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

Federico Lazzari[1,2,*], *Andrea* Contu[3], *Riccardo* Fantechi[2], *Jibo* He[4], *Brij Kishor* Jashal[5], *Maurizio* Martinelli[6,7], *Michael J.* Morello[2,8], *Arantza* Oyanguren[9,10], *Lorenzo* Pica[2,8], *Giovanni* Punzi[1,2], *Qi* Shi[4], *Francesco* Terzuoli[2,11], *Giulia* Tuci[12], *Ao* Xu[2,8], and *Jiahui* Zhuo[9,10]

[1]Università di Pisa, Pisa, Italy
[2]INFN sezione di Pisa, Pisa, Italy
[3]INFN sezione di Cagliari, Cagliari, Italy
[4]University of Chinese Academy of Sciences, Beijing, China
[5]Rutherford Appleton Laboratory, Chilton, United Kingdom
[6]Università degli Studi di Milano-Bicocca, Milano, Italy
[7]INFN sezione di Milano, Milano, Italy
[8]Scuola Normale Superiore, Pisa, Italy
[9]Universitat de València, València, Spain
[10]Consejo Superior de Investigaciones Científicas, Madrid, Spain
[11]Università degli Studi di Siena, Siena, Italy
[12]Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany

Abstract. The LHCb collaboration is planning an upgrade (LHCb "Upgrade-II") to collect data during Run 5 at an instantaneous luminosity an order of magnitude larger than the current one (Run 3). LHCb relies on a complete real-time reconstruction of all collision events at LHC-Point 8, which will have to cope with both the luminosity increase and the introduction of correspondingly more granular and complex detectors. After an intensive R&D programme, LHCb approved a FPGA-based system to pre-reconstruct tracks in the SciFi detector at readout level during Run 4, as an intermediate step towards a system that could be extended to other tracking detectors in the future. It is based on the "artificial retina", an extremely parallel architecture. Using simulated data, the performance of a hardware demonstrator of this architecture has been tested as a function of instantaneous luminosity and system size, and was found to have $O(n)$ complexity, which is a crucial feature for high luminosity applications.

1 Introduction

LHCb addresses flavour physics at low p_T [1], which is characterised by a very high cross section with respect to Higgs and electroweak physics. Simple quantities usually available at the first trigger stage, like the energy deposition on the calorimeters or the presence of muons, do not allow to trigger efficiently. For this reason, LHCb adopted in Run 3 a full-software trigger system, called High Level Trigger (HLT), that reconstructs all collision events at LHC-Point 8 in real-time [2]. The first level (HLT1) runs on 489 Nvidia RTX A5000 GPUs, the second level (HLT2) runs on $O(100k)$ CPU cores. For LHC's Run 5, the LHCb collaboration

*e-mail: federico.lazzari@cern.ch

is planning an upgrade (LHCb Upgrade-II) to collect data at the instantaneous luminosity up to a maximum of $\mathcal{L} = 1.5\times10^{34}$ cm^{-2} s^{-1}, a factor 7.5 times larger than the current one. Since the event reconstruction time typically scales quadratically with the luminosity, the required computing power would be naively estimated to be a factor of 50 larger than the current one. This need drives LHCb to keep searching for new processing solutions.

A possible solution is to start event reconstruction before full events are built, using information available locally on the sub-detectors. The data structures produced at this stage ("primitives") can be formatted as data from a sub-detector and embedded in the raw detector data by the readout system. Then the primitives can be used as a starting point by HLT for event reconstruction, off-loading it of some processing tasks. To some extent, this strategy has already been used in both LHCb [3] and other experiments [4], the innovation is to extend this strategy to a task that requires to exchange information between all the readout boards of a sub-detector and persist the primitives up to the other processing levels, making this a major part of the event reconstruction process. However, with the scaling of the luminosity, also the computing power required by the pre-processor scales. Therefore, this paradigm shows a major advantage when moving a task to an earlier stage allows to reduce the time complexity of the task itself. In the present work, we describe why a system based on the "artificial retina" scales linearly with luminosity.

