Proceedings of CTD 2020
PROC-CTD2020-23
September 28, 2020

An optical network for accelerating real-time tracking with FPGAs

Andrea Contu[1], Federico Lazzari[1,2], Giovanni Bassi[1,3], Giovanni Punzi[1,4], Giulia Tuci[1,4], Michael J. Morello[1,3], Mirco Dorigo[1], Riccardo Fantechi[1], Simone Stracka[1,4], Wander Baldini[1,5]

On behalf of the LHCb-RTA project,
[1] *INFN, Italy*
[2] *Università degli Studi di Siena, Italy*
[3] *Scuola Normale Superiore, Italy*
[4] *Università di Pisa, Italy*
[5] *Università degli Studi di Ferrara, Italy*

ABSTRACT

The "Artificial Retina" is a highly-parallelized tracking architecture that promises high-throughput, low-latency, and low-power consumption when implemented in state-of-art FPGA devices.
Working on not-builded events, the "Artificial Retina" needs a dedicated distribution network of large bandwidth and low latency, delivering to every FPGA the hits required to perform track reconstruction. This is a technologically challenging aspect of the system which has not yet been tested in a life-size application.
The upgraded LHCb DAQ is an ideal environment for a preliminary test of this methodology. Since the upcoming LHC Run-3, the LHCb Level-0 hardware trigger will be removed. The experiment will readout, build and deliver events to the High Level Trigger system at the full LHC collision rate of 30 MHz. The present work is part of a wider Real Time Analysis project, which has been formed with the purpose of organising data processing within this novel environment. The "Artificial Retina" represents an R&D project towards future fast technologies for real-time tracking.
We used as benchmark the reconstruction of tracks within the new VELO-pixel detector, that is composed of a limited number of readout units. We introduce the design of the fast optical network, carrying a total bandwidth of ∼15 Tb/s, studied for performing VELO tracking in real time. and the results of our tests of an actual prototype assembled and integrated in a vertical slice of the upgraded LHCb DAQ.

PRESENTED AT

Connecting the Dots Workshop (CTD 2020)
April 20-30, 2020

1 Introduction

Computational needs required by high energy physics (HEP) experiments keep increasing, not just for event reconstruction but also for data distribution to the server farms. As an example, during Run-3 the LHCb Event Builder will have to handle the sizeable bandwidth of 40 Tbps. If the current approach is adopted also in Run-5, the required bandwidth will increase by an additional factor of 10, at least. In this view, it is valuable to explore new solutions based on heterogeneous computing.

Modern FPGAs have the capability to perform highly-parallel processing, with high throughput and low latency. This allows us to develop a tracking system that can operate at the very fist processing level in a high-rate environment like the LHC [1]. With such a system we can instrument just the desired sub-detectors and replace hit data with tracks in real time. Thus, the data volume is reduced by providing trigger primitives, even before performing Event Building.

As shown in Figure 1(a), data from different layers are typically read out separately by the DAQ. Following our approach, data could be forwarded to several Tracking Boards. Track reconstruction requires to combine data from several different layers, hence Tracking Boards need to exchange information through a Patch Panel. If the entire process is completed within few μs, tracks can be added to the event record before data are forwarded to the Event Builder.

To answer these needs, we have developed the "Artificial Retina" tracking architecture which grants highly-parallel tracking to be implemented at the very first processing level of a HEP experiment.

