Real-time pattern recognition with FPGA at LHCb, an O(n) complexity architecture

Federico Lazzari on behalf of the LHCb collaboration

27th International Conference on Computing in High-Energy and Nuclear Physics
CHEP2024 – October 19-25

The challenge

- Progress of experiment goes together with increasing data processing rate.

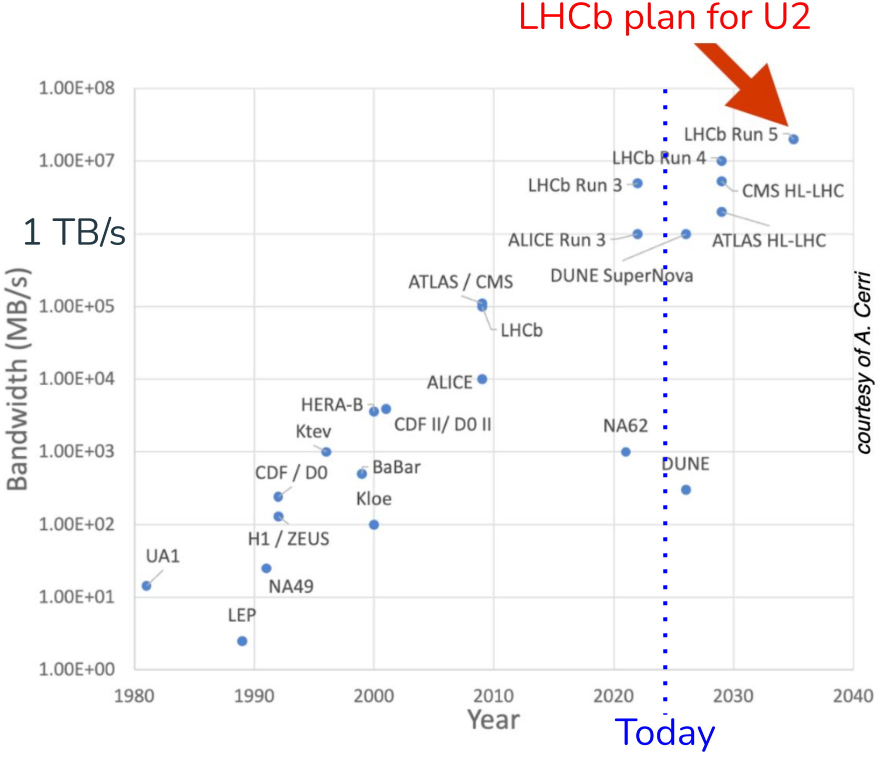
- Flavor physics at low P_T is more demanding: LHCb have a higher data rate than other LHC experiment even if smaller and with lower lumi.

- In Run 5 (2035) luminosity will be increased by a factor up to 7.5 [LHCB-TDR-026].

- Reconstruction complexity is typically $O(n^2)$
  → 50x computational power.

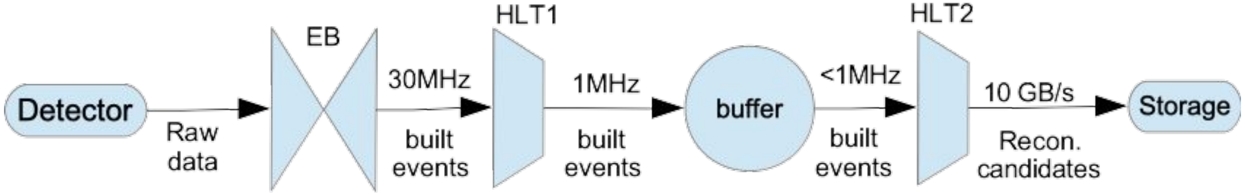
- Renew reconstruction paradigm is mandatory.

LHCb plan for U2



courtesy of A. Cerri

The LHCb reconstruction model



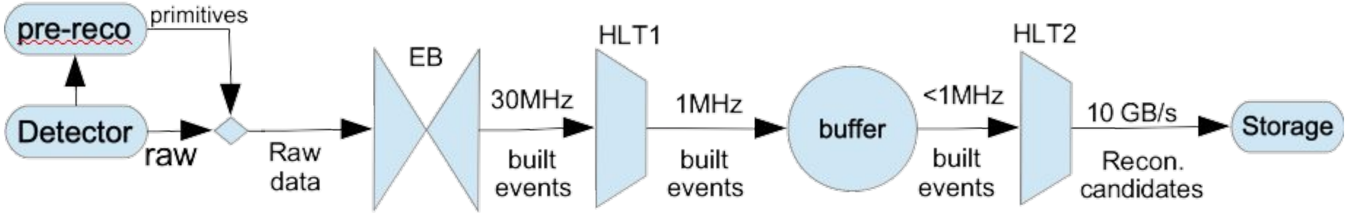
- Flavour physic has very-high cross section respect to Higgs and EW: $\sigma_b \sim 10^4\ \sigma_Z$ and $\sigma_b \sim 10^7\ \sigma_H$
  → No L0 trigger on simple quantities (e.g. P_T , E_T , muons) [LHCB-TDR-016, Alessandro talk Mon Track2].

- Reconstructs of every event, at the LHC average rate (~30 MHz):
  - HLT1 (GPU): partial reconstruction.
  - HLT2 (CPU): full detector reconstruction and final selection.

- Alignment computed between HLT1 and HL2 (buffer).
  - Provides offline quality to HLT2.

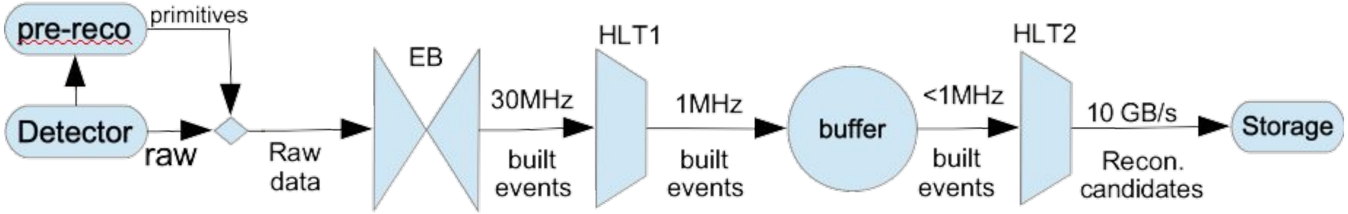
- To cope with higher luminosities we need to accelerate HLT.

Toward primitive-based reconstruction



- Reconstruct intermediate data (primitives) using “local” information.
- Embed primitives (e.g. clusters, track segments) in raw data.
    - Off-loads HLT from processing tasks.
    - Allows to reduce data flow at the source (e.g. dropping hits not part of tracks).
- Not trivial:
    - Must process all the events (30 MHz).
    - Constrained latency → can’t rely on time-multiplexing.
- This paradigm works only if the pre-processing has a complexity < $O(n^2)$.

Toward primitive-based reconstruction



- Reconstruct intermediate data (primitives) using “local” information.
- Embed primitives (e.g. clusters, track segments) in raw data.
  - Off-loads HLT from processing tasks.
  - Allows to reduce data flow at the source (e.g. dropping hits not part of tracks).
- Not trivial:
  - Must process all the events (30 MHz).
  - Constrained latency → can’t rely on time-multiplexing.
- This paradigm works only if the pre-processing has a complexity < $O(n^2)$.

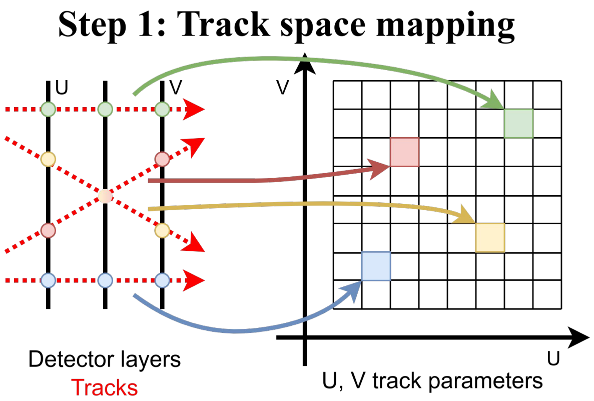
The “Artificial Retina” architecture allows us to do this.

The “artificial retina” architecture

- Highly-parallel architecture for pattern recognition.

The "artificial retina" architecture

- Highly-parallel architecture for pattern recognition.

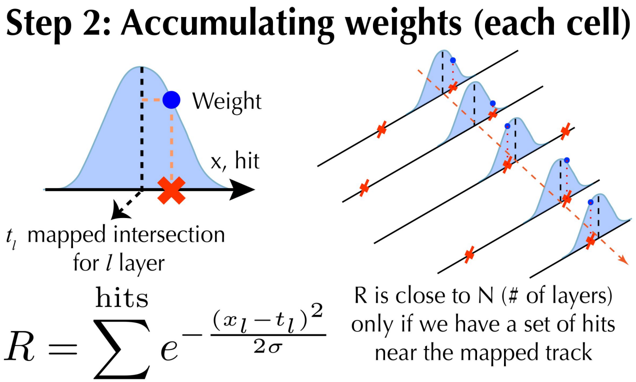


[LHCb-PUB-2024-001]

- Track parameter space represented in a matrix of processing units (cells).
  - Each cell specialised to reconstruct tracks neighbour to a reference track.