2 The "artificial retina"

The "artificial retina" [5] is a highly-parallel computing architecture aimed at fast finding of combinations of input data that are compatible with a set of pre-calculated reference patterns. It is an arrangement of many computing units (cells) fed by a custom distribution network. From a mathematical point of view, the operations performed by the cells resemble the "Hough transform" [6]. However, the most distinctive feature of this approach lies in the specific arrangement of its elements, that takes full advantage of the features of modern FPGAs, like parallel computing capabilities, low latencies, high-bandwidth transceivers (XCVRs).

As a preliminary step, the track space to be covered is discretised in a regular grid of reference track (Fig. 1 left). For each reference track is computed its interceptions with the detector layers (receptors), and a cell is implemented. The cells receive hits as input and check whether they are compatible with the assigned reference track. To this purpose, a computational engine assigns a weight to every received hit, that is determined by its distance from the corresponding receptor. The weight function is conveniently truncated to zero at some distance from the receptor ("search distance"). The weights of all received hits are accumulated into a single value, whose final value at the end of the event represents the "excitation level" of the cell (Fig. 1 centre). A high value indicates good compatibility of the set of received hits to the reference track. After all cells have received all relevant hits from the event at hand, the cells compare their excitation level with the one of the neighbour cells, verifying if they are a local maxima. Local maxima over a threshold value are taken as track candidates, with the position of their centroid within the matrix taken as an estimate of the candidate track's parameters (Fig. 1 right).

Even if a cell requires a relatively small amount of logic resources, the largest available FPGA does not have enough logic resources to accommodate a Retina device covering an entire LHCb sub-detector, therefore the implementation is realized in practice distributing the cells over several FPGAs. A distributed system can retrieve data more easily than a monolithic or a time-multiplexed one. Even a small sub-detector requires multiple readout boards. If the reconstruction system requires to collect all the data related to an event in a single device, it needs a second system to merge that data in a single data structure (event

building). Moreover the single processing device must have enough bandwidth to receive it. Instead, each Retina FPGA can be connected to a different readout board. In this way, each chip receives the portion of the event read by the corresponding readout board. The Retina architecture includes a custom distribution network that exchanges hits between the FPGAs through their XCVRs and, being programmed with which hits are required in which cells, delivers to each cell only the hits within their search distance.

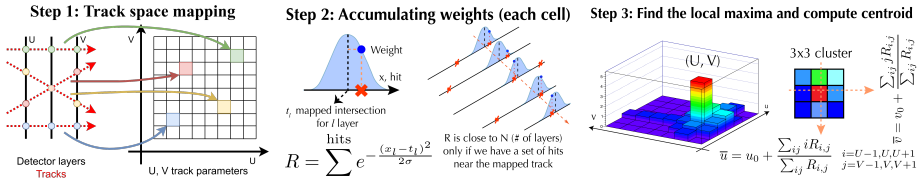

Figure 1. Track reconstruction steps with the "artificial retina" architecture [7].

The hits distribution and the reconstruction steps are executed in a pipeline, so that all the different parts of the device are always active. Each step is executed using the output of the previous one as input, but the two can be performed at the same time on different parts of the data, possibly on different events. The same paradigm is also adopted within the single firmware components, according to a full dataflow protocol. This ensures optimal exploitation of the hardware for maximum throughput.

3 Time complexity

The time complexity of the system can be determined by evaluating the time complexity of the single processing steps. The first step (track space mapping) is a configuration stage not performed at runtime, therefore it does not enter in the time complexity computation.

The weight accumulation step is performed in the cells. Each cell has its own computational engine, therefore the processing time does not depend directly on the number of cells. The computational engine performs only operations on the single hits: measurement of the distance between the hit and the receptor, computation of the weight, and sum of the weight to the excitation level. Not performing operation over hits combination, and processing the hits in a pipeline, the execution time of this stage scales linearly with the number of hits.

Finally the search of local maxima and the centroid computation are performed in parallel by each cell. Again the processing time does not depend on the number of cells or the number of track candidates. Globally the Retina complexity is $O(n)$. This has been verified with a test on a real device.

