- Heterogeneous solution in Run 3 (2022-2025):
  → heterogeneous trigger:
  GPU (HLT1) + CPU (HLT2)
- What about HL-LHC?
  → Increase in luminosity translates in higher computational power
- LHCb established a coprocessor testbed:
  → testing new heterogeneous computing solutions with realistic conditions provided
- We developed a tracking system demonstrator for the LHCb Vertex Locator based on FPGAs

LHCb-FIGURE-2020-016

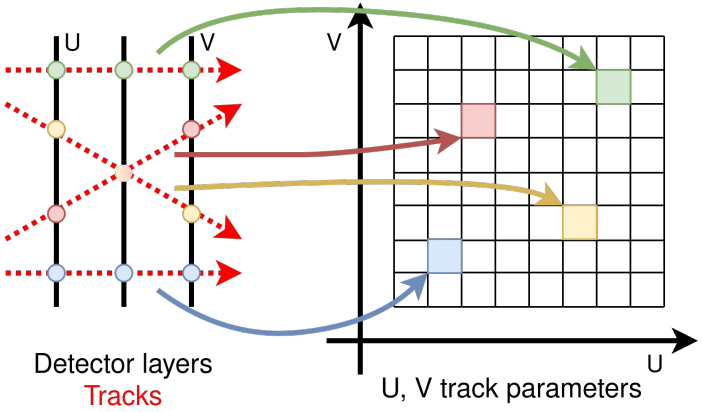


Why use FPGAs

- Modern FPGAs can perform highly parallel processing, with high throughputs and low latencies.
- Less power-hungry than CPUs and GPUs
- **Tracking system can be moved at the earlier stages**
  → **tracks available before event building**
  → leave resources at higher level to other tasks
- Reconstructing tracks requires to combine data from several different layers, typically read out separately by the DAQ boards.
  - FPGAs have high-bandwidth transceivers (XCVRs) that allow to exchange information with great flexibility.
- The “Artificial Retina” is a highly-parallel architecture conceived for this scenario [2]

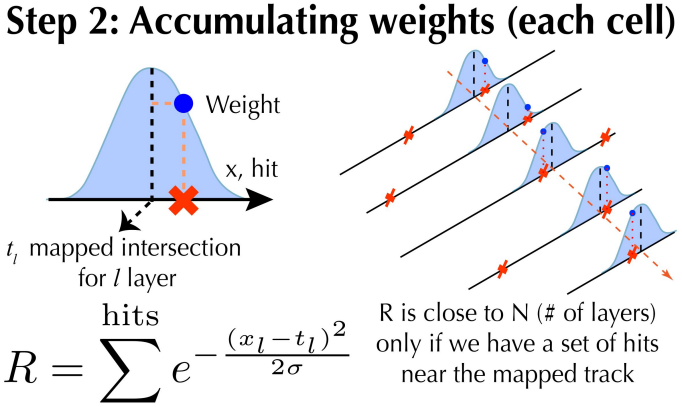


The “Artificial Retina” architecture



- Track parameter space represented by a matrix of processing units (cells)

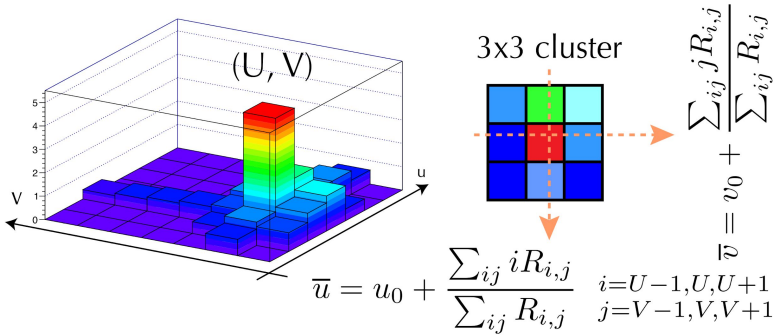
The “Artificial Retina” architecture



- Track parameter space represented by a matrix of processing units (cells)
- Each cell computes a weighted sum of hits near the reference track

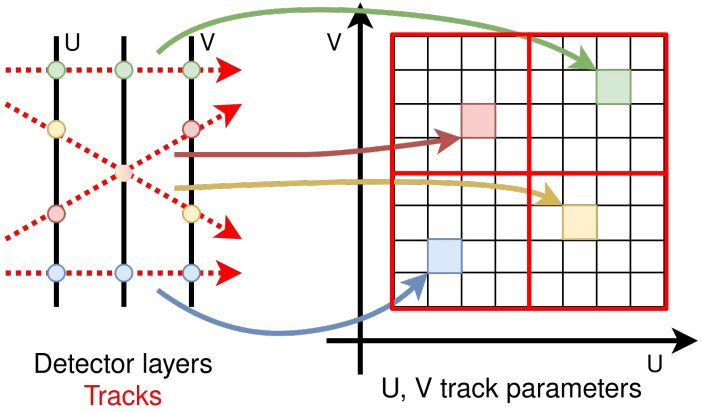
The “Artificial Retina” architecture

Step 3: Find the local maxima and compute centroid



- Track parameter space represented by a matrix of processing units (cells)
- Each cell computes a weighted sum of hits near the reference track
- Reconstructed tracks identified as local maxima in the cells matrix response
  - Interpolating responses of nearby cells for obtaining real tracks parameters