The “artificial retina” architecture

- Highly-parallel architecture for pattern recognition.


Step 2: Accumulating weights (each cell)
Weight
x, hit
t_l mapped intersection for l layer

$$R = \sum^{\text{hits}} e^{-\frac{(x_l - t_l)^2}{2\sigma}}$$

R is close to N (# of layers) only if we have a set of hits near the mapped track

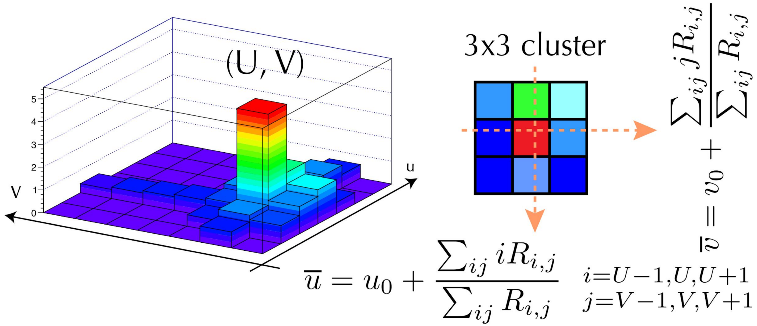
[LHCb-PUB-2024-001]

- Track parameter space represented in a matrix of processing units (cells).
  - Each cell specialised to reconstruct tracks neighbour to a reference track.
- Each cell computes its response (*R*) as the weighted sum of hits.

The “artificial retina” architecture

- Highly-parallel architecture for pattern recognition.

Step 3: Find the local maxima and compute centroid [LHCb-PUB-2024-001]



$$\overline{v} = v_0 + \frac{\sum_{ij} jR_{i,j}}{\sum_{ij} R_{i,j}}$$

$$\overline{u} = u_0 + \frac{\sum_{ij} iR_{i,j}}{\sum_{ij} R_{i,j}}$$

$$i = U-1, U, U+1$$
$$j = V-1, V, V+1$$

- Track parameter space represented in a matrix of processing units (cells).
  - Each cell specialised to reconstruct tracks neighbour to a reference track.

- Each cell computes its response (*R*) as the weighted sum of hits.

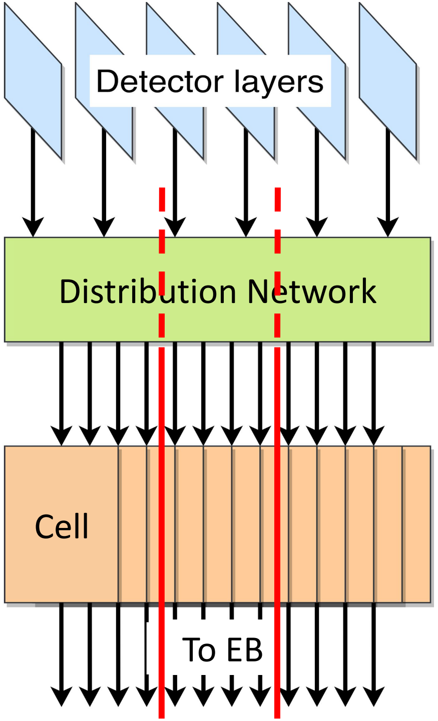
- Local maxima in the matrix of cells response correspond to reconstructed tracks.

Unique features

1) Specifically conceived for FPGAs:

- Programmable logic resources.
  - ○ Each component has its dedicated resources.
  - → Everything works in parallel.
  - → No need to access shared memory.
- Programmable data paths.
  - ○ FPGAs can fan out signals and sustain very-high bandwidth.
  - → Each Hit is distributed to the cells in parallel.
- Numerous high-bandwidth transceivers (XCVRs).
  - ○ Can overcome size limitation exchanging data between FPGAs.
  - → Cells are spread over several chips.

2) Tracks reconstructed processing hits and **not** their **combinations**.



The “artificial retina” complexity

Step 1:

- Configuration stage: happens before data taking.
  - → No processing time consumed.

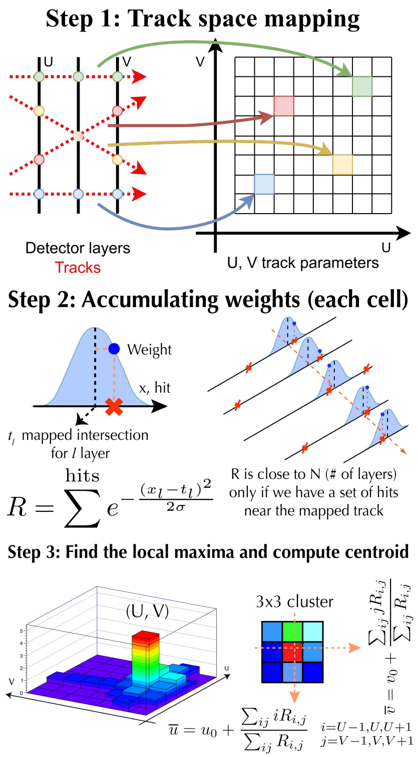
Step 2:

- Cells work in parallel.
  - → Processing time do not depend on the number of cells.
- Each cell can process few hits per clock cycle.
  - → Processing time **scales linearly** with the number of hits.

Step 3:

- Cells check if they represent local maxima in parallel.
  - → Processing time do not depend on the number of cells and tracks.

Can we prove this?



Step 1: Track space mapping

Step 2: Accumulating weights (each cell)

$$R = \sum^{\text{hits}} e^{-\frac{(x_l - t_l)^2}{2\sigma}}$$

R is close to N (# of layers) only if we have a set of hits near the mapped track

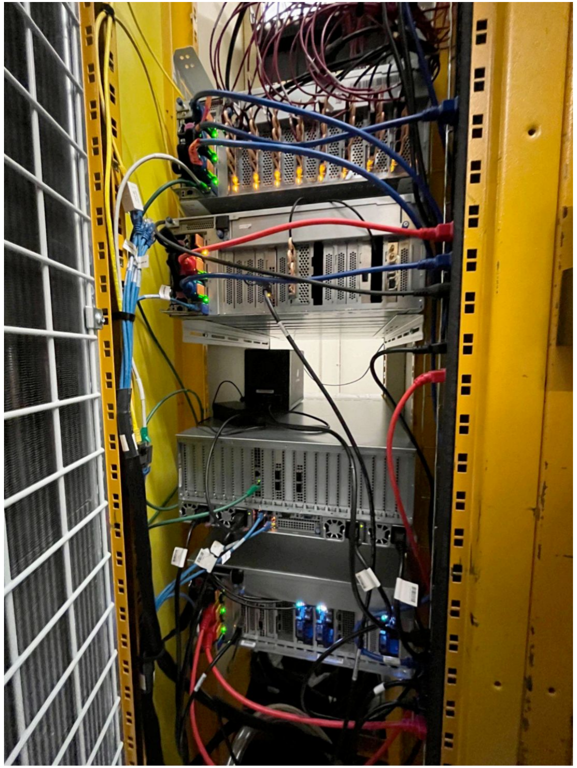
Step 3: Find the local maxima and compute centroid

$$\overline{u} = u_0 + \frac{\sum_{ij} iR_{i,j}}{\sum_{ij} R_{i,j}} \qquad \overline{v} = v_0 + \frac{\sum_{ij} jR_{i,j}}{\sum_{ij} R_{i,j}}$$

$i{=}U{-}1, U, U{+}1$
$j{=}V{-}1, V, V{+}1$

Hardware demonstrator

- A complete Retina demonstrator was installed and tested at the LHCb TestBed facility (Point 8) [10.1051/epjconf/202429502009].
- Implemented on 8 PCIe-hosted FPGA cards.
- Reconstructs a quadrant of the LHCb Vertex Locator (VELO).
  - Scalable to the whole detector by adding more FPGA cards.
- Working on:
  - LHCb live data.
  - LHCb MC data:
    - Nominal luminosity (2x10^{33} cm^{-2}s^{-1}).
    - Longest continuous run: 27 days (no error detected).
    - Event rate: 19.6 MHz.
    - Power consumption: 550 W.

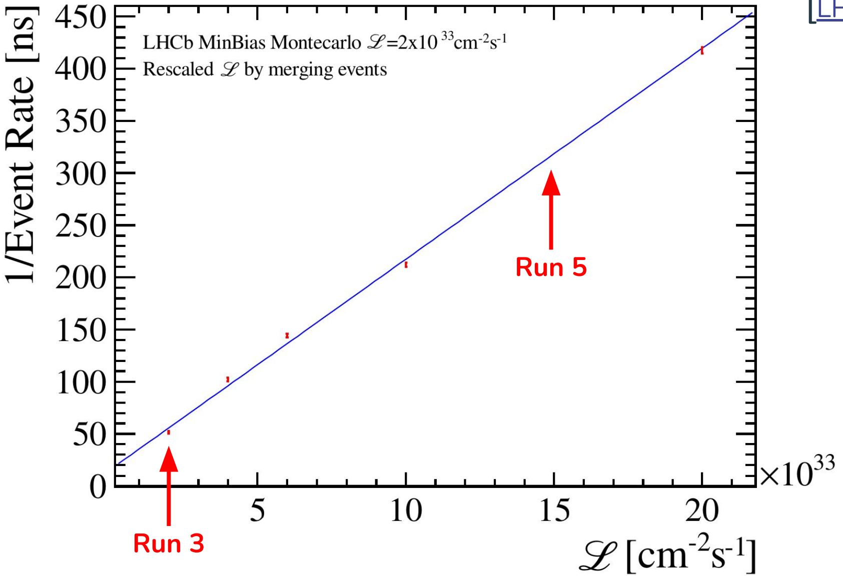


Throughput scaling

- We can emulate higher luminosities condition merging events at lower luminosity.

