FPGA-based real-time cluster finding for the LHCb silicon pixel detector

Giovanni Bassi on behalf of the LHCb RTA project
giovanni.bassi@cern.ch

26th International Conference on Computing in High-Energy and Nuclear Physics (CHEP2023), 8th - 12th May 2023
Background

- LHCb is at the leading edge of high-precision heavy-flavor measurements
- Uncertainties on many key observables are still dominated by statistics ⇒ **bigger data samples, increasing luminosity**
- Need to **process data at low data acquisition levels** to limit data flow, while reducing computing resources needs and power consumption
- Heterogeneous computing is one of the most promising solutions and **FPGA accelerators** are well suited to address, in a high parallel way, heavy repetitive tasks
- Grouping contiguous pixels in single hits (**clustering**) is both a time demanding (2D pixel geometry) and a repetitive task
- We developed a FPGA-friendly clustering algorithm, based on the Retina project [CHEP2023, May 9, 14:15, track 2], to tackle 2D clustering during early DAQ stages, while keeping the **same tracking performance** wrt CPU clustering

Today

Giovanni Bassi

FPGA-based real-time cluster finding
LHCb Upgrade

- LHCb is a single arm spectrometer, designed for precision studies of b- and c-hadrons
- LHCb has just completed a major upgrade
- Constraints:
  - $L = 2 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$ (x5 wrt Run 2)
  - 7.6 vertices (1.1 in Run 2)
- Design choices:
  - upgrade of the majority of sub-detectors
  - readout system dealing with a 30 MHz data processing rate (average LHC bunch crossing rate)
  - 32 Tb/s data flow from the detector to the High Level Trigger farm (HLT1-GPU + HLT2-CPU)

Giovanni Bassi

FPGA-based real-time cluster finding
The VELO detector of LHCb

- The clustering algorithm has been tailored for the LHCb Vertex Locator (VELO):
  - 26 layers each made of 2 modules
  - Each module consists of 4 sensors
  - 1 DAQ card per module
  - 41 M pixels in total

- Pixels are read in groups of 2x4 pixels (SuperPixels)
- VELO clusters are typically made of few pixels (1-4)
- The first step of the cluster reconstruction is to flag isolated SuperPixels (isolated = none of the 8 neighbors SPs have any active pixel)
Isolated SPs are resolved with a Look Up Table (LUT) allowing for an extremely fast processing of isolated SPs, with a very limited amount of logic resources within the FPGA.

LUT connects each of the 256 ($2^8$) possible pixel configurations inside a SP to the center of mass of the cluster/s (if two clusters are generated).
Algorithm overview

- SPs with neighbors fill a set of matrices, 3x3 SPs each (6x12 pixels)
- First SP filling a matrix determines position of the matrix in the detector set of coordinates of SPs that can fill the matrix
- If a SP belongs to a matrix it fills it, otherwise it moves forward, checking the next matrix or filling a blank one in the center
Algorithm overview

- At the end of each event, in a fully parallel way, each pixel checks if it belongs to one of the following patterns, if so a cluster candidate is identified.
- Each cluster candidate is resolved using a LUT.

Algorithm parameters:
- Matrix shape and size $\Leftarrow$ average number of neighbor SPs and their arrangement.
- Number of matrices $\Leftarrow$ distribution of total number of not isolated SPs per event.
- Cluster maximum dimension (3x3 pixels) $\Leftarrow$ distribution of cluster sizes.

Giovanni Bassi

FPGA-based real-time cluster finding
Having defined the algorithm behavior, the corresponding firmware has been developed in VHDL.
The first functioning prototype of the clustering firmware was developed and tested within the LHCb-Pisa INFN laboratory. To run as a real-time algorithm, the firmware has to sustain the LHC average bunch crossing rate of 30 MHz. Firmware tested on a prototyping board to measure:

- Amount of logic required on the chip
- Average frequency of events clusterized (throughput)

We measured:

- 26% logic occupancy of the chip
- Average throughput of 38.9 MHz

The low amount of logic required and the achieved throughput (>30 MHz) ease the full integration of the clustering firmware inside existing DAQ boards, without extra costs.
Physics performance

- Moving from a full-fledged software implementation of the VELO clustering to a FPGA-based one required a careful evaluation of possible impacts on physics performances in terms of:
  - Cluster efficiency $\rightarrow$ find hit on detector
  - Cluster residual $\rightarrow$ match hit position
  - Track efficiency $\rightarrow$ find track
  - Track resolution $\rightarrow$ match track parameters