The “Artificial Retina” architecture



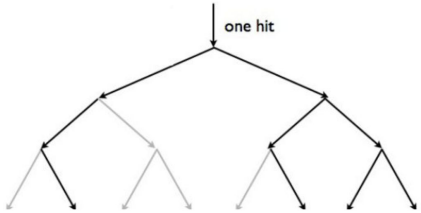
- Track parameter space represented by a matrix of processing units (cells)
- Each cell computes a weighted sum of hits near the reference track
- Reconstructed tracks identified as local maxima in the cells matrix response
  - Interpolating responses of nearby cells for obtaining real tracks parameters

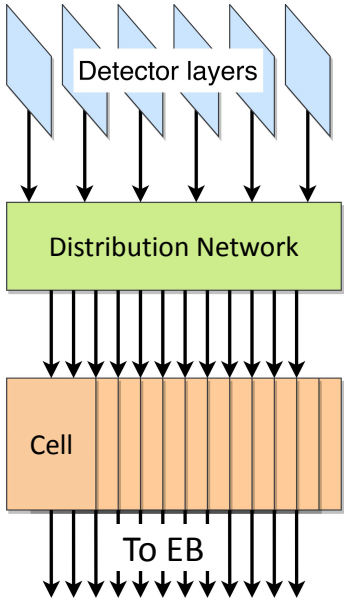
Cells work in a fully parallel way *for reaching high-throughput and low-latencies*

Overcoming FPGA size limitations (without increasing latency) with ***cells spread over several chips***

System implementation

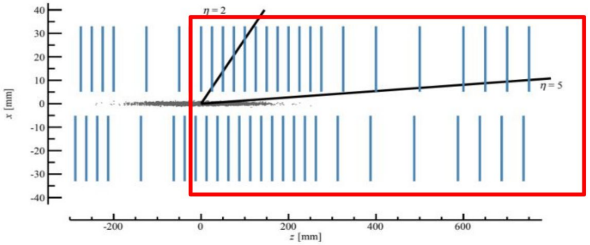
- Input from detector DAQ
- Data preparation (detector specific)
  - e.g. Hit clustering
- Distribution network
  - Switch: routes hits only to specific cells
  - Optical communication: exchanges hits between tracking boards
  - Hits get delivered to cells where their contribution is significant
    - More than one cells can need the same hit
    - Bandwidth increasing the deeper we delve in the network
- Cell
  - Engine: computes and accumulates weights
- Track Processing Unit (matrix of adjacent cells)
  - Max-finder: finds local maxima and interpolates responses
- Output to Event Builder



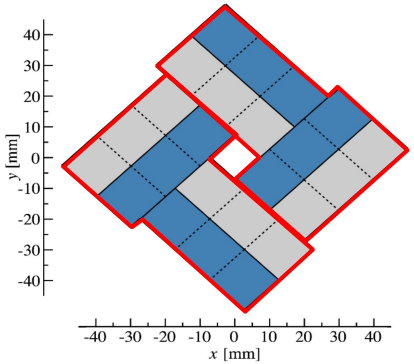


The VELO detector

- The VErtex Locator is a crucial subdetector for LHCb physics program
- 38 modules in the forward region (< 10% of LHCb data size).
- Time consuming (25% of HLT1 time budget [3])
  → Compact FPGA system can significantly ease the load of HLT1
- Physics performances study of FPGA tracker system available [4]
- A good test-case for future and larger-scale applications



VELO top view



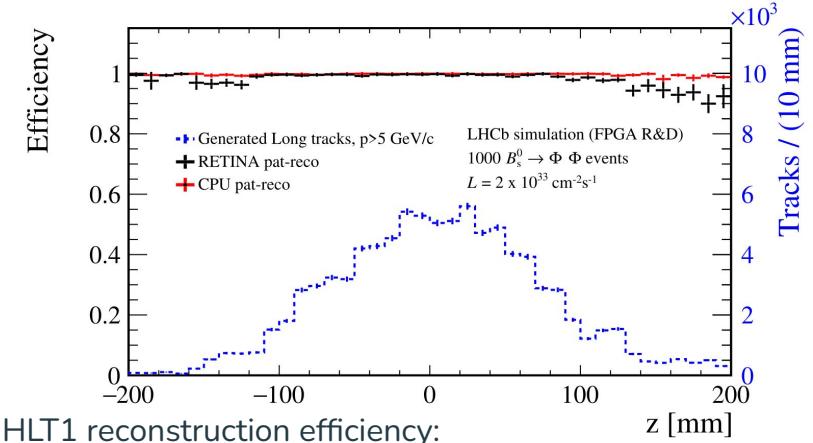
VELO front view

- Partial data preparation executed by the detector
  - Hit clustering currently adopted in Run 3
    - Architecture originated from “Artificial Retina” [5]
    - Integrated in DAQ boards

Physics performances

- Studies done with C++ emulation of the “Artificial Retina” architecture
- *Bs→ΦΦ* sample from official LHCb full simulation
- Inject result into LHCb’s track performance benchmark
- Comparison with standard CPU algorithm shows very close efficiency performance on fiducial tracks (-200mm < z < 200mm) [4]

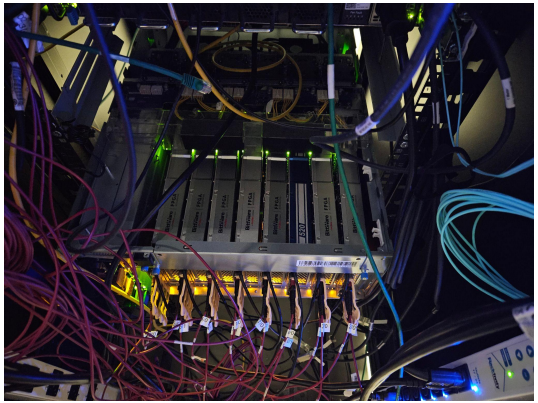
Track type	ε CPU pat-reco (%)	ε FPGA pat-reco (%) all z	ε FPGA pat-reco (%) fiducial z-region
Long tracks with $p > 5$ GeV/c and hits in VELO> 5	99.84 ± 0.02	99.27 ± 0.06	99.45 ± 0.05
Long tracks from b with $p > 5$ GeV/c and hits in VELO> 5	99.61 ± 0.13	99.24 ± 0.21	99.41 ± 0.18
Long tracks from c with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



HLT1 reconstruction efficiency: comparison between CPU and “Artificial Retina”.