Measured on hardware demonstrator

[LHCb-PUB-2024-001]



- Performance **scales linearly** up to very high luminosities.
- How can we run at high luminosities keeping the required event rate (30 MHz)?

The “artificial retina” complexity

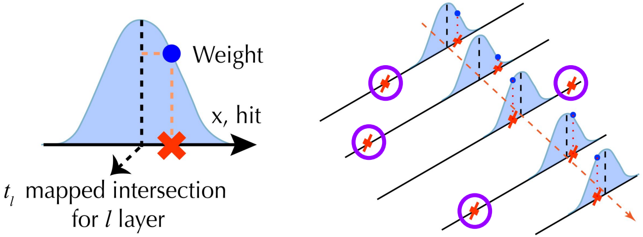
Processing time **can** depend on the number of cells:

- **Hits distant** from the mapped track have a **null weight**.
  → These hits can be delivered only to certain cells.

- The “artificial retina” architecture includes by default a custom switch to do that.
  ○ Hits from specific regions of the detector are routed only to a subset of cells.
  → Each cell processes only hits **near the reference track**.

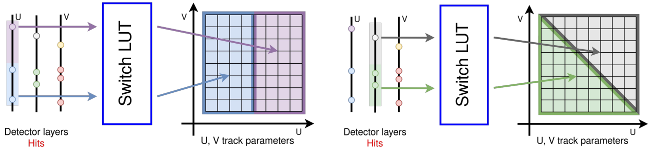
- We can increase cell density of the parameter space.
  → More cells (more reference tracks)
  → each cell covers less parameter space
  → less hits processed by a cell
  → higher speed

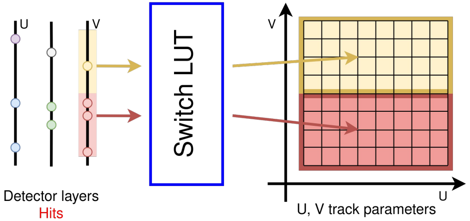
Step 2: Accumulating weights (each cell)



$$R = \sum^{\text{hits}} e^{-\frac{(x_l - t_l)^2}{2\sigma}}$$

R is close to N (# of layers) only if we have a set of hits near the mapped track

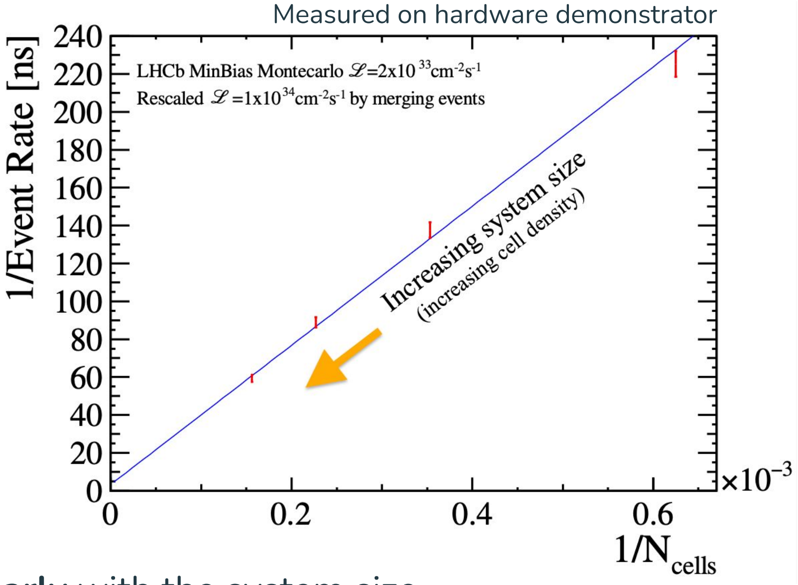




Throughput scaling

- We can emulate a bigger system by increasing the cell density of the demonstrator.

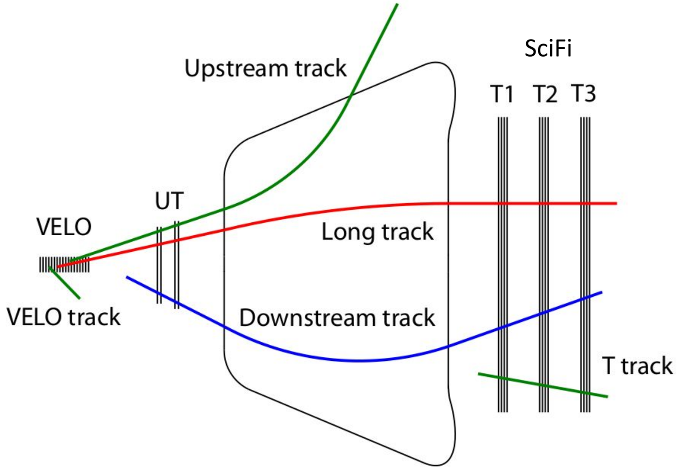
Measured on hardware demonstrator

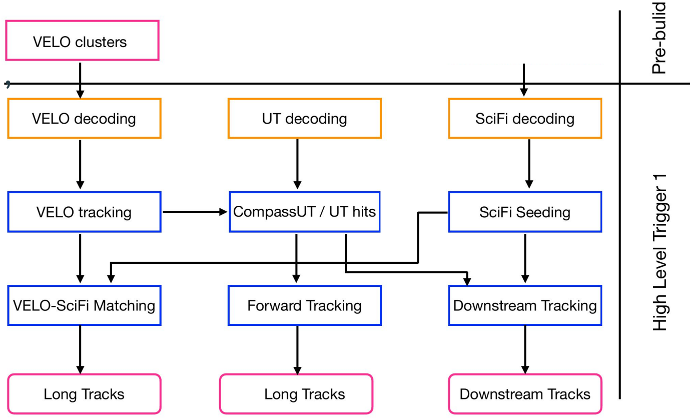


- Performance **scales linearly** with the system size.
  - → We can maintain the system throughput at high luminosity.
- What can we do to improve the LHCb event reconstruction?

Tracking at LHCb [Jiahui talk Tue Track 2]

- Velo tracks: hits on the VELO.
- T tracks: hits on the SciFi.
- Long tracks: hits on at VELO-(UT)-SciFi.
  - The most used in analysis.
- Downstream tracks: hits on UT and SciFi.
  - Most interesting for studying:
    Neutral kaons and lambdas ($D^0 \to K_S K_S$, $K_S \to \mu\,\mu$, etc.),
    Lifetime-unbiased $D^0 \to K_S \pi\,\pi$,
    Exotics LLPs.





VELO clusters

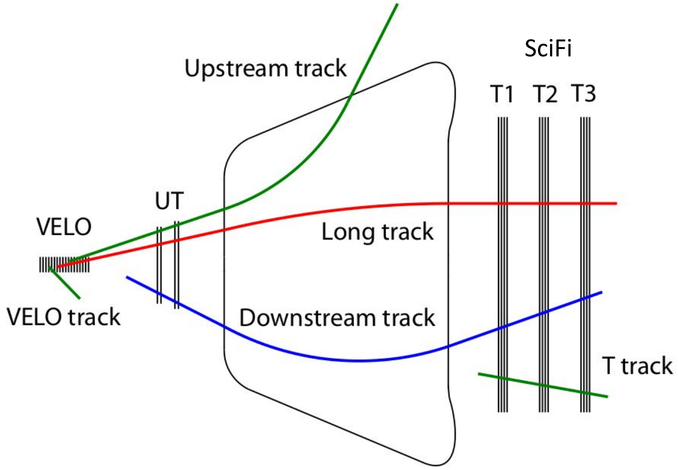
VELO decoding → VELO tracking → VELO-SciFi Matching → Long Tracks

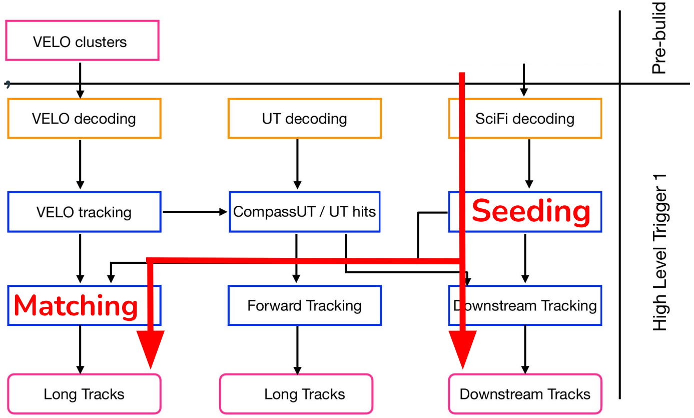
UT decoding → CompassUT / UT hits → Forward Tracking → Long Tracks

SciFi decoding → SciFi Seeding → Downstream Tracking → Downstream Tracks

Tracking at LHCb [Jiahui talk Tue Track 2]

- Velo tracks: hits on the VELO.
- T tracks: hits on the SciFi.
- Long tracks: hits on at VELO-(UT)-SciFi.
  - The most used in analysis.
- Downstream tracks: hits on UT and SciFi.
  - Most interesting for studying:
    Neutral kaons and lambdas ($D^0 \rightarrow K_S K_S$, $K_S \rightarrow \mu\,\mu$, etc.),
    Lifetime-unbiased $D^0 \rightarrow K_S \pi\,\pi$,
    Exotics LLPs.
- Downstream tracks are reconstructed starting from T tracks.
- Long tracks can be reconstructed starting from T tracks.