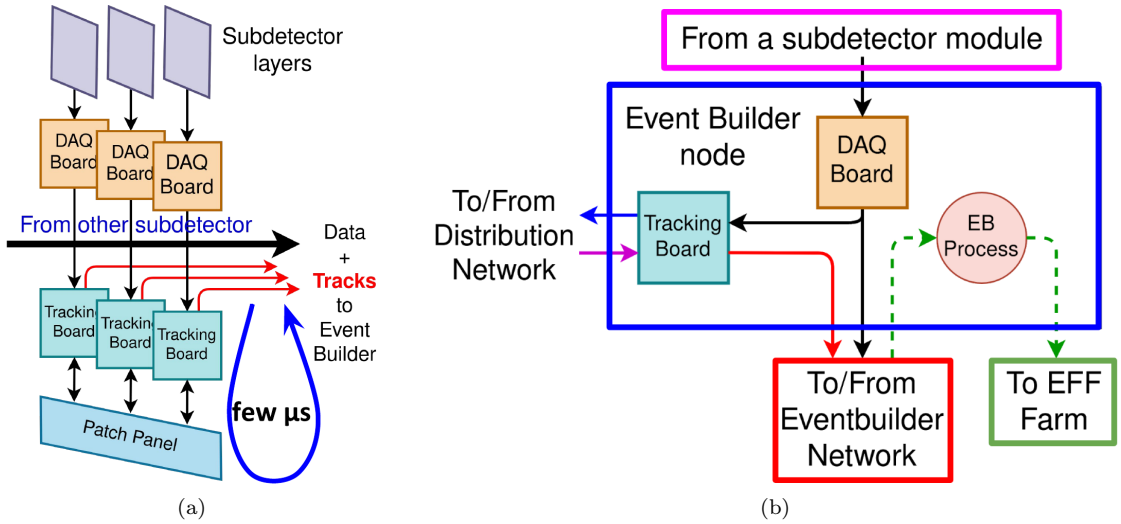


Figure 1: Schematics of the data flow in a tracking system that can operate at the very first level of processing (a), and its integration in the LHCb Event Builder (b).

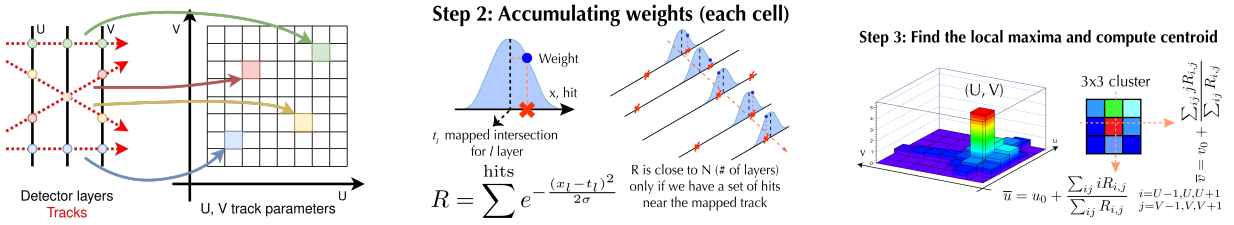
2 The "Artificial Retina" architecture

The "Artificial Retina" is a highly-parallel tracking architecture formally similar to the "Hough transform", a method already applied to find lines in image processing.

The track parameters space is represented by a matrix of cells (Figure 2a). The center of each cell corresponds to a reference track in the detector that intersects the layers in specific spatial points called receptors.

Each cell computes a weighted sum of hits nearby the reference track. The weights are proportional to the distance between the hits and the receptor. The resulting sum approaches the number of subdetector layers as the set of hits gets closer to the reference track (Figure 2b).

Subsequently, the cells search local maxima in the matrix of weighted sums. Track parameters are reconstructed by interpolating the responses of cells nearby the local maxima (Figure 2c).



(a) (b) (c)

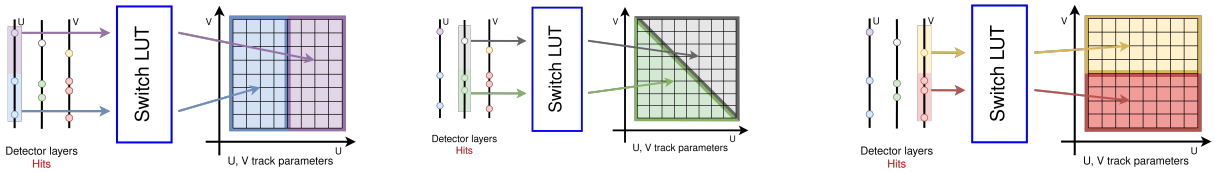
Figure 2: Track reconstruction steps with the "Artificial Retina" architecture.

To reach high-throughput, cells work in a fully-parallel way, being spread over several chips to overcome FPGA size limitations without increasing latency. Each FPGA processes a portion of every event, differently to the usual HEP approach in which each CPU inside HPC farms reconstructs a set of events.

The crucial backbone of the "Artificial Retina" is a Distribution Network capable of exchanging data quickly and effectively between FPGA boards.

3 A dedicated distribution network

Each Tracking Board is paired with a DAQ board, so that it only gets data from a portion of the sub-detector. A custom network redistributes the hits by track parameter coordinates, performing something similar to a "change of reference system". Furthermore, by using Lookup Tables (LUTs) the Distribution Network delivers to each cell only the hits close to its receptors (Figure 3), resulting in a smaller number of hits to be processed by the cell. This allows the system to reach higher throughput.



(a) (b) (c)

Figure 3: Hits routing from the detector coordinate space to the track parameters space. Hits coming from a specific region are delivered only to the cells in the corresponding region.

We designed a modular Distribution Network, with the dispatcher as the basic block. It has two inputs and two outputs, being able to send any input to any output (even both) according to the LUT routing scheme. Combining a sufficient number of dispatchers it is possible to build a Distribution Network with the desired number of inputs and outputs. The network is implementable within the same array of FPGAs performing the tracking, in separate and independent locations of the chip.

