FPGA based data processing in the ALICE High Level Trigger in LHC Run 2

Heiko Engel$^1$, Torsten Alt$^2$ and Udo Kebschull$^3$ for the ALICE Collaboration

$^1$Institut für Informatik, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
$^2$Frankfurt Institute for Advances Studies, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
E-mail: hengel@cern.ch

Abstract. The ALICE High Level Trigger (HLT) is a computing cluster dedicated to the online compression, reconstruction and calibration of experimental data. The HLT receives detector data via serial optical links into FPGA based readout boards that process the data on a per-link level already inside the FPGA and provide it to the host machines connected with a data transport framework. FPGA based data pre-processing is enabled for the biggest detector of ALICE, the Time Projection Chamber (TPC), with a hardware cluster finding algorithm. This algorithm was ported to the Common Read-Out Receiver Card (C-RORC) as used in the HLT for Run 2. It was improved to handle double the input bandwidth and adjusted to the upgraded TPC Readout Control Unit (RCU2). A flexible firmware implementation in the HLT handles both the old and the new TPC data format and link rates transparently. Extended protocol and data error detection, error handling and the enhanced RCU2 data ordering scheme provide an improved physics performance of the cluster finder. The performance of the cluster finder was verified against large sets of reference data both in terms of throughput and algorithmic correctness. Comparisons with a software reference implementation confirm significant savings on CPU processing power using the hardware implementation. The C-RORC hardware with the cluster finder for RCU1 data is in use in the HLT since the start of Run 2. The extended hardware cluster finder implementation for the RCU2 with doubled throughput is active since the upgrade of the TPC readout electronics in early 2016.

1. Introduction
ALICE [1] is one of the four major experiments at the Large Hadron Collider at CERN and is dedicated to the study of the physics of strongly interacting matter in central heavy-ion collisions. ALICE consists of 19 sub-detectors that are read out with a custom optical link protocol called Detector Data Link (DDL). The DDL serves as interconnect between the front-end electronics of the detectors and the computing clusters of Data Acquisition (DAQ) [2] and High Level Trigger (HLT) [3] for detector configuration and data readout. The DDL is available at different link rates between 2 and 6 Gbps [4].

The Data Acquisition system and the High Level Trigger use custom FPGA based Read-Out Receiver Cards (RORCs) as interface to the DDL. The Common Read-Out Receiver Card (C-RORC) [5] was developed as a joint effort between the Data Acquisition and the High Level Trigger groups for Run 2 to enable the readout of detectors at higher link rates, to extend the hardware based online processing of the data and to provide state-of-the art interfaces with a
common hardware platform. The advances in technology additionally provide an increased link density per board and therefore reduce the overall number of required boards and host machines. The ATLAS experiment joined the project to use the same hardware for the upgrade of their readout system. The C-RORC hardware is installed and successfully used in the production systems of ALICE Data Acquisition, ALICE High Level Trigger, ATLAS TDAQ Readout System and ATLAS Region-of-Interest Builder since the start of LHC Run 2 [6]. Additionally, it serves as a development platform for the ALICE Run 3 readout system [7]. The C-RORC hardware comes with a XILINX Virtex-6 FPGA and an eight lane PCI Express Generation 2 interface to the host machine. It provides up to 12 serial optical links with up to 6.6 Gbps each. The board hosts two DDR3 SO-DIMM sockets that provide large on-board memory to the FPGA if needed. A configuration controller supports multiple on-board FPGA configuration images and mechanisms to safely configure and reconfigure the FPGA even if the PCI Express link is down. An overview of the hardware is shown in figure 1.

Figure 1. C-RORC hardware overview.

In the ALICE online architecture experimental data from the detectors is sent to the C-RORCs of the DAQ system for readout into the data acquisition environment. Additionally, a copy of the raw data is sent to the High Level Trigger via the DDL already inside the DAQ C-RORC. On HLT side the data is again received via C-RORCs as data entry point into the computing cluster. The HLT processed data is sent back to the DAQ system using C-RORCs and DDL. The data from HLT allows the DAQ system to discard the original raw data from detectors in favor of the compressed data from HLT. This mechanism significantly reduces the data volume towards permanent storage and enables the experiment to cope with Run 2 data rates in the first place. An overview of the ALICE online architecture with focus on the C-RORC installation is shown in figure 2.