VELO clusters

VELO decoding

UT decoding

SciFi decoding

VELO tracking

CompassUT / UT hits

Seeding

Matching

Forward Tracking

Downstream Tracking

Long Tracks

Long Tracks

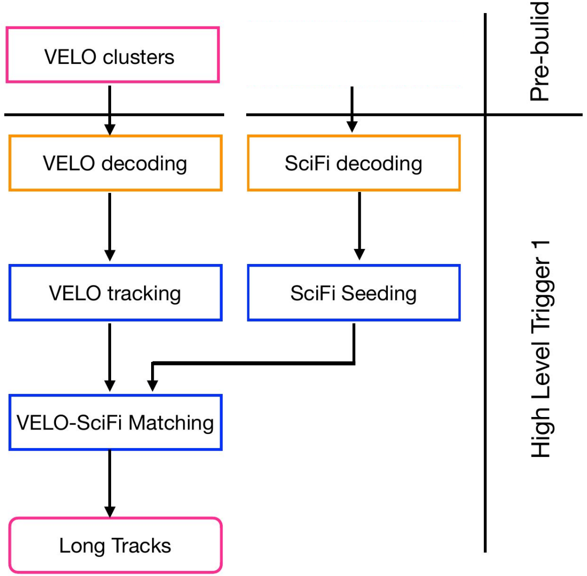
Downstream Tracks

Pre-bulid

High Level Trigger 1

The matching sequence

- Long tracks by matching VELO tracks and T tracks.
- One of the possible HLT1 reconstruction sequence at LHCb.
- Execution time:
  - Total: **7.2 µs**
  - Seeding: **1.5 µs**



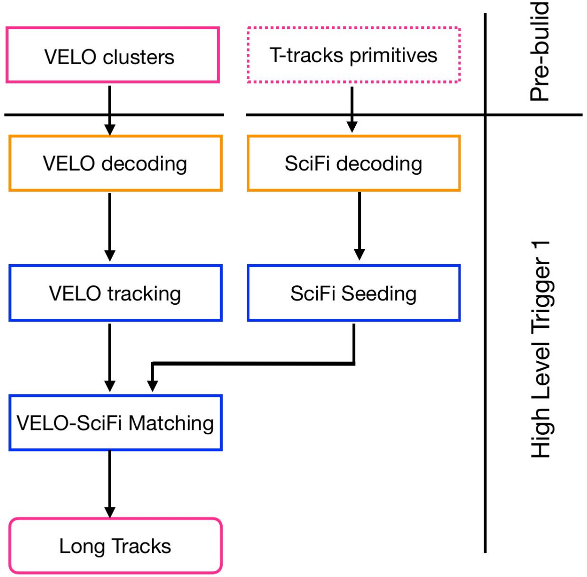
The matching sequence

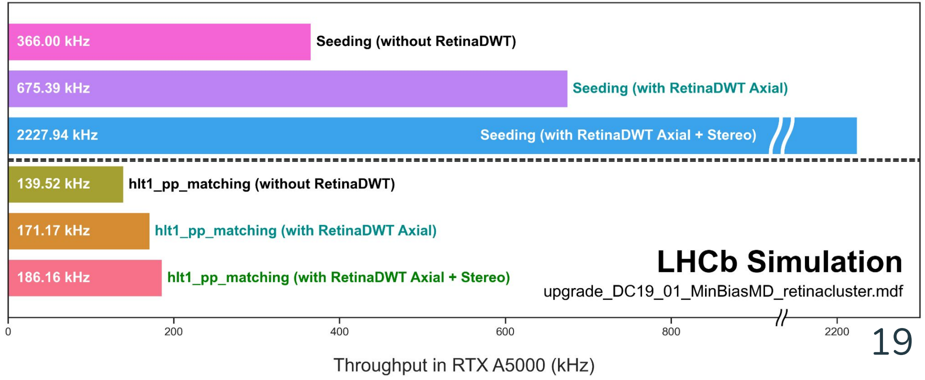
- Long tracks by matching VELO tracks and T tracks.
- One of the possible HLT1 reconstruction sequence at LHCb.
- Execution time:
  - Total: **7.2 µs**
  - Seeding: **1.5 µs**

What if T tracks primitives were available?

- Replace seeding with primitive decoding and refitting.
- Execution time:
  - Total: **5.4 µs**
  - New algorithms: **0.06 µs**
- New algorithms add a small overhead.
- **Saved more time than replaced**:

  (7.2 – 5.4) µs = 1.8 µs > 1.5 µs.

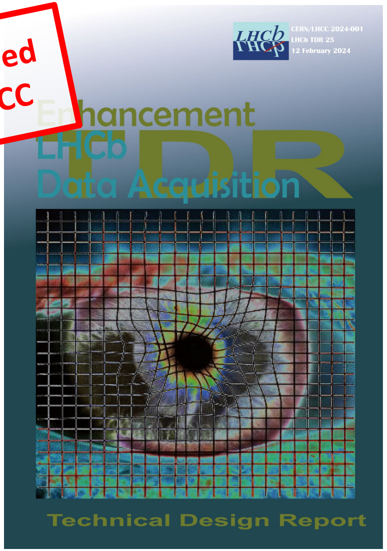


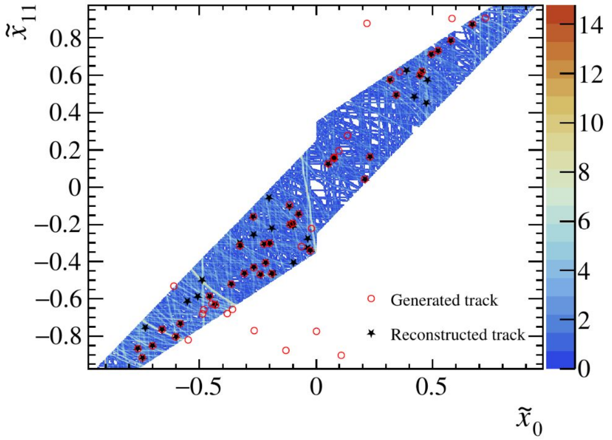


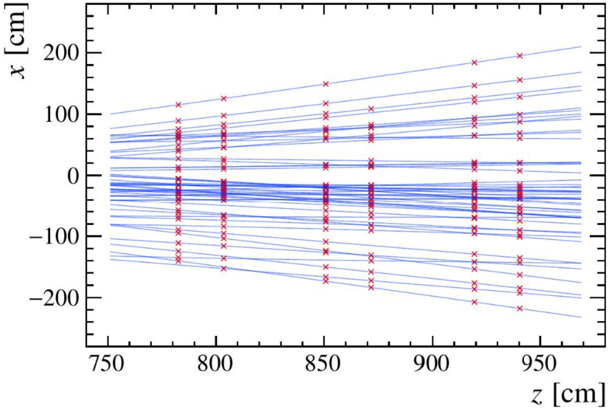
The Downstream Tracker

- LHCb plans to build a device (DWT) for reconstructing T track primitives using the “artificial retina” architecture [LHCB-TDR-025].
- Available also a detailed public note [LHCb-PUB-2024-001].
- Requires ~100 FPGAs boards (new LHCb readout boards).
- DWT will take data in Run 4.

Approved by LHCC







Summary

- In the future HEP experiment have to process more data and more complex.
- Pre-process data near the detector allows to save processing power and network resources.
- The “artificial retina” is a highly-parallel architecture for pattern recognition.
- Its complexity is intrinsically O(n).
  → Particularly interesting for LHCb Run 5.
- LHCb planned to build for Run 4 a device for reconstructing T track primitives using this architecture.
- If included in default sequence, HLT1 throughput increased by 33% (matching sequence).

- Experience gained with this new technology will be precious in studying possible applications to the challenging environment of LHCb-U2.

Backup

Introduction

What are primitives?

- ***Primitives*** is not something new at LHCb.
  - Object produced from raw data, required to produce higher level object.

    E.g. Active channels → **SciFi hits (clusters)** → tracks

    SciFi hits → **T-tracks** → Long/Downstream tracks

- Evaluated **during readout** and included in **raw event**.
  - Can be used to accelerate both HLT1 and HLT2.
  - Possibility to also drop some raw data → reduce B/W needs.