Modern FPGAs have numerous high-bandwidth transceivers (XCVRs), that can be used to implement optical serial links between the boards. Optical serial links allow to exchange hits between several boards with great flexibility, connecting distant boards, increasing the number of links on a busy path, or implementing only useful paths.

4 Prototype throughput and latency performance

We built a prototype of the "Artificial Retina" system based on Intel Stratix-V FPGAs and tested its track reconstruction performances with simulated events. The simulated detector is composed by a generic 6 axial-layers layout. Input data are stored in circular buffers injecting 'realistic' hit data at 30 MHz, as in LHC running. Since data come from different readout units, it is crucial to simulate realistic time skews between several inputs. At Run-3 running conditions the maximum expected delay is $\sim$10 μs. We delayed one input by this value to stress the internal buffers and ensure the tolerance to input latencies.

While testing the buffer size remained under control, and the output rate matched the input one. We verified that every track has been correctly reconstructed comparing the actual output with the tracks reconstructed by a C++ simulation of the system.

We did split the system between two FPGAs: one with the Distribution Network and one containing the Cells. The boards were interconnected by 12 optical fibers. In this configuration we achieved the same event rate as when the system is implemented in a single FPGA, demonstrating that the "Artificial Retina" can be distributed over more FPGAs without decreasing the throughput [2]. In this configuration we measured a latency of $\sim$0.4 μs.

5 Application to LHCb

Our system will be part of a testbed that the LHCb experiment is deploying for the specific purpose of testing new computing accelerators during data taking in the upcoming LHC Run.

LHCb is a forward spectrometer designed to study b and c physics. It is being upgraded and in 2021 will start to acquire data with an almost completely new detector and trigger system [3]. Instantaneous luminosity will increase by a factor of 5, and to take full advantage of the increased statistics, the upgraded LHCb will have a trigger-less readout system. The full inelastic collision rate of 30 MHz will be processed by the fully-software trigger (HLT). The Event Builder will handle the sizable bandwidth of 40 Tbps.

Figure 1(b) shows how the "Artificial Retina" can be integrated in the LHCb DAQ system. In LHCb data flow from the detector to the Event Filter Farm (EFF) through the Event Builder nodes and its network. The Tracking Boards could be installed in the Event Builder nodes, reading hits from the DAQ boards and providing tracks. The Event Builder performs the building, seeing the "Artificial Retina" like a virtual sub-detector.

Reconstructing tracks in the LHCb Vertex Locator (VELO) is the most time consuming task in the first stage of the HLT reconstruction, requiring the 48.3% of the overall processing time. Hence, this sub-detector represents an interesting use case for the implementation of the "Artificial Retina". We already performed studies concerning the physic performance of an "Artificial Retina"-based VELO tracker [4]. However building a large network among the Tracking Boards is technically challenging, and demonstrating the feasibility of a life-size network is an important step for the realisation of the system.

6 Life-size network prototype

The VELO detector is composed of 52 silicon pixel modules, of which 38 are placed in the forward region. The remaining layers are mainly used to optimise the primary vertex precision. A single DAQ board acquires data from a VELO module, therefore we need at least 38 Tracking Boards to read data from the forward region. To ensure an appropriate bandwidth between the boards, we split the VELO track parameters space in 4 quadrants. Cells of the same quadrant share a major numbers of hits, requiring a highly interconnected network between the corresponding FPGAs. Cells of different quadrants can also share hits, but needing less bandwidth, the network implemented is simpler.

We can implement 4 full-mesh networks, one for quadrant, 10 FPGAs each. Then the i-th FPGA of a network is connected to the i-th FPGA of the other three networks creating a full-mesh of full-meshes (Figure 4). Hit exchange between not directly connected FPGAs require a hop, increasing the communication latency. However the total latency remains below the μs limit.

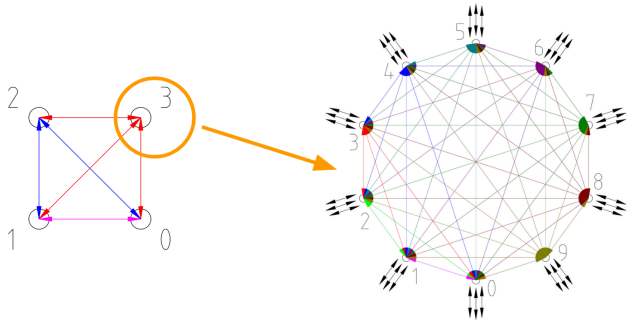


Figure 4: Network topology of a full-mesh network of 4 full-mesh sub-networks based on 10 FPGAs each.