Figure 2. ALICE online architecture in RUN 2. Experimental data from the detectors is received in the DAQ system. A copy of the data is sent to HLT, processed and provided back to the DAQ system. Both systems use C-RORCs as interface to the DDL.
2. The Common Read-Out Receiver Card in the ALICE High Level Trigger

The High Level Trigger is a computing cluster for online reconstruction, compression and calibration of ALICE experimental data. It consists of 180 identical server nodes each with a Graphics Processing Unit (GPU) and interconnected with a 56 Gbps Infiniband FDR network [8]. 74 of these nodes are additionally equipped with C-RORCs as the main data input and output interface of the HLT from and to the DAQ system. A custom data transport framework [9] based on the publisher-subscriber model handles the gathering of associated data fragments from different sources, the distribution of processing tasks across nodes as well as the processing tasks themselves. A framework interface to AliRoot [10], the ALICE Offline framework for simulation, reconstruction and analysis, instantiates AliRoot tasks as the actual processing entities. This enables the same code to be used online in the HLT and in the Offline environment.

The HLT uses hardware accelerators for the online processing of detector data. Raw data from the Time Projection Chamber (TPC) [11], the detector with the highest data volume in ALICE, is already processed inside the C-RORC FPGAs with a hardware cluster finding algorithm. This processing step is described in more detail in section 3. The GPU in each node is used to accelerate the reconstruction of particle trajectories [12].

The C-RORC in the HLT is used in two different data flow directions. As HLT input interface for experimental data from the DAQ it receives the raw data from the DDL and transfers it into the host RAM via Direct Memory Access (DMA). If the data source is a DDL from TPC, the data is additionally processed with the cluster finding algorithm. The C-RORC as HLT output interface for the processed data towards the DAQ fetches data from the host RAM via DMA and pushes it onto the DDL. An overview of the firmware variants is shown in figure 3.

![Figure 3. 74 HLT nodes are equipped with C-RORCs as either input or output interface to the HLT with data flow from DDL to host or from host to DDL.](image)

The DDL link rate depends on the connected detector and a total of four different link rates are in use throughout the whole C-RORC installation. In order to keep the number of different firmware images at a minimum and to support any future link rate changes, a common input firmware was developed that supports changing the DDL link rate at run time by software. For the internal clocking architecture of the TPC readout with the cluster finder it turned out to be more effective to maintain a separate firmware image for the TPC readout. This variant uses discrete link rates for the TPC DDLs. A list of active firmware images in the HLT production system is shown in table 1. In order to distribute the input and output bandwidth effectively across the cluster and to optimize the performance of the data processing framework, not all C-RORC use all 12 available links.

| Firmware variant | Data flow   | DDLs | Link speed [Gbps] | Data processing |
|------------------|-------------|------|-------------------|-----------------|
| HLT-IN           | DDL to host | 12   | up to 5.3125      | -               |
| HLT-IN-FCF       | DDL to host | 6    | 2.125 or 3.125   | TPC cluster finder |
| HLT-OUT          | host to DDL | 4    | up to 5.3125      | -               |

Table 1. List of C-RORC FPGA firmware images deployed in the HLT cluster
The DDR3 memory on the C-RORCs is used for data replay purposes in the HLT. Previously recorded, simulated or generated data can be loaded into the on-board RAM and replayed as if it were coming from the DDL. The HLT output firmware can be configured to discard data right before it would be pushed into the DDL towards the DAQ. This method can be used for all C-RORCs in the cluster simultaneously and provides a setup to test the full HLT cluster in a controlled environment. Data replay tests are independent of any DAQ, detector or LHC state which allows us to conduct them at any time. Configurable event replay rates enable to test the limits of the full system. Replaying invalid or erroneous data can be used to verify a wide range of error handling and error mitigation techniques applied in the HLT processing chain.