- We are talking about producing more complex *primitives* bringing forward the first stage of tracking.
  - E.g. Clusters → **sets of aligned hits** → tracks

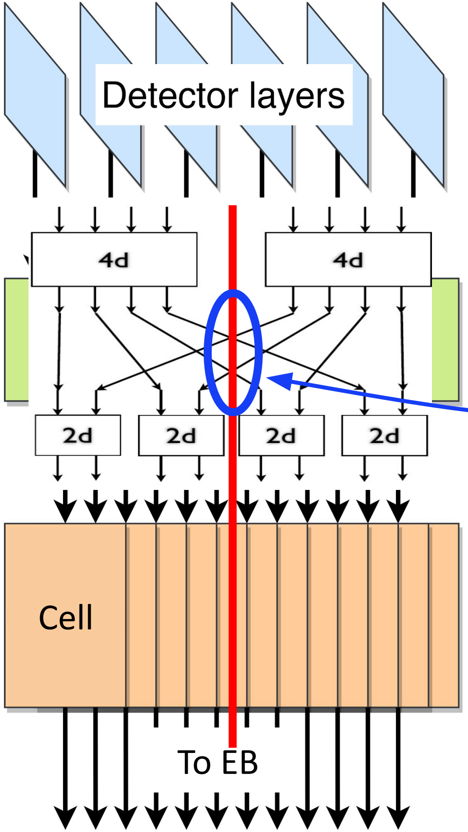
- **HLT completes the reconstruction** starting from pre-processed data.
  - *Primitives* can still be refined to increase quality.
  - Load balance between the two systems can be optimized according to needs, exploiting the strengths of each architecture.

Benefits of embedded primitives

- Hits in the VELO detector of LHCb appear as 2D clusters of pixels.
- In Run 3, firmware deployed in FPGA to make clusters on the fly [10.1109/TNS.2023.3273600].
- Uses spare resources in DAQ boards → No extra hardware.
- Raw pixel information dropped and replaced by hit positions during readout → saves 14% of b/w
- FPGA implementation saves 11% of HLT1 computing power.
- Uses 1/50th of the electrical power required by HLT1 for the same task (130 W vs 6 kW).

The “artificial retina”

A modular design



Input from detector and data preparation.

Distribution network:

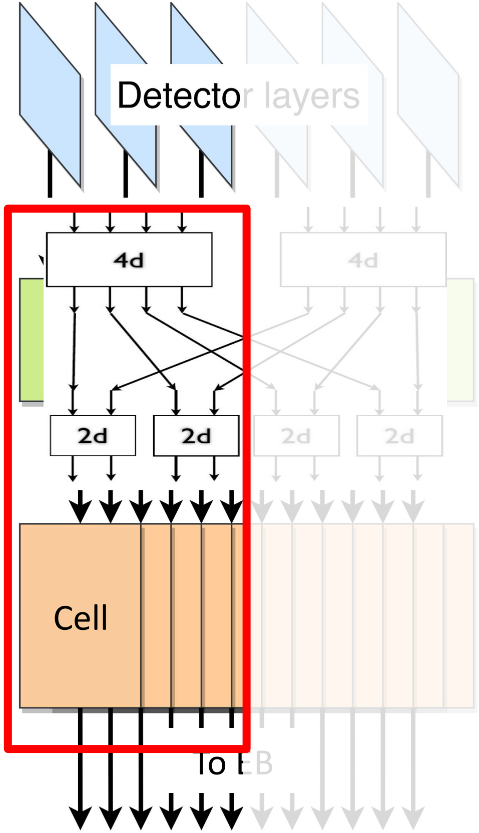
- Switch: routes hits only to appropriate cells using lookup tables.
- Optical communication: exchanges hits between boards.

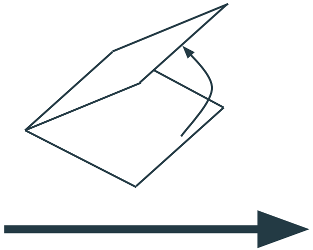
Cell:

- Engine: computes and accumulates hit weights.
- Max-finder: finds tracks (local maxima)

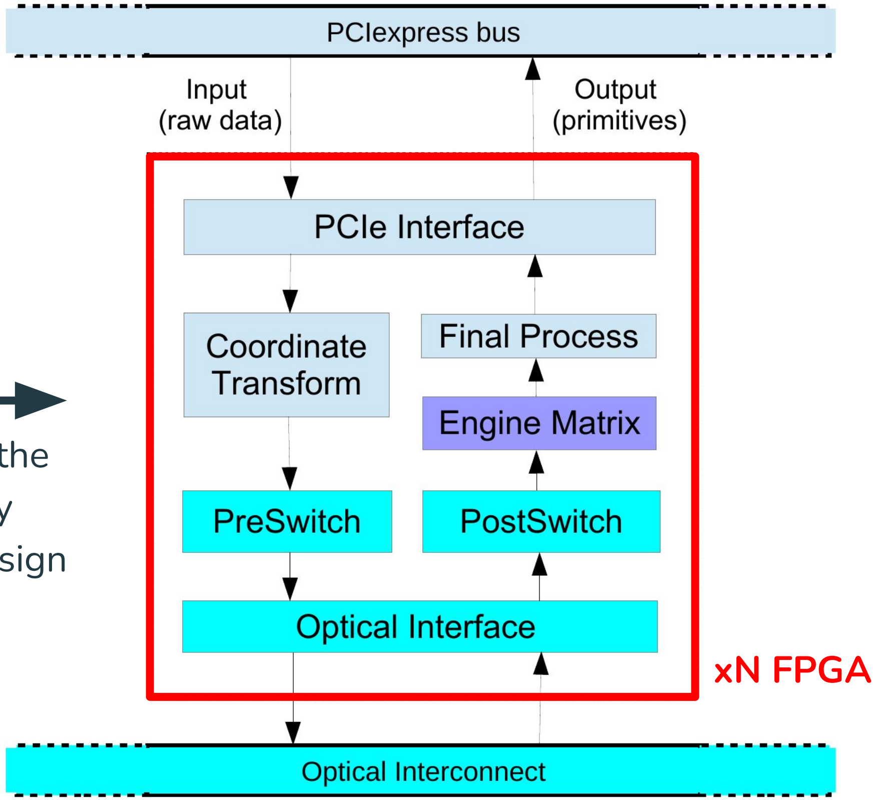
Primitive tracks are forwarded to the Event Builder.

Physical implementation



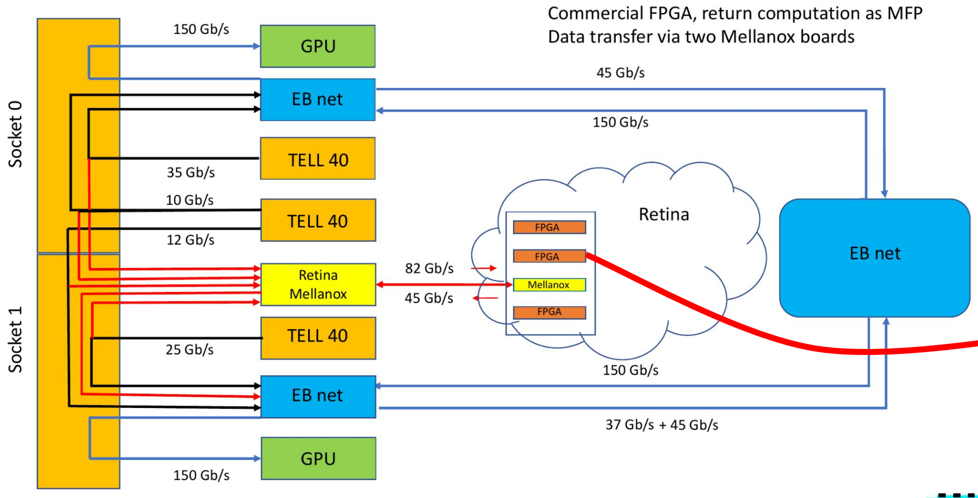


Embedded in the same FPGA by folding the design

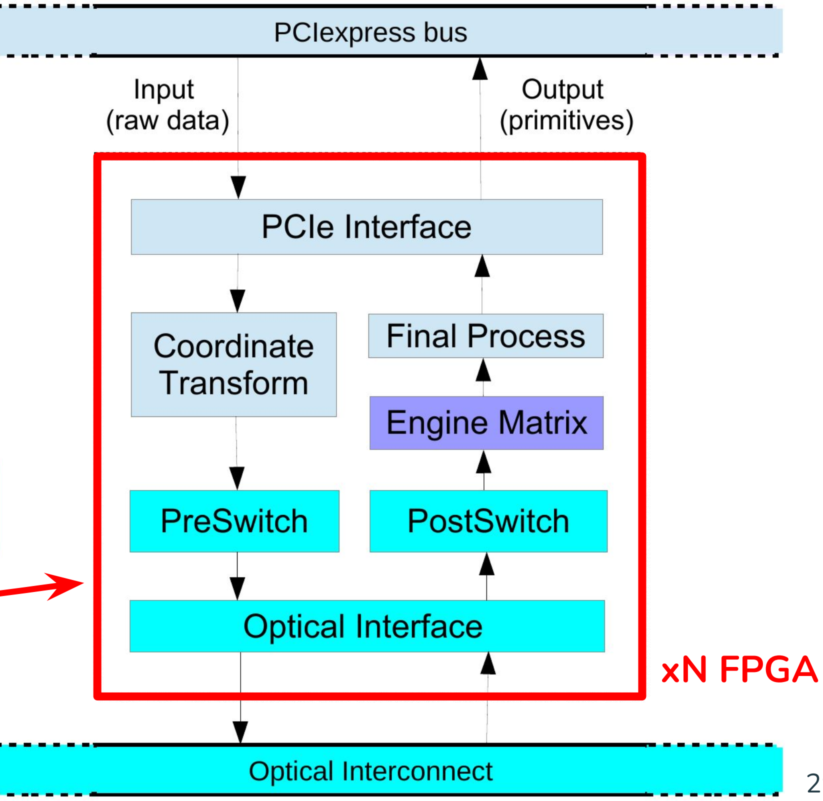


Physical implementation

- FPGA mounted on external boxes connected to SciFi EB nodes.
- In a future scenario could be implemented inside readout boards.