<table>
<thead>
<tr>
<th>Track type</th>
<th>Quantity</th>
<th>CPU clusters [%]</th>
<th>FPGA clusters [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>All VELO tracks</td>
<td>efficiency clone</td>
<td>98.254 ± 0.007</td>
<td>98.254 ± 0.007</td>
</tr>
<tr>
<td></td>
<td>clone</td>
<td>1.231 ± 0.006</td>
<td>1.234 ± 0.006</td>
</tr>
<tr>
<td>Long tracks</td>
<td>efficiency clone</td>
<td>99.252 ± 0.006</td>
<td>99.252 ± 0.006</td>
</tr>
<tr>
<td></td>
<td>clone</td>
<td>0.806 ± 0.006</td>
<td>0.806 ± 0.006</td>
</tr>
<tr>
<td>Ghost</td>
<td></td>
<td>0.848 ± 0.003</td>
<td>0.928 ± 0.003</td>
</tr>
</tbody>
</table>

- FPGA algorithm tracking performance is nearly indistinguishable from CPU/GPU clustering.
Integration

- Having verified that the clustering has a high enough throughput, while satisfying the very demanding LHCb physics requirements, the clustering firmware has been fully integrated within the VELO DAQ firmware.

- LHCb DAQ cards are equipped with an Intel Arria 10 FPGA:
  - Detector data received on optical links
  - SP data are decoded, time aligned and sent to the clustering block
  - Clusters are sent to the host server via PCIe

- Clustering requires:
  - 31% of the logic elements
  - 11% of the M20K memory blocks
  - 350 MHz clock

- The entire VELO firmware, including clustering, requires 73% of logic elements and 71% of M20K memory blocks.
Throughput, bandwidth & power consumption

- Moving VELO clustering from the HLT1 sequence (GPUs) to early data processing (FPGAs) allows HLT1 to accept a **11% higher rate of events**, since clusters are already available.
- Moreover, outputting clusters instead of SPs leads to an additional benefit of a **14% bandwidth reduction**, as reconstructed clusters are less than input SPs.
- With the addition of cluster reconstruction, VELO readout cards need to perform more operations. We measured the additional FPGA power consumption to be 2.5W, comparing SP and cluster firmwares (+130W on all 52 VELO DAQ cards).
- As a comparison, performing clustering on GPUs would have required roughly 6kW ⇒ **clustering on FPGAs requires O(50x) less electrical power wrt the GPU implementation**.
New opportunities

- With cluster reconstruction occurring inside VELO DAQ cards, it becomes possible to perform beam parameter measurements, such as:
  - Luminosity
  - Spillover
  - Beam position

- These parameters can be measured at a very high rate in the firmware, and accessed via slow control, even when HLT1 is not running

- Luminosity counters have been implemented, calibrated and tested during luminosity and Van der Meer scans

Giovanni Bassi
FPGA-based real-time cluster finding
Summary

- Despite being a conceptually simple task, clustering requires a non-negligible amount of the time needed for the entire HLT1 sequence and consumes a non-negligible amount of power.
- Being simple and highly parallelizable makes clustering an ideal candidate to be moved from HLT1 to a preprocessing stage, with benefits for the entire data acquisition chain.
- We have developed, implemented, and commissioned, for the first time, a 2D FPGA-based clustering algorithm that processes events from VELO-pixel sensors in real time, at the unprecedented speed of 30 MHz.
- Given the limited amount of logic (31%) and memory (11%) required, the firmware has been fully integrated in the existing DAQ readout cards, without extra costs, leading to:
  - 11% increase in HLT1 event rate
  - 14% reduction of the required bandwidth
  - O(50x) reduction in power consumption
- The FPGA-based clustering algorithm has been fully commissioned at LHCb in 2022, and is now the adopted solution for physics data taking for the Run 3.
A FPGA-based architecture for real-time cluster finding in the LHCb silicon pixel detector


Abstract—This article describes a custom VHDL firmware implementation of a two-dimensional cluster-finder architecture for real-time cluster finding in the LHCb silicon pixel detector. The architecture was embedded in the Xilinx Zynq system on chip, has been deployed to the existing FPGA stage that performs the rejection of the VETOs, as a further enhancement of the VETO system, and may run in real time during physics data taking, replacing the current VETO firmware implemented in the FPGA-based VETO. The VETO processing allows the first level of the software trigger to accept an 1% higher rate of events, in the presence of a full luminosity, which represents a significant increase in efficiency. The firmware has been tested in real conditions and consumes significantly less electrical power. It additionally allows for the real data to be flushed down to the event level, thus saving approximately 14% of the FPGA bandwidth. Detailed simulations studies have shown that the use of this real-time cluster finding does not introduce any appreciable degradation in the tracking performance in comparison to a full-fledged software implementation. This work is part of a continuous effort aimed at enhancing the running processing capability of SHP experiments, and it is expected that the firmware will be deployed in the next versions of the data acquisition chain.