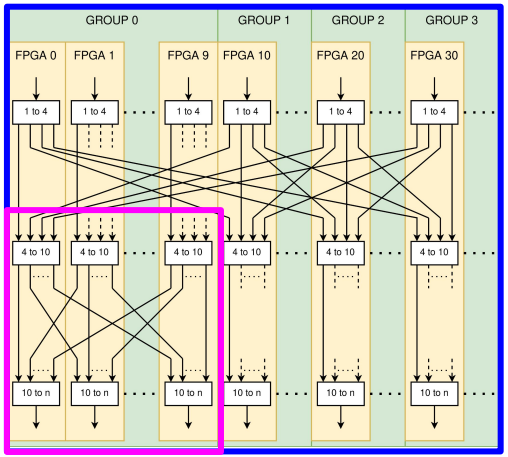
The Demonstrator

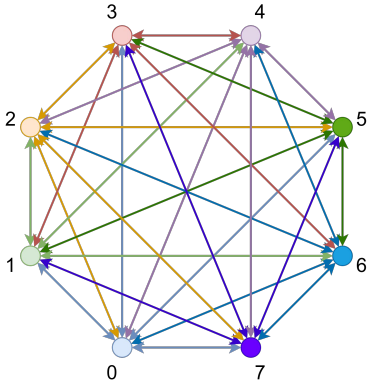
- “Artificial Retina” demonstrator installed at the testbed facility
- Target is demonstrating the capability of
  → Reconstructing an actual detector
  → Working in real-time at nominal LHCb luminosity ($2x10^{33}$ $cm^{-2}s^{-1}$)
  → Being integrated in the LHCb DAQ environment
- Current design
  - Coverage: VELO quadrant
  - 8 Intel Stratix 10 FPGA boards processing 16 VELO modules
    - 2 VELO modules per board
  - Data exchanged by distribution network (Switch + optical communication)
  - Engines (grouped in TPUs) compute weights and reconstruct tracks
    - Engines on different boards cover different regions on track parameters space
  - Tested with Montecarlo simulated data and output is compared bitwise to C++ emulation

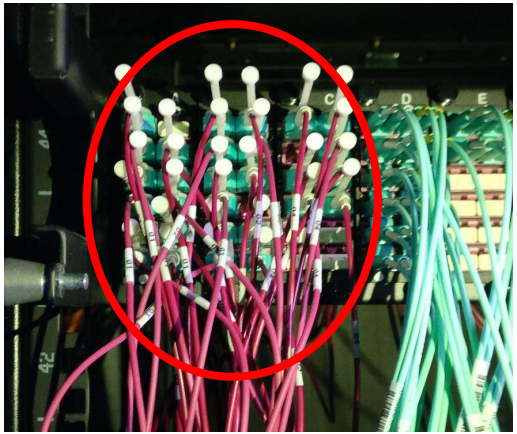


Distribution network

- Portion of the whole VELO distribution network currently implemented
  - 8 nodes full-mesh network
  - 28 full-duplex links at 25.8 Gbps
  - Total bandwidth 1.41 Tbps

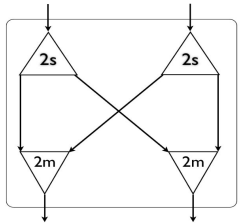




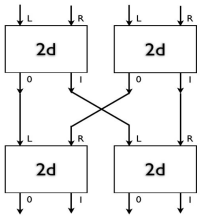


Optimising the hits distribution

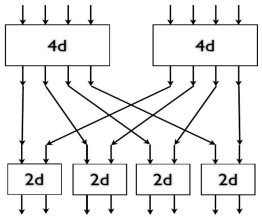
- The switch (modular design) handles the hits distribution by routing the hits to the correct implemented TPUs using look-up-tables (LUTs)
- Implemented 64 TPUs over 8 boards (8TPUs/board)
- First optimised: last step of the switch (8x8 dispatcher)
  - One switch/chip: 1TPU/Out_line and 2 VELO modules/Input_line
- The optimisation
  - Pairing the TPUs (2 by 2) with highest number of common hits
  - Iterate over the paired TPUs as we move to higher switch levels
    - Move hits duplication towards last switch layers



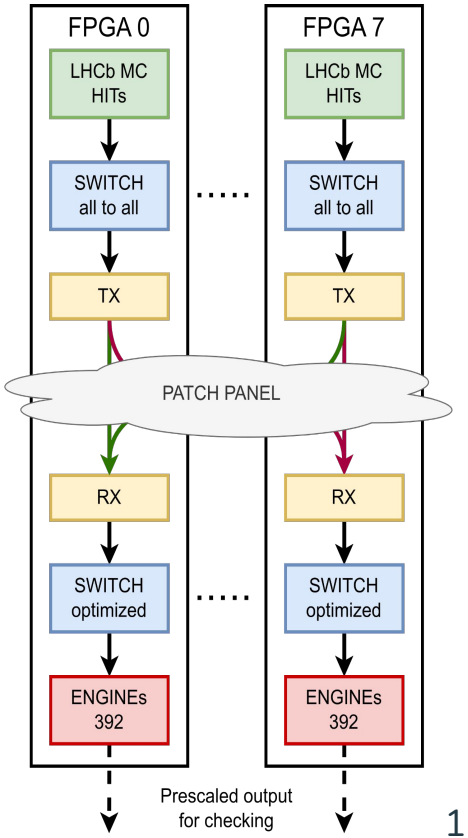
2-way dispatcher (2d)