3. TPC Hardware Cluster Finding
The Time Projection Chamber is the main device for tracking and identification of charged particles in ALICE. It consists of a cylinder filled with gas in a magnetic field with a strong electric field from a central electrode towards the end plates. Charged particles ionize the gas while traversing the detector. The electric field makes the ionization electrons drift towards the end plates where their arrival point and time is precisely measured with arrays of readout pads arranged in rows and columns on trapezoidal planes. The analog pad signals are pre-amplified and shaped before analog-to-digital conversion. Baseline correction and zero suppression are then applied in the ALICE TPC ReadOut (ALTRO) chip on Front-End Cards (FECs). The Readout Control Unit (RCU) interfaces up to 25 FECs per board to the Data Acquisition system using the DDL. The TPC consists of 18 trapezoidal sectors per end plate, each sector consisting of six partitions. Each DDL from one RCU delivers data from one partition, thus covering a trapezoidal region of rows and pads.

The first generation of RCU boards as used during Run 1 and up to the end of 2015 in Run 2 provides data with a bandwidth of up to 160 MB/s but has limited buffering capabilities. For that reason the readout was done in two interleaved branches per row: Each row of pads is divided approximately in half. With 68 pads in the innermost row of partition 0, pads 0 to 33 belong to branch A while pads 34 to 67 belong to branch B. Interleaved readout means that samples first come from pad 0, then pad 34, followed by pad 1, then pad 35 and so on. The second generation of RCU boards installed early 2016 provide sufficient buffering capabilities to provide the data from all pads in numerical order and with a bandwidth of up to around 310 MB/s [13].

The HLT performs cluster finding already in the C-RORC FPGA with the FastClusterFinder core. The preprocessing core was developed for and used during Run 1 with the first generation of RORCs [14]. The pad and time response functions of the TPC provide characteristic signals of charge distributions within the TPC volume in pad and time direction. The cluster finder searches for these signatures in pad and time direction in the raw data and calculates their properties. The clusters found during the cluster finding process typically span across a few pads and a few time stamps, which allows us to calculate the position and width of these distributions at a precision below the intrinsic resolution of pad size and time stamp granularity. The shape of the clusters can be described with sufficient accuracy by Gaussian distributions in both pad and time direction which can be implemented efficiently in hardware. The underlying math consists of weighted sums of the charge samples with pad number or time stamp as weighting factors.

![Figure 4. Fast-ClusterFinder firmware components](image-url)
The implementation of the FastClusterFinder is sketched in figure 4. An input FIFO decouples the core from the DDL so the cluster finder can be operated at a clock frequency higher than the DDL clock. The RCU Decoder decodes the incoming stream of 32 bit data words from the RCU into a serial sequence of charge samples. With up to three samples per DDL word the RCU Decoder has to run at least with three times the clock frequency of the DDL in order to handle the data rate. Headers in the data stream are used to look up row and pad numbers and a per-channel gain correction factor from a Block-RAM. This makes it possible to compensate differences in the charge sensitivity of the individual channels or mask-out channels.

The way the data is provided from the RCU and the fact that the weighted sums are additive across contributions from different pads enables to process the data in two steps. In the Channel Processor the stream of samples is only analyzed in time direction on a per pad basis to find peaks and pre-calculate properties in time direction. In a second step the candidates found in time direction are merged with nearby candidates from the neighboring pad in the Channel Merger. Both entities are able to separate signals from overlapping clusters. The Channel Merger is implemented as a state machine that compares the current input candidate to the output of a local FIFO queue. This is the only element in the chain with data dependent processing capabilities. The interleaved readout of data from two branches described above requires to handle the cluster candidates from the different branches independently. For that reason two FIFOs are used inside the Channel Merger to correlate data from neighboring pads, one for each branch. Up to this point all calculations are done as fixed point arithmetic and do not cut any precision. For the calculation of the final time and pad values as well as their deviations they all have to be divided by the total charge of the cluster. This division is done in single-precision floating point representation. As division is an expensive operation in hardware in terms of resource usage or throughput, only a single but fully pipelined divider instance is integrated into the design and shared for all division operations.

4. The FastClusterFinder in Run 2
The original FastClusterFinder as used during Run 1 was implemented on a XILINX Virtex-4 FPGA and was running at 133 MHz [14]. Two instances were implemented per RORC. With a DDL clock of 40 MHz this corresponds to a clock scaling factor of 3.3 that proved to be sufficient for Run 1. In order to use the same algorithm in the C-RORC for Run 2 the code was ported to the Virtex-6 FPGA. The input interface from the DDL could be reused while the output interface required some changes due to the transition of the host interface from PCI-X to PCI Express. Technological advances in the FPGA floating point implementation now provide a bit-wise identical output compared to a software reference implementation while the Virtex-4 implementation was matching only up to only the second least significant bit. This greatly simplifies the verification of the implementation for Run 2. The port of the FastClusterFinder for readout of the RCU1 was in use in the first phase of Run 2 until end of 2015 with six cluster finder instances per C-RORC.