Artivo: Time-Clustering, Connected Component Labeling, FPGA, LHCb, VETO

1. INTRODUCTION

The LHCb experiment has collected data over the past decade, during the Run 1 and Run 2 of the LHC, and currently undergoes a major upgrade for the current Run 3. In addition to replacing most of the subsystems, the front-end electronics and data-acquisition systems were completely reviewed [1]. To lead out and perform the complete information of the detector at the full LHC beam-stripping rate of 80 MHz (50 MHz averaged over the LHC ring) this change is motivated by the needs of the LHCb physics programme, which requires the collection of low-transverse-momentum events that need high-level processing to be distinguished from background events [2]. This evolution puts a large computing

Additional contributors:


IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 71, NO. 7, SEPTEMBER 2014

1

Giovanni Bassi

FPGA-based real-time cluster finding

IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 71, NO. 7, SEPTEMBER 2014

15

REFERENCE

The paper has been accepted for publication by IEEE Transactions on Nuclear Science and it is available as a preprint on 10.1109/TNS.2023.3273600

● More details about the clustering algorithm and the related firmware can be found in the paper

● The paper has been accepted for publication by IEEE Transactions on Nuclear Science and it is available as a preprint on 10.1109/TNS.2023.3273600
BACKUP
Field Programmable Gate Arrays

- FPGAs are integrated circuits that can be configured by the user.
- FPGAs contain an array of programmable logic blocks, that can be programmed to perform different logic functions.
- I/O ports, PLLs, memory blocks and clock distribution lines are integrated within the FPGA.
- Configuration is done using a hardware description language (HDL).

FPGA-based real-time cluster finding

Giovanni Bassi
The track parameters space is represented by a matrix of cells. 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.

Cells search local maxima in the matrix of weighted sums. Track parameters are reconstructed by interpolating the responses of cells nearby the local maxima.

Giovanni Bassi

FPGA-based real-time cluster finding
LHCb trigger

- Run 2 trigger
  - hardware Level-0 stage: 40 MHz → 1 MHz
  - HLT1 fast tracking: 1 MHz → 100 kHz
  - HLT2 full event reconstruction: 100 kHz → 12.5 kHz
- Moving from Run 2 to Run 3 we need to categorise different “signals” → access as much of the event as possible, as early as possible
- Run 3 trigger:
  - Full 30 MHz and x5 pileup
  - HLT1: 30 MHz → 1 MHz
  - HLT2: 10GB/s to permanent storage

Giovanni Bassi  
FPGA-based real-time cluster finding
Event building (EB)

- Each sub detector sends raw data, asynchronously, to a unique EB node, that receives data through DAQ cards.
- EB nodes exchange detector data using an EB network.
- All detector data for an event are sent to a specific EB node that builds the event.
- GPU cards run HLT1 within the EB servers, reducing the data rate at the output of the EB by a factor of 30-60.
- An array of disk servers buffers the HLT1 output data.
- A separate server farm runs HLT2.

FPGA-based real-time cluster finding
Firmware overview - detailed view

Giovanni Bassi

FPGA-based real-time cluster finding
Algorithm peculiar behaviors

- Algorithm parameters:
  - Matrix shape and size $\cong$ average number of neighbor SPs and their arrangement
  - Number of matrices $\cong$ distribution of total number of not isolated SP per event
  - Cluster maximum dimension (3x3 pixels) $\cong$ distribution of cluster sizes

- Examples of peculiar behaviors of the algorithm:
LHCb track classification

Tracking performance is evaluated using LHCb simulated events, comparing the output of the track reconstruction using FPGA and CPU clustering algorithms.

LHCb track classification

- **Velo**: track with hits on the VELO
- **Long**: track with hits on the three LHCb trackers (VELO - UT - Scifi)
- **Ghost**: a reconstructed track not associated to a simulated track
- **Clone**: a second reconstructed track associated to the same simulated track