4-way dispatcher (4d)

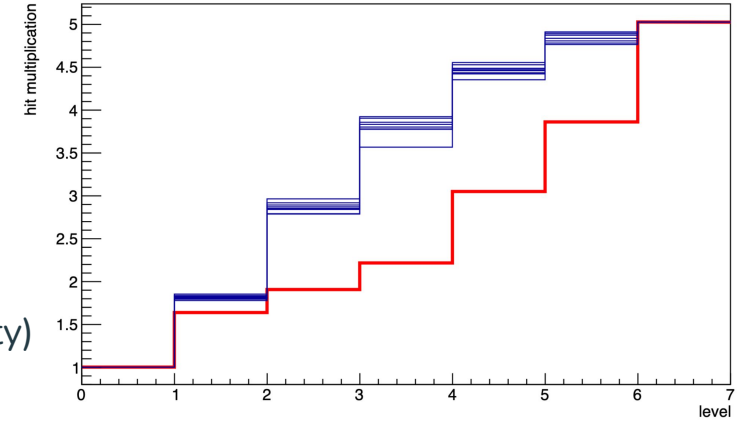


8-way dispatcher (8d)



Optimisation results

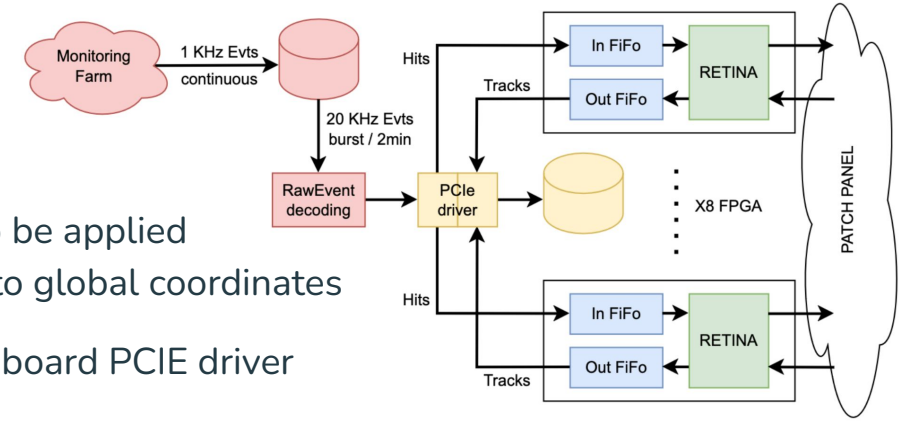
Studies on the switch by F. Novissimo [6]



- Optimised by running over 1000 simulated events at full luminosity
- Busiest lines lie at the output of this switch and in entrance to the engines
- The **speed attainable is improved from** 2.38MHz at 1.43×10^{32} $cm^{-2}s^{-1}$ (scaled to **170kHz** at full luminosity) **to 10.9MHz** still at full luminosity
  - Speeds of designs with a dual-input line per TPU (two 8x8 dispatcher side-by-side each handling half the arriving hits for the VELO modules)
- Moved to quad-input design (implementing two 8x16 dispatchers, replacing one level with splitters) and firmware optimisation (dead times reduced) → speed increased to **19.0 MHz**
  - **Still many handles for improving this, currently being worked on**
- Runned several days (~10) without hiccup (much higher than typical bunch life)
- Exact bitwise alignment with C++ emulator

Running live on data from collisions

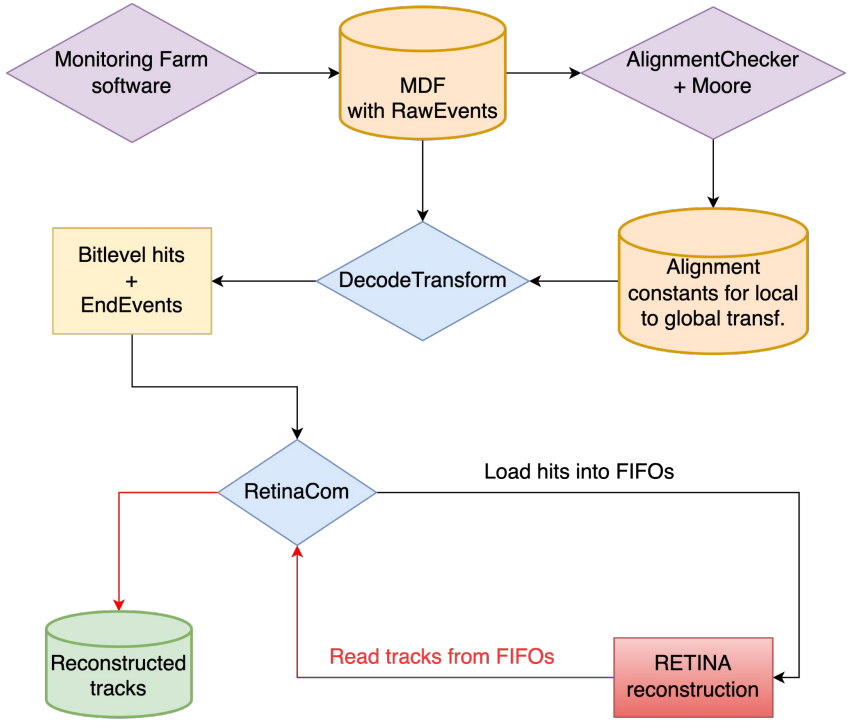
- Testbed facility is fed with data from the Monitoring Farm (1kHz evts) and they are stored on disk
  - Chunks of RawEvents (1.9MB/chunk)