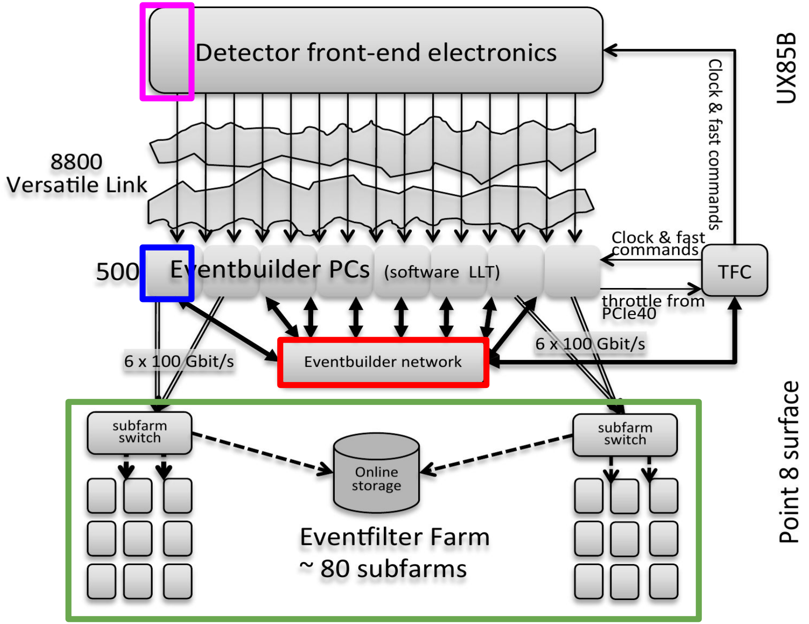


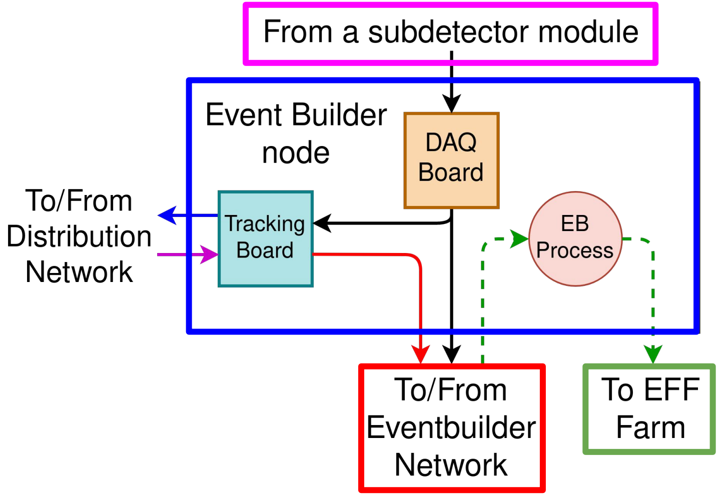
Commercial FPGA, return computation as MFP
Data transfer via two Mellanox boards



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.



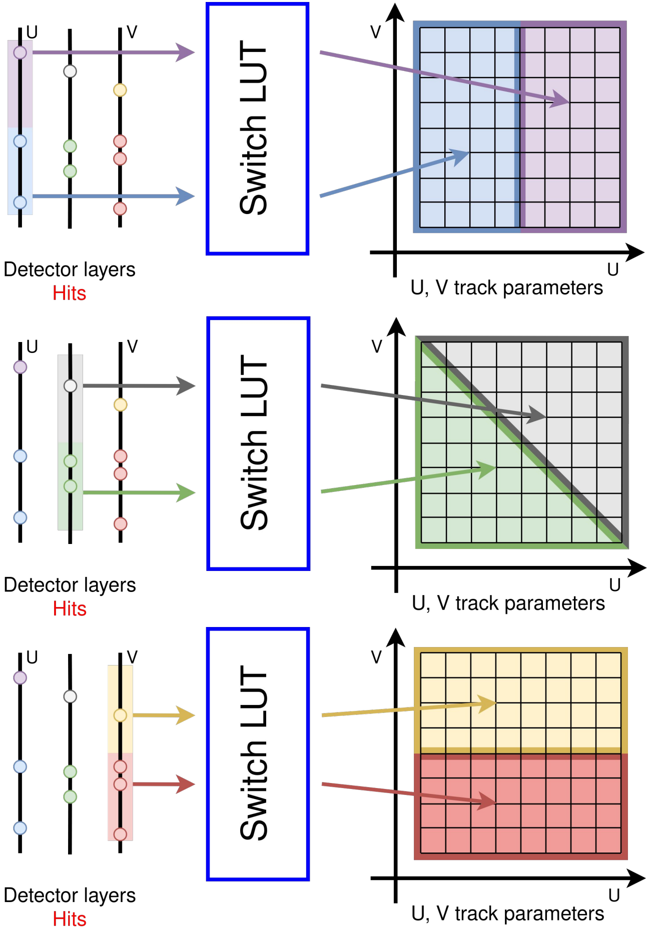


From a subdetector module
Event Builder node
DAQ Board
To/From Distribution Network
Tracking Board
EB Process
To/From Eventbuilder Network
To EFF Farm

→ Data from a subdetector module, all events
→ Subset of data from a subdetector module, all events
→ Subset of data from a subdetector, all events
→ Subset of tracks in a subdetector, all events
⇢ Data and tracks of the entire detector, some events

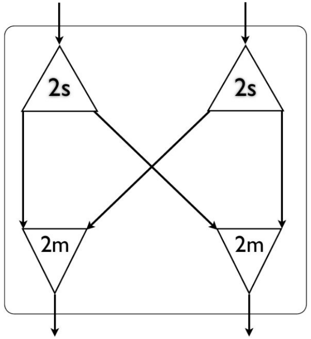
The Distribution Network

- 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.

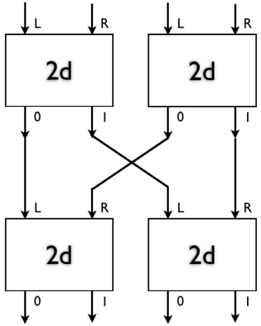


Switch

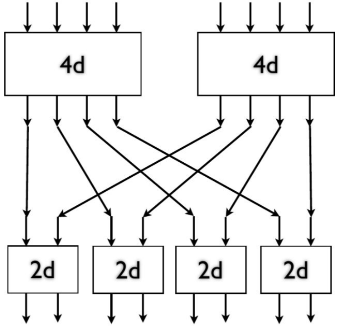
- 2-way dispatcher (2d): 2 splitters (1 input - 2 outputs) and 2 mergers (2 inputs - 1 output).
- Combining 2-way dispatchers is possible to build a switch with the desired number of lanes:
  - Switch with $N = 2^n$ lanes requires M 2-way dispatchers: $\begin{cases} M(0) = 0 \\ M(n) = 2M(n-1) + 2^{n-1}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



2-way dispatcher (2d)



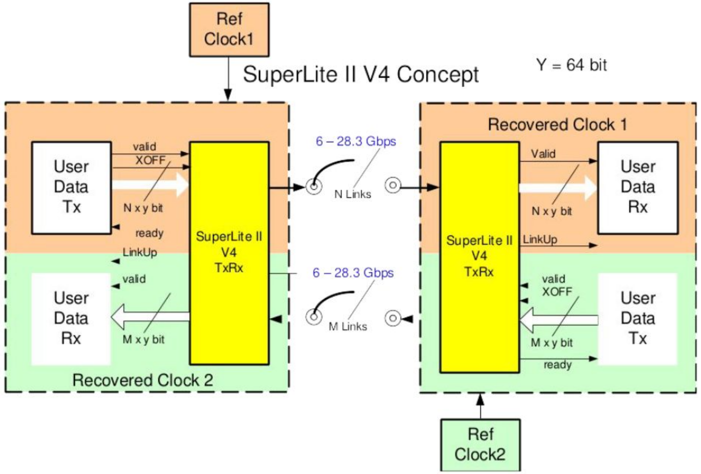
4-way dispatcher (4d)



8-way dispatcher (8d)