The installation of the TPC readout electronics upgrade with the RCU2 beginning of 2016 imposed several changes also to the HLT FastClusterFinder implementation. The biggest implication is the increased readout bandwidth via the DDL from below 160 MB/s with the RCU1 to around 310 MB/s for the RCU2. This roughly doubles the maximum data rate from the TPC and the cluster finder has be able to handle that without throttling the readout. Several parts of the FastClusterFinder implementation were adjusted, extended with additional register stages and partly rewritten in order to be able to handle the expected data rates. The RCU2 FPGA has more resources available than its predecessor so the branch merging can already be done there. Data from the RCU2 is therefore already fully ordered in contrast to the interleaved readout scheme as with the RCU1. This also means, that the secondary buffer in the Channel Merger is not needed anymore for the RCU2. On the other hand, removing this buffer from the
C-RORC firmware means that only RCU2 data can be handled. This is a strong limitation for the HLT system level verification mechanisms using C-RORC data replay. In order to keep the HLT as flexible as possible a common extended firmware was developed that supports both link rates and both RCU versions. Table 2 lists the resource usage of the initial and the extended common version of one instance of the FastClusterFinder implementation.

| Implementation       | RCU1 FCF | common ext. FCF |
|----------------------|----------|-----------------|
| Flip-Flops           | 5169     | 8465            |
| Look-Up Tables       | 4274     | 4250            |
| DSPs                 | 8        | 8               |
| Block-RAMs           | 23       | 25              |
| Frequency            | 159.4 MHz| 312.5 MHz       |

Table 2. Cluster finder resource usage comparison of the initial FCF port for RCU1 readout and the extended common version for the RCU1 + RCU2 supporting twice the throughput.

The RCU2 uses the available DDL bandwidth more efficiently than the RCU1. For that reason the FastClusterFinder clock scaling factor of 3.3 as used during Run 1 is not sufficient to handle the RCU2 data without temporarily stalling the readout. Verification with around five million sub-events from different runs throughout 2015 and 2016 confirm this. The Channel Merger state machine processes cluster candidates from high entropy events in this case too slow with respect to the input data rate. As a result the FIFOs run full and ultimately pause the detector readout.

The left graph in figure 5 shows measurements with the same data set and a clock scaling factor of four. With this implementation the Channel Merger is idle on average 43% of the time, leaving only a small fraction of comparably small events at a full load scenario. The work load of these events is mitigated by the FIFOs between the different processing stages. This implementation can handle the full DDL input bandwidth without throttling the readout. Implementing this factor four in the firmware requires each of the six FastClusterFinder instances per C-RORC to run at 312.5 MHz, which demands a carefully designed firmware. The cluster finder reduces the input data volume on average by a factor of around 1.5, however the output from individual sub-events may still be above the input data size as shown in figure 5 on the
right. This requires the output interface to the DMA engine to have a higher bandwidth than the input interface.

On top of doubling its throughput, the FastClusterFinder was extended with error detection and reporting mechanisms. Erroneous TPC channels can be spotted immediately while running. Another verification option is an additional readout path for the raw data before it enters the cluster finder. A secondary data path with its own DMA engine optionally transfers all or a configurable fraction of the raw TPC data to the host machine to analyze it. This was used to verify the functionality of the RCU2 branch merging mechanism, because the initial data ordering can not be reconstructed from the cluster finder output.

The branch merging in the RCU2 firmware also brings improvements to the physics performance of the FastClusterFinder. With the RCU1, clusters close to the branch boundaries could not be merged with clusters on neighboring pads from the other branch because of the interleaved readout. With the branch merging in the RCU2 this limitation is gone, leading to around 2–3% less (fake border) clusters. In cooperation with TPC offline experts a tagging mechanism for split clusters in the FastClusterFinder as well as an improved noise resilience of the peak finding mechanism is being evaluated to improve dE/dx calculations.