- In addition to hit clustering, alignment need to be applied
  - Conversion of VELO clusters from local to global coordinates
- Need communication with FPGAs using stock board PCIE driver
  - Loading VELO hits to the boards
  - Reading reconstructed tracks from the boards
  - Checking FPGAs error registers
- Moreover due to the the demonstrator not being integrated in the LHCb online system
  - Decoding incoming RawEvents from Monitoring Farm
  - Selecting the detector sources (VELO modules compatible with the chosen quadrant)

Live data feeding system

- Custom C++ software developed for such tasks
  → Running on the server mounting the boards
- Flow managed by 4 distinct process
  - Monitoring Farm Software
  - AlignmentChecker
    (invokes internal LHCb software
    for offline analysis)
  - DecodeTransform
  - RetinaCom
- All concurrently running
  - Communication between processes via
    Named Pipes (software FIFOs)
  - RawEvents, alignment constants and
    rec. Tracks saved on disk

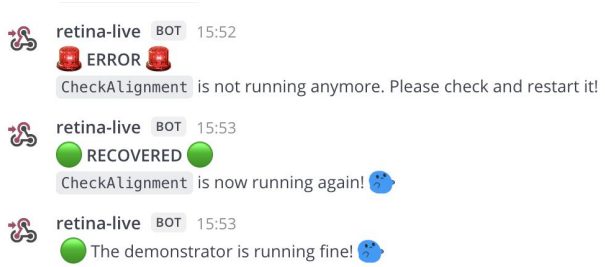


The RetinaLive dataflow

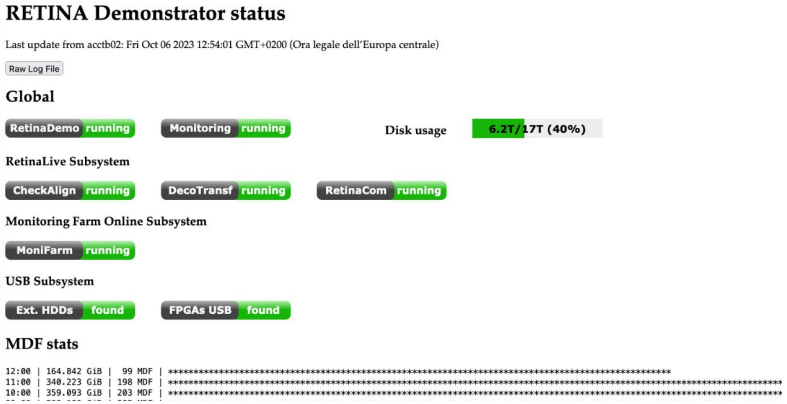
- **CheckAlignment**
  - From run number retrieve the alignment constants using internal LHCb software
- **DecodeTransform**
  - Perform **local to global transformation in real-time** as done at HLT1
  - Transforms global coordinate to the u-v Retina parameter space
- **RetinaCom**
  - Direct memory Access (DMA) still under development → For now can access one register/clock (one word/clock)
    - Speed-up by multi-threading (1 thread/board)
  - Load clusters onto the chips, retrieved reconstructed tracks and stores them on disk
    - Loading performed in small chunks and if input FIFOs are empty
    - Reading of output FIFOs until depleted
      - Avoiding FIFOs overflow
  - Checks on error registers after writing and reading from the chip

Monitoring and running the system

- Errors are handled within each different process
  → Exits implemented for particular critical occurrences
- A shell script checks that the process are running
- A Mattermost Webhook and a simple ‘curl’ in the monitoring script take care of alerting us if the demonstrator stopped