LHCb established the Coprocessor TestBed facility to test, during the Run 3, new processing solutions for Upgrade-II in realistic DAQ conditions. Within this initiative we built the Retina demonstrator [8]. It implements the complete functionality expected from a future device.

Our demonstrator system includes 16 modules of the right-hand side of the VELO [9]. The VELO detects charged particles in the region closest to the interaction point, aiming at reconstructing primary and secondary vertexes with a spatial resolution smaller than typical decay lengths of b- and c-hadrons in LHCb ($c\tau \sim 0.01 - 1$ cm), in order to discriminate between them. The choice of the VELO for our demonstrator was motivated by several reasons: it is a complex, high resolution detector that is crucial to the experiment, and comes first in the reconstruction sequence; detailed simulations of Retina reconstruction of the VELO were

already available at the time of starting the demonstrator project [10]; its data are read out over a comparatively smaller number of lines in comparison to other LHCb sub-detectors. This allows us to build a demonstrator covering a meaningful portion of the target detector with an affordable quantity of hardware: using just 8 FPGA boards the demonstrator covers a region corresponding to about a quadrant of VELO tracking space. The whole system fits within a single server, and was installed and tested within the LHCb Coprocessor TestBed facility. We used commercially available boards, each carrying one Stratix 10 FPGA device (1SG280HN2), produced by Bittware/Molex and commercialised under the name 520N [11].

The demonstrator can be configured to process live data arriving from LHCb's monitoring farm at a pre-scaled rate, or events preloaded in the FPGA RAM at full speed. The correctness of the demonstrator reconstruction has been validated by extensive tests, showing perfect bit-by-bit matching between its output and the tracks generated by a C++ emulation. Then demonstrator ran on real data without errors for long uninterrupted periods during LHCb physics data-taking in mid July and September 2023 [8].

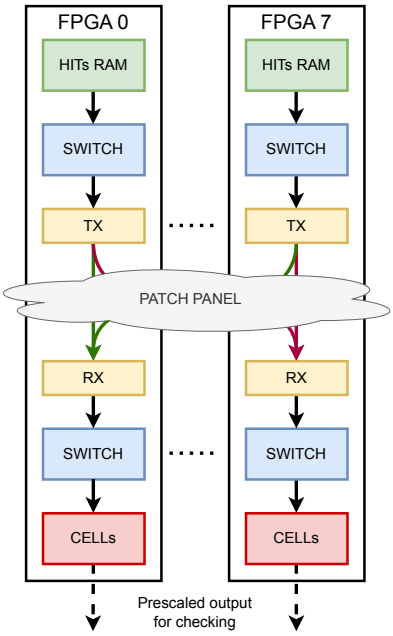

Figure 2. Structure of the demonstrator's firmware with RAMs for events preload.

Figure 2 shows the organisation of the demonstrator firmware with RAMs for events preload. Each FPGA stores hits from two VELO modules in an internal RAM. Hits are read in-loop to provide a continuous flux of data. The distribution network is composed of the switches, the transmitting (TX) and receiving (RX) units of the FPGA XCVRs, and an optical patch panel. The switches are programmed to route the hits to the appropriate cells. The XCVRs and the optical patch panel allow hits exchange between the FPGAs. Finally the cells perform the track reconstruction as already explained.

For the purpose of verifying Retina complexity, we tested the throughput of the hardware demonstrator on simulated data samples of increasing luminosities. The demonstrator

was made to process bunches of events generated by the official LHCb simulation [1, 12] at Run 3 conditions: centre of mass energy $\sqrt{s}$ = 13.6 TeV and instantaneous luminosity $\mathcal{L} = 2 \times 10^{33}$ cm^{-2} s^{-1}. To reproduce the condition at higher luminosities, we merged the hit population of multiple events into a single one, and shuffled the hits within a merged event in order to properly mix the originally separated events. We created five statistically independent sets of events, at a luminosity increased by a factor x1, x2, x3, x5, x10 with respect to the Run 3 one. The instantaneous luminosity in Run 5 will be up to $\mathcal{L} = 1.5 \times 10^{34}$ cm^{-2} s^{-1}, a factor 7.5 times larger than the Run 3 one.