This configuration requires 12 links for each FPGA. Several off-the-shelf FPGA boards are equipped with 16 XCVRs, so this topology is implementable without the need for custom hardware.

We are building a prototype of a 10-node full-mesh network in the LHCb Data Center using half-sized FPGA cards. The first operational plan is to process simulated data, gradually moving to parasitic operation on real VELO data during Run-3 data taking.

We performed a preliminary test with 5 cards in a full-mesh configuration (Figure 5). Every board sends pseudo-random sequences (PRBS31) to the other 4 FPGAs through 10 Gbps optical links. We tested the system for 23 consecutive days at maximum speed, without detecting transmission errors on all but one link. The Bit Error Rate (BER) of the links is lower than $2 \cdot 10^{-16}$ (CL = 95%). The BER of the fault link is $\sim 10^{-13}$. By reducing the transmission speed, the errors disappeared.

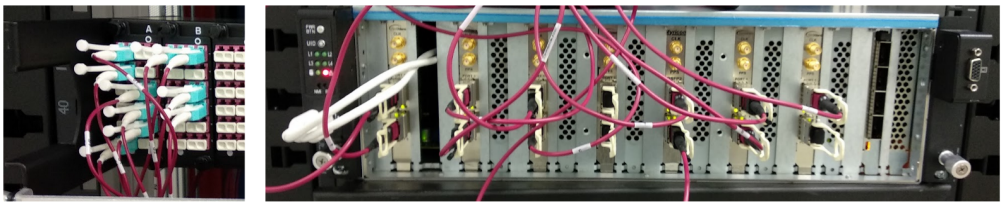


Figure 5: The hardware used for the test. The cables go to a patch panel (left) that allows to interconnect the FPGAs (right) with the desired network topology.

Afterwards we implemented a realistic Distribution Network using simulated hits, written into the FPGAs memories, from different events, instead of pseudo-random sequences. Special words, called EndEvent, separate hits of different events, while carrying the event identification number. Data reading on different FPGAs is not synchronised, for a better correspondence to the actual environment in which the system will be places. In each FPGA a network of Dispatchers routes hits to 8 different lines according to the FPGA of destination. We based the data transmission on the Intel SerialLite II, which provides flow control, preventing transmission when the receiving buffer is full. Having implemented 8 lines using only 5 FPGAs, some XCVRs are closed in serial loopback. The Cells require that the events coming from the XCVRs are aligned, at their receiving side. If an EndEvent is missing or a transmission error modifies the event ID, the alignment can not be ensured and an error flag is raised. This is useful to monitor the correct behaviour of the Distribution Network and of the XCVRs. As a debugging step, we can also read the output data from the host computer, and compare it with the simulation. During a 5 weeks test we did not detect any error.

7 Conclusions

In future HEP experiments event building and tracking will become more challenging. In this view, it is important to explore new solutions based on heterogeneous computing. The “Artificial Retina” is a highly-

parallel tracking system that can provide efficient real-time processing of data already at pre-build level, takings full advantage of modern FPGAs capability.

A fast dedicated Distribution Network is a crucial element of the system, which allows to collect data from several DAQ nodes, overcoming FPGA size limits, while reaching high-throughput and low-latencies. We designed a network suitable for a real application in LHCb, and we now have a life-size prototype in advanced state of realisation, processing simulated data. Our system will be part of a testbed that LHCb is deploying for the specific purpose of testing new computing accelerators. The plan is to perform parasitic operation on real VELO data during Run-3 data taking.

ACKNOWLEDGEMENTS

We are grateful for the funding granted by INFN under CSN1 and CSN5 project RETINA.

References

[1] G. Punzi Et al., "Real-time reconstruction of pixel vertex detectors with FPGAs," The 28th International Workshop on Vertex Detectors **Volume 373**, Pages 047 (2019) [https://pos.sissa.it/373/047/].

[2] F. Lazzari Et al., "Performance of a high-throughput tracking processor implemented on Stratix-V FPGA," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **Volume 936**, Pages 344 - 345 (2019) [https://doi.org/10.1016/j.nima.2018.08.025].

[3] LHCb Collaboration, "LHCb Trigger and Online Upgrade Technical Design Report," CERN-LHCC-2014-016 (2014) [https://cds.cern.ch/record/1701361].

[4] G. Tuci Et al., "Reconstruction of track candidates at the LHC crossing rate using FPGAs," 24th International Conference on Computing in High-Energy and Nuclear Physics (2019) [https://indico.cern.ch/event/773049/contributions/3474316/].