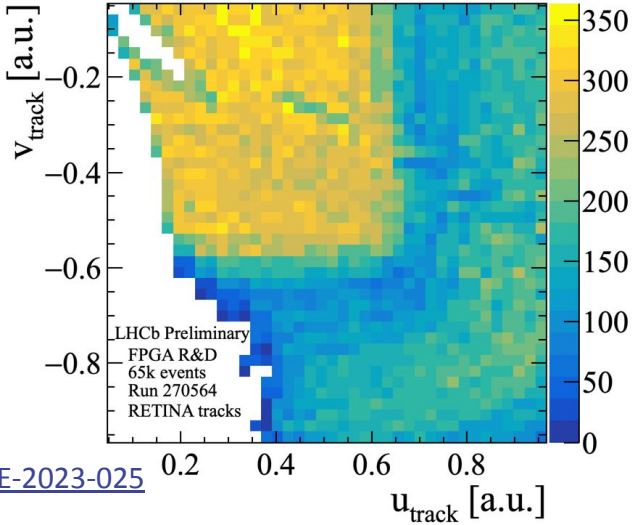


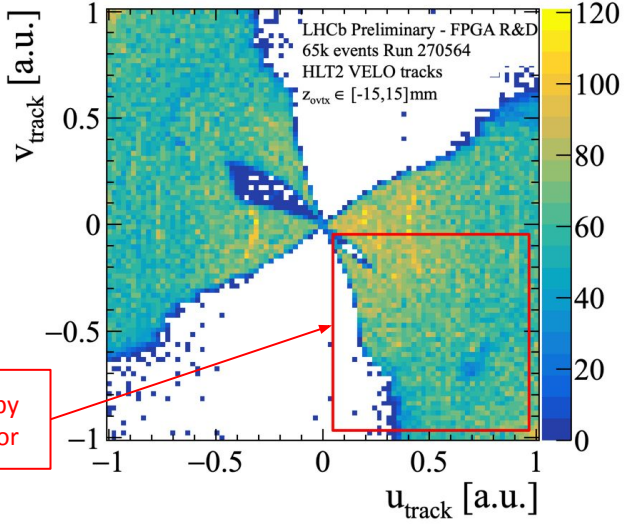
- Deployed website
  - Live status of the demonstrator
  - Statistics of the processed files
- The system has run since 1st September
  - Errors (6 observed) due to causes independent from the firmware or the data feeding chain



Live reconstruction from demonstrator

- The output binary files have been decoded and the tracks plotted in the u-v parameter space (left)
- The same RawEvents are reprocessed offline and the tracks reconstructed at HLT2 plotted (right)





LHCb-FIGURE-2023-025

Empty region is compatible with anomalous VELO open position in 2023 data taking (halved acceptance)

Tracks from demonstrator appear to be sensible from qualitative comparison with HLT2

System initially not tuned for VELO open → still good response → quickly tuned → higher efficiency

Results

- The demonstrator is capable of reconstructing the VELO detector (a quadrant) in **real-time**
- As we work for full integration in the LHCb online chain
  - Custom data-feeding chain has been developed for bridging the gap between the Monitoring Farm and the demonstrator
    - Working without hiccups for more than one month
    - Can steadily provide detector hits to the FPGA boards
- Tracks reconstructed in real-time are sensible when *qualitatively* compared with HLT2 performance
- Good reconstruction performance observed on simulated data with C++ emulator
  → Bitwise alignment between emulator and hardware assures good performance also of the latter
- In progress
  - Developing DMA for PCIe communication with the FPGA boards
  - Optimising a few points of the design in order to reach the desired 30MHz
- Proposal for a real tracking system in Run 4 is currently under review by LHCb collaboration

Thanks for your attention!

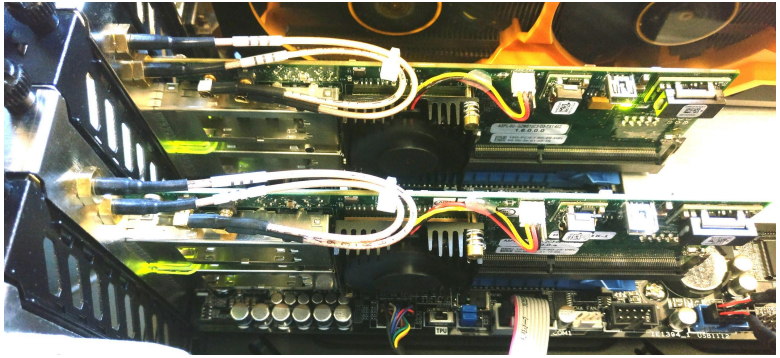
Bibliography

1. LHCb Collaboration, **LHCb Trigger and Online Upgrade Technical Design Report**, CERN-LHCC-2014-016
2. G. Punzi et al. on behalf of the LHCb Real-Time Analysis project, ***Real-time reconstruction of pixel vertex detectors with FPGAs***, PoS(Vertex2019) - Tracking and vertexing
3. LHCb Collaboration, ***LHCb Upgrade GPU High Level Trigger Technical Design Report***, CERN-LHCC-2020-006
4. G. Tuci, ***Reconstruction of track candidates at the LHC crossing rate using FPGAs***, CHEP 2019
5. G. Bassi *et al.*, ***A FPGA-Based Architecture for Real-Time Cluster Finding in the LHCb Silicon Pixel Detector***, IEEE Transactions on Nuclear Science, vol. 70, no. 6, pp. 1189-1201, June 2023
6. F. Novissimo, ***Real-time reconstruction of tracks with RETINA algorithm at LHCb***, Master thesis presented at Università di Pisa, Pisa, 2022