Figure 3 shows that the inverse of the demonstrator event rate scales linearly with the instantaneous luminosity, confirming with a test on a real Retina system that the time complexity of the Retina architecture is $O(n)$.

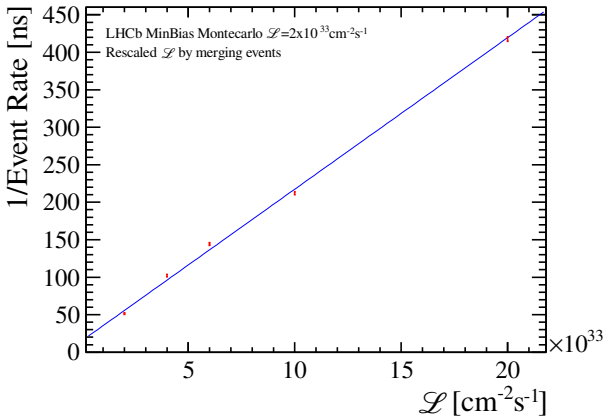

Figure 3. Inverse of the demonstrator event rate versus instantaneous luminosity [7]. Simulated data, merging multiple events to reproduce higher luminosities.

4 Scaling with system size

In a time-multiplexed system, a process performs the reconstruction of an event in parallel with other processes working on different events. Therefore, it is possible to increase the throughput of the system increasing the number of workers, i.e. increasing the size of the system. In the Retina architecture, each FPGA processes a portion of each event. How the event rate of a system based on this architecture scales with its size was never documented.

The size of a Retina system is determined by the number of cells. Increasing the number of cells, without increasing the size of the track space, leads to an increased cell density. Consequently, the cell search distance must be reduced to reduce the size of the track space covered by the cell and avoid an overlap with the region covered by the other cells. However, the distribution network is programmed to send to each cell only the hits within its search distance. Therefore, each cell will receive less hits and will require less time to process an event.

From geometrical consideration over the number of hits that fall within the cell search distance, we expect a linear scaling of the processing time with the inverse of the number of cells. Like in the previous section, we can verify this with the Retina demonstrator.

Due to the limited number of boards of the demonstrator, we cannot directly increase the number of cells (the demonstrator has 1600 cells). We can still increase the cell density

covering a smaller track space and calculate the number of required cells to cover the previously mapped track space. We loaded in the FPGA RAM the data sample rescaled to the instantaneous luminosity of $\mathcal{L} = 2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ merging 5 events. Then programmed the distribution network and the cells to reproduce the behaviour of a system with 1600, 2844, 4444, and 6400 cells.

Figure 4 shows that the inverse of the demonstrator event rate scales linearly with the inverse of the number of cells (size of the covered track space covered by a cell), confirming that processing time of a system based on the Retina architecture scales linearly with the dimension of the cells.

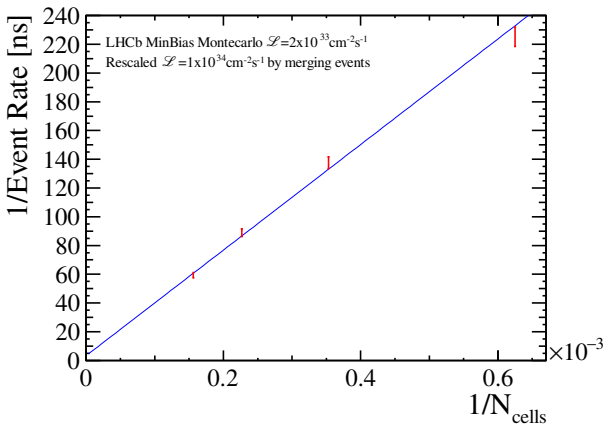

Figure 4. Inverse of the demonstrator event rate versus the inverse of the number of cells. Data from MC production, merging multiple events to reproduce higher luminosities.