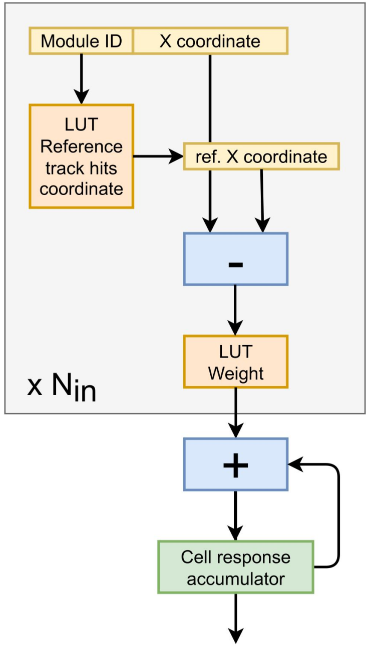
Optical communication

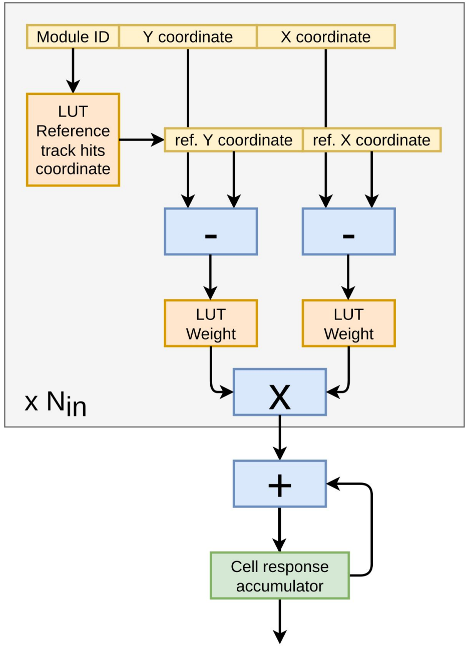
- Uses Intel SuperLite II v4 communication protocol.
  - Fully free and available in source code.
  - Supports flow control.
  - Can be used to connect various FPGA families (already available on A10, S10, Agilex).
- Design adapted to implement the desired number of independent links.
- Extensively tested:
  - Long run: up to 2 months.
  - High-speed: up to 26 Gbps.
  - Multiple boards: up to 5 boards.
  - Large patch-panel: up to 64 links.



Engines

- Accepts 1D- and 2D-hits.
- Multiple inputs (N_{in} = 4) for accepting up to 4 hits per clock cycle.

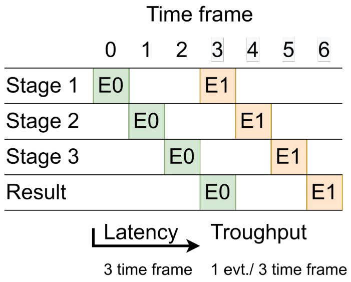




The firmware paradigms

Pipeline:

- Like an assembly line, an event is processed as soon as possible, without waiting for the previous one to go through all the steps.
- This paradigm is extended to the hit level → 1 hit/clk cycle.

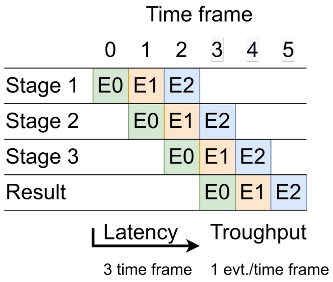


Time frame

	0	1	2	3	4	5	6
Stage 1	E0			E1			
Stage 2		E0			E1		
Stage 3			E0			E1	
Result				E0			E1

Latency Troughput

3 time frame 1 evt./ 3 time frame



Time frame

	0	1	2	3	4	5
Stage 1	E0	E1	E2			
Stage 2		E0	E1	E2		
Stage 3			E0	E1	E2	
Result				E0	E1	E2

Latency Troughput

3 time frame 1 evt./time frame

Parallel computing:

- Hits flow through the distribution network via parallel lines.
- Cells work in a fully parallel way (both weight accumulation and maxima finding).
- Cells have also parallel inputs to process more hit per clock cycle.
- A bigger system has more parallel processor, so its throughput is similar to the one of a small system.

Modularity:

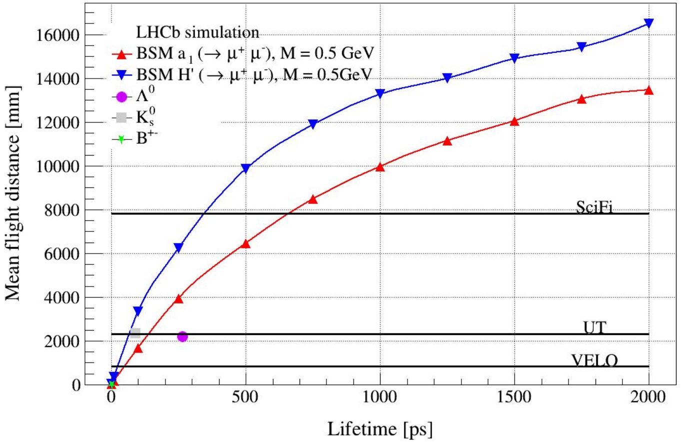
- Each component (switch, matrix of cell, ecc.) is a repetition of basic blocks.
- A bigger system is implemented instantiating more copies of the same modules.
- Modules can be freely spread over multiple devices overcoming FPGA size limitation.

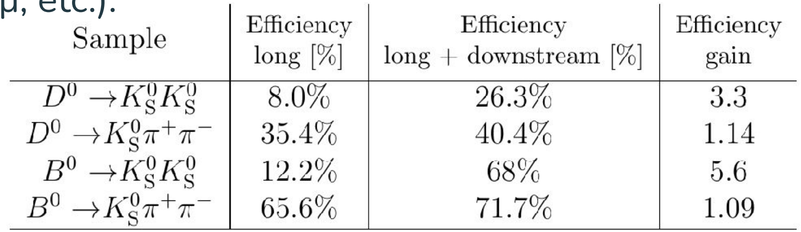
This is different from other systems that rely to time multiplexing.

The Downstream Tracker

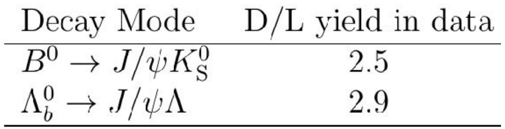
The importance of Downstream tracks

- Long tracks: hits at least on VELO and SciFi.
  - Flight distance < 1 m
  - Few LLPs reconstructible as Long tracks.
- Downstream tracks: hits on UT and SciFi.
  - Reconstructed from **T-tracks** adding UT hits.
- Triggering on Downstream tracks at HLT1 level extends the LHCb baseline physics program in interesting ways:
  - Neutral kaons and lambdas ($D^0 \rightarrow K_S\, K_S$, $K_S \rightarrow \mu\, \mu$, etc.).
  - Lifetime-unbiased trigger for $D^0 \rightarrow K_S\, \pi\, \pi$.
  - Exotics LLPs.
- Important to preserve them also at higher luminosities.





Sample	Efficiency long [%]	Efficiency long + downstream [%]	Efficiency gain
$D^0 \rightarrow K^0_S K^0_S$	8.0%	26.3%	3.3
$D^0 \rightarrow K^0_S \pi^+ \pi^-$	35.4%	40.4%	1.14
$B^0 \rightarrow K^0_S K^0_S$	12.2%	68%	5.6
$B^0 \rightarrow K^0_S \pi^+ \pi^-$	65.6%	71.7%	1.09



Decay Mode	D/L yield in data
$B^0 \rightarrow J/\psi K^0_S$	2.5
$\Lambda^0_b \rightarrow J/\psi \Lambda$	2.9

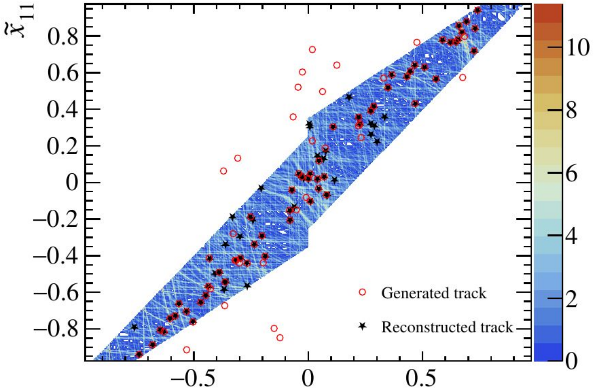
Simulation study of DWT

- Studies performed with realistic DWT Emulator.
- LHCb MC productions for Run 3.

- Reconstruction steps:
  - Axial pattern recognition (Retina).
  - Ghost removal (χ^2 fit).
  - Stereo pattern recognition (Retina).
  - Ghost removal (χ^2 fit).

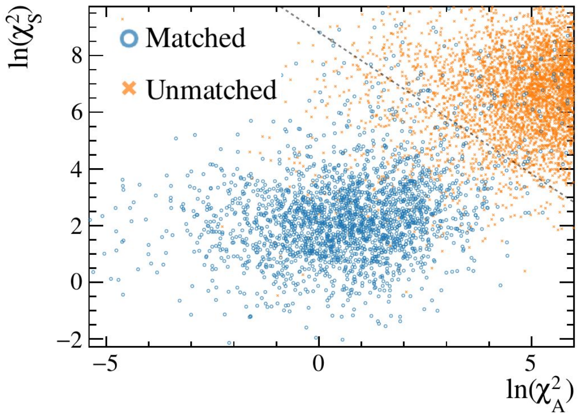
- SciFi reconstruction.
  - Axial part (*x*-*z* view): 64 FPGAs.
  - Stereo part (*y*-*z* view): 32 FPGAs.