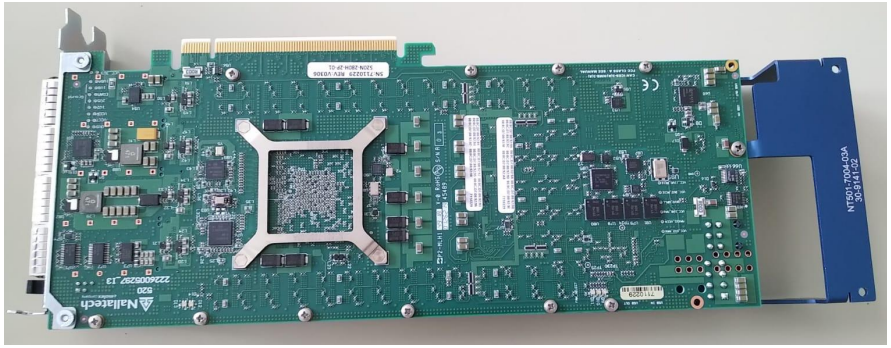
Backup

Hardware

- Prototyping board,
  2 Intel Stratix V FPGAs,
  96 optical links



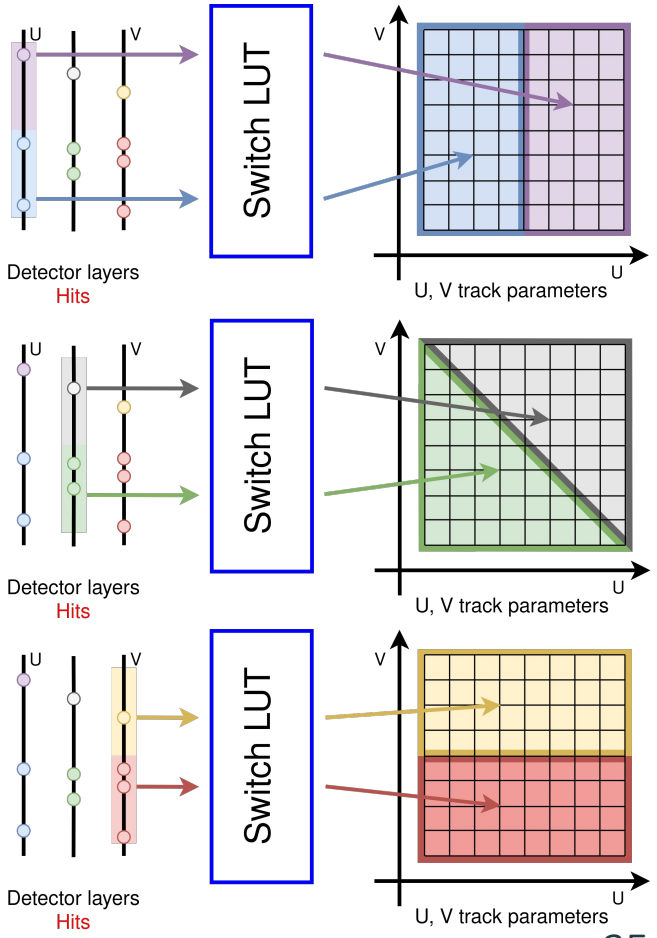
- PCIe 8x board, 1 Intel Arria V GX FPGA, 8 optical links



- PCIe 16x board, 1 Intel Stratix 10 FPGA, 16 optical links

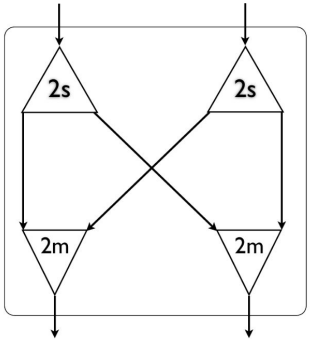
The Distribution Network

- Hits are provided to different Tracking boards arranged by sub-detector DAQ board.
- A custom distribution network rearranges the hits by track parameters coordinates (similar to a “change of reference system”).
- Using Lookup Tables (LUTs), the Distribution Network delivers to each cell only hits close to the parametrized track, enabling large system throughput.
- The Distribution Network is a single entity transversal to all the Tracking boards.
- We designed a modular Distribution Network spread over the same array of FPGAs performing the tracking.

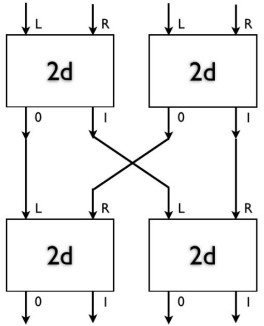


Switch

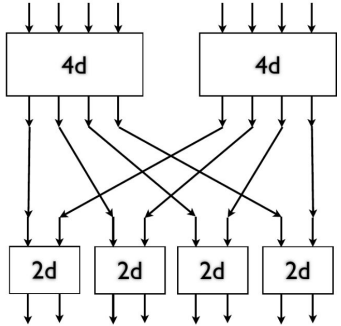
- 2-way dispatcher (2d): 2 splitters (1 input - 2 outputs) and 2 mergers (2 inputs - 1 output).
- Combining 2-way dispatchers is possible to build a switch with the desired number of lanes:
  - Switch with $N = 2^n$ lanes requires M 2-way dispatchers: $\begin{cases} M(0) = 0 \\ M(n) = 2M(n-1) + 2^{n-1} \end{cases}$
- We can implement any 2^n lanes switch changing a single parameter.



2-way dispatcher (2d)



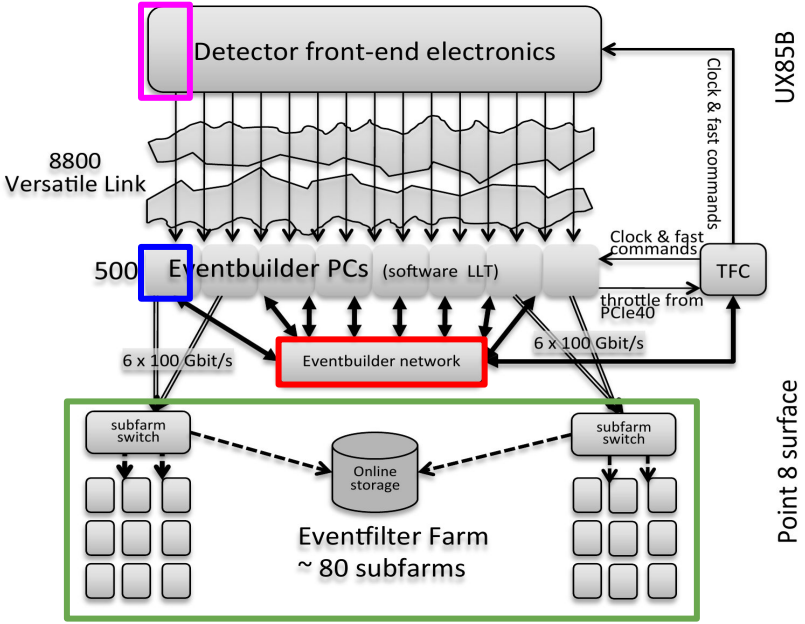
4-way dispatcher (4d)