5 Conclusion

The Retina demonstrator, installed at the LHCb Coprocessor TestBed facility, represents an ideal test-bench for confirming the features of the Retina architecture. In particular we demonstrated that the time complexity of this architecture is $O(n)$ and that the processing time scales linearly with the inverse of the number of cells. These two features make this technology an attractive solution in future HEP experiments, where the instantaneous luminosity will be an order of magnitude higher, with respect to more traditional solutions that usually have worse time complexity.

Current HEP experiments can also take advantage of this architecture. After an intensive R&D programme, LHCb approved a system based on the Retina architecture, called Downstream Tracker (DWT), to reconstruct track primitives in the SciFi detector during Run 4 [13]. This device represents an intermediate step towards a system that could be extended to other tracking detectors in LHCb Upgrade-II. The SciFi track primitives will be added to the raw data. Starting from pre-processed data allows the HLT to save processing time, with corresponding benefits in throughput. In the case of the DWT, HLT does not need to run the algorithm performing standalone track search in the SciFi, that is a complex and heavy pattern recognition task. In a specific reconstruction sequence of the first trigger level (HLT1), where the tracks in the SciFi play a crucial role, we measured an increase of throughput of 33% [7].

Acknowledgements

We gratefully acknowledge R&D funding received from INFN, the coordination provided by the LHCb Real Time Analysis project, and support from the LHCb Online group at the Coprocessor TestBed facility.

References

[1] LHCb Collaboration, The LHCb Upgrade I. Journal of Instrumentation **19**, P05065 (2024). https://doi.org/10.1088/1748-0221/19/05/P05065

[2] LHCb Collaboration, *LHCb Trigger and Online Upgrade Technical Design Report* (CERN, Geneva, 2014). http://dx.doi.org/10.17181/CERN.5F5X.FDJM

[3] G. Bassi et al., A FPGA-Based Architecture for Real-Time Cluster Finding in the LHCb Silicon Pixel Detector. IEEE Transactions on Nuclear Science **70, 6**, 1189–1201 (2023). https://doi.org/10.1109/TNS.2023.3273600

[4] W. Adam et al., Beam test performance of prototype silicon detectors for the Outer Tracker for the Phase-2 Upgrade of CMS. JINST **15**, P03014 (2020). https://doi.org/10.1088/1748-0221/15/03/P03014

[5] L. Ristori, An artificial retina for fast track finding. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **453, 1**, 425–429 (2000). https://doi.org/10.1016/S0168-9002(00)00676-8

[6] P. Hough, Machine Analysis of Bubble Chamber Pictures. Proc. Int. Conf. High Energy Accelerators and Instrumentation **C590914**, 554–558 (1959). https://inspirehep.net/literature/919922

[7] W. Baldini et al., *Proposal for FPGA-based tracking in the LHCb downstream region* (CERN, Geneva, 2014). https://cds.cern.ch/record/2888549

[8] F. Lazzari et al., Demonstration of track reconstruction with FPGAs on live data at LHCb. EPJ Web Conferences **295**, 02009 (2024). https://doi.org/10.1051/epjconf/202429502009

[9] LHCb Collaboration, *LHCb VELO Upgrade Technical Design Report* (CERN, Geneva, 2013). http://dx.doi.org/10.17181/CERN.4DGI.MZN4

[10] G. Tuci and G. Punzi, Reconstruction of track candidates at the LHC crossing rate using FPGAs. EPJ Web Conferences **245**, 10001 (2020). https://doi.org/10.1051/epjconf/202024510001

[11] Bittware, 520N product page (2023). https://web.archive.org/web/20230603104042/https://www.bittware.com/products/520n/

[12] G. Corti et al., Software for the LHCb experiment. IEEE Transactions on Nuclear Science **53, 3**, 1323–1328 (2006). https://doi.org/10.1109/TNS.2006.872627

[13] LHCb Collaboration, *LHCb Data Acquisition Enhancement TDR* (CERN, Geneva, 2024). http://dx.doi.org/10.17181/CERN.L9E7.N06X