- Track parameters:
  - *x*-coordinate on first and last layer.
  - *y*-coordinate at the middle of SciFi.
  - Extra: *x*-*z* curvature from fit.



Axial retina response

$\chi^2_A < 60$

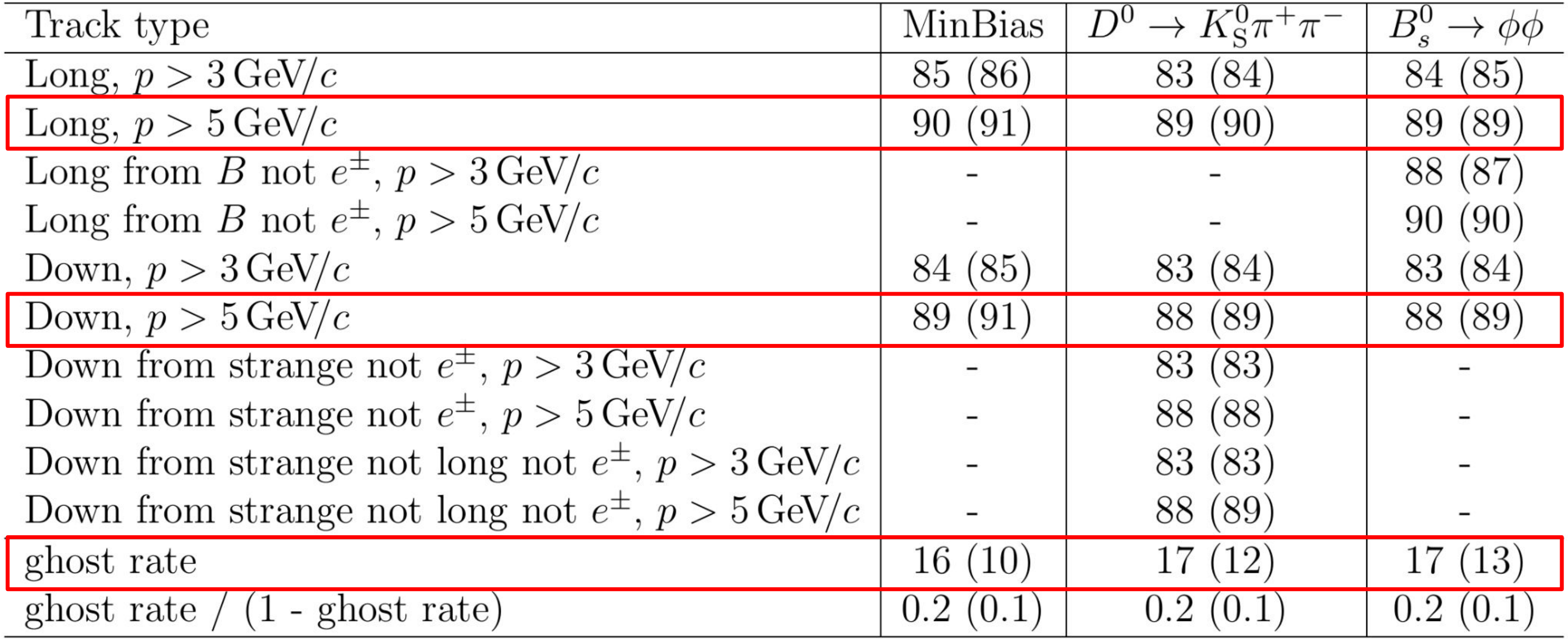


χ^2 scatter plot

DWT tracking performance

- Fiducial requirements: p_T > 200 MeV/c; $2 < \eta < 5$.

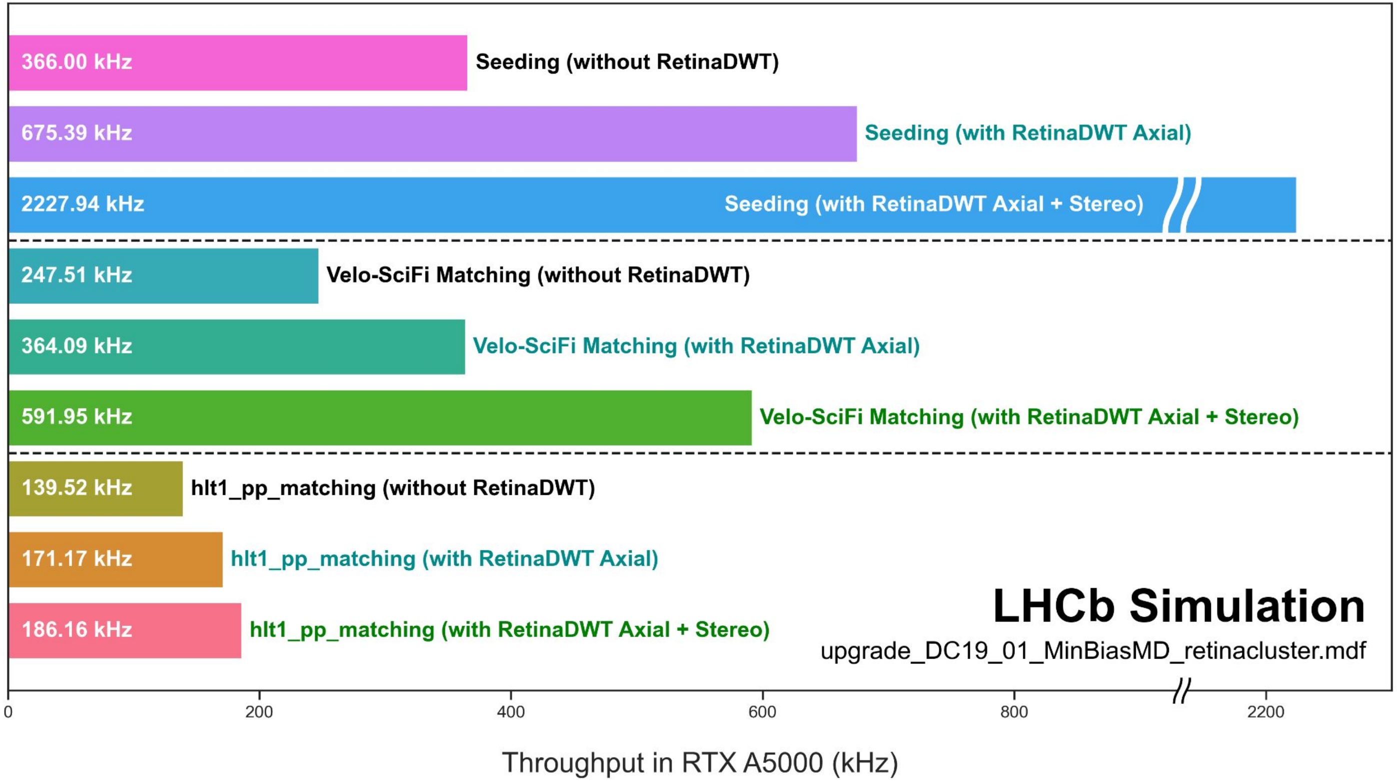
Event-averaged values in brackets



Track type	MinBias	$D^0 \to K^0_S\pi^+\pi^-$	$B^0_s \to \phi\phi$
Long, $p > 3\,\mathrm{GeV}/c$	85 (86)	83 (84)	84 (85)
Long, $p > 5\,\mathrm{GeV}/c$	90 (91)	89 (90)	89 (89)
Long from B not $e^\pm$, $p > 3\,\mathrm{GeV}/c$	-	-	88 (87)
Long from B not $e^\pm$, $p > 5\,\mathrm{GeV}/c$	-	-	90 (90)
Down, $p > 3\,\mathrm{GeV}/c$	84 (85)	83 (84)	83 (84)
Down, $p > 5\,\mathrm{GeV}/c$	89 (91)	88 (89)	88 (89)
Down from strange not $e^\pm$, $p > 3\,\mathrm{GeV}/c$	-	83 (83)	-
Down from strange not $e^\pm$, $p > 5\,\mathrm{GeV}/c$	-	88 (88)	-
Down from strange not long not $e^\pm$, $p > 3\,\mathrm{GeV}/c$	-	83 (83)	-
Down from strange not long not $e^\pm$, $p > 5\,\mathrm{GeV}/c$	-	88 (89)	-
ghost rate	16 (10)	17 (12)	17 (13)
ghost rate / (1 - ghost rate)	0.2 (0.1)	0.2 (0.1)	0.2 (0.1)

- Performance similar to current HLT1 already at the primitive level.

HLT1 Throughput


366.00 kHz
Seeding (without RetinaDWT)
675.39 kHz
Seeding (with RetinaDWT Axial)
2227.94 kHz
Seeding (with RetinaDWT Axial + Stereo)
247.51 kHz
Velo-SciFi Matching (without RetinaDWT)
364.09 kHz
Velo-SciFi Matching (with RetinaDWT Axial)
591.95 kHz
Velo-SciFi Matching (with RetinaDWT Axial + Stereo)
139.52 kHz
hlt1_pp_matching (without RetinaDWT)
171.17 kHz
hlt1_pp_matching (with RetinaDWT Axial)
186.16 kHz
hlt1_pp_matching (with RetinaDWT Axial + Stereo)
LHCb Simulation
upgrade_DC19_01_MinBiasMD_retinacluster.mdf
Throughput in RTX A5000 (kHz)