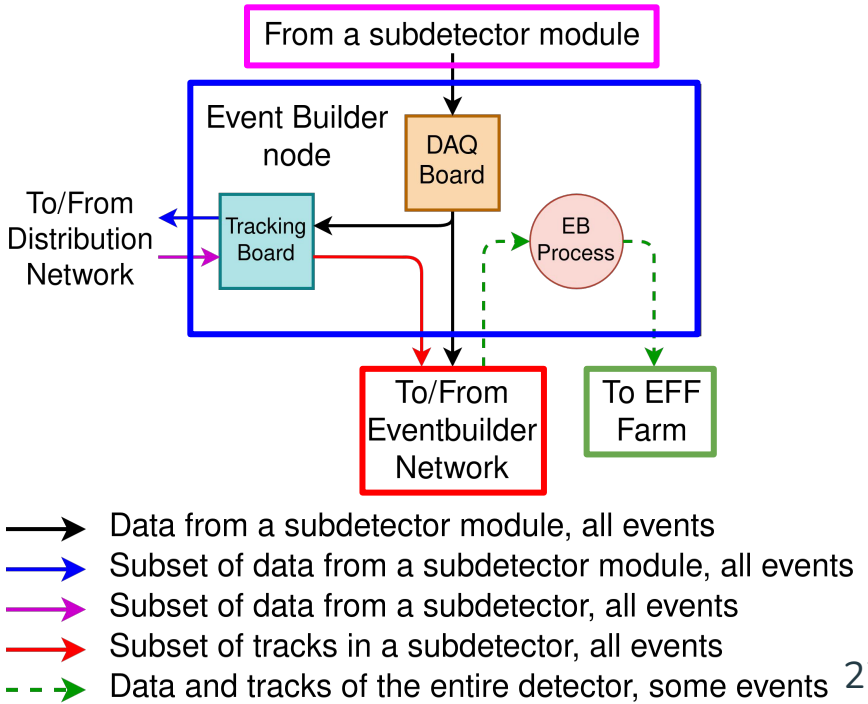


8-way dispatcher (8d)

Integration in DAQ system

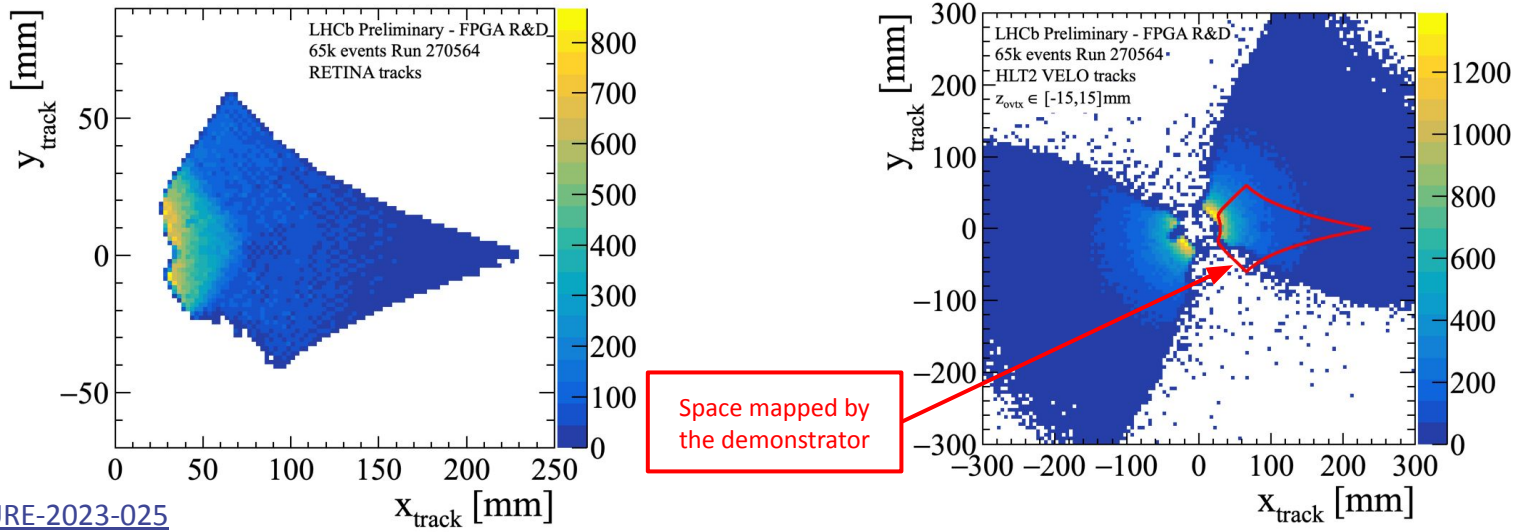
- The “Artificial Retina” could find a place in the Event Builder nodes using PCIe boards.
- The Event Builder collects the tracks and performs the building, treating the “Artificial Retina” like a virtual sub-detector.





Reconstructed tracks in real x-y space

- The reconstructed tracks by the demonstrator (left) and HLT2 (right)



LHCb-FIGURE-2023-025

Tracks from the demonstrator are anti-transformed from the u-v parameter space to their intersection with a transverse virtual plane located at z=700mm, from origin of the LHCb coordinates system