Implementation changes and especially algorithmic changes to the FastClusterFinder have to be verified very carefully because the original raw data is not stored in physics runs. In this context, the functionality of the hardware implementation has to be checked against an approved software reference. In order to simplify this verification a dedicated verification firmware variant with the FastClusterFinder was developed. With a combination of existing modules from the HLT-OUT and the HLT-IN firmware variants, raw data can be streamed from the host into the FPGA via DMA, processed with the FastClusterFinder and streamed back to the host via DMA for storage of the results or comparison against a reference. This method is significantly faster than using the regular input firmware with the DDR3 replay mechanism and was used to verify the processing of millions of sub-events in a reasonable time with only a single FPGA. In addition, the overall output of the HLT including the cluster finding was verified by TPC offline experts by comparing the offline analysis results from raw TPC data with those deduced from the corresponding HLT compressed data as input.

The FastClusterFinder in the HLT input firmware is designed to handle the full DDL input bandwidth, which corresponds to around 310 MB/s for the RCU2. Processing the data already in hardware is not only convenient because it is going through the FPGA anyway, it also provides a significant performance improvement compared to a raw readout in combination with software based cluster finding. Figure 6 shows processing time measurements of the software...
reference implementation with around 260k Pb–Pb events from 2015 being processed repeatedly on an HLT cluster node. The software reference can handle on average around 15 MB/s. This however cannot be compared directly with the 310 MB/s from the hardware implementation. The hardware has to be able to handle the full input bandwidth in order not to stall the readout at any time even though there are comparably long gaps without input data between successive events. A software implementation with large input buffers only has to be able to handle the average data rate across several events. Taking this into consideration results in a CPU core equivalent of around two to five cores per hardware cluster finder instance depending on beam type and event rate. With 216 hardware cluster finder instances in the HLT this adds up to considerable savings in the required CPU processing power.

5. Conclusion

The C-RORC is in use in the HLT since the start of Run 2 in June 2015. The combination of firmware and software integrated into the HLT data processing framework has proven to be a reliable interface to the ALICE experimental data at different link rates. The hardware platform made it possible for detectors to increase their readout bandwidth for Run 2 and it is actively used for developments on the readout system for Run 3. The hardware cluster finding on TPC raw data in the C-RORC already reduces the data volume and provides significant savings on the required CPU processing power of the HLT. In combination with the CPU based data compression, raw TPC data is discarded in the DAQ in favor of the HLT results to reduce the amount of data to be saved on permanent storage. This data compression allows the experiment to collect more statistics with the same storage quota. With the upgrade of the TPC readout electronics to the RCU2, the cluster finder could be extended to double the initial processing bandwidth. A generic firmware implementation that supports data from both the old and the new TPC readout electronics transparently enables the HLT to continue to use any TPC data for system level benchmarks and tests with the C-RORC data replay. The development of a dedicated verification firmware with the extended cluster finder made it possible to verify the implementation also against large data sets and confirmed its performance in hardware. The HLT cluster finder is continuously being improved. Current evaluations are targeting split cluster tagging and improved noise resilience to the peak finding step.

References

[1] ALICE Collaboration 2008 The ALICE Experiment at the CERN Large Hadron Collider JINST 3 S08002
[2] F Carena et al 2014 The ALICE data acquisition system NIM Phys. Res. A 741 130–162
[3] Krzewicki M et al. 2017 ALICE HLT Run 2 performance overview JPCS
[4] Carena F et al. 2015 DDL, the ALICE data transmission protocol and its evolution from 2 to 6 Gb/s JINST 10 C04008
[5] Engel H and Kebschull U 2013 Common Read-Out Receiver Card for ALICE Run2 JINST 8 C12016
[6] Borga A et al. 2015 The C-RORC PCIe card and its application in the ALICE and ATLAS experiments JINST 10 C02022
[7] Costa F et al. 2016 2016 IEEE-NPSS Real Time Conference (RT) The ALICE C-RORC GBT card, a prototype readout solution for the ALICE upgrade pp 1–5
[8] Lehrbach J et al. 2017 ALICE HLT Cluster operation during ALICE Run 2 JPCS
[9] Rohr D et al. 2017 Improvements of the ALICE HLT data transport framework for LHC Run 2 JPCS
[10] ALICE Offline Group ALICE Off-line Project, http://aliweb.cern.ch/Offline/ accessed Jan. 2017
[11] Alme J et al. 2010 The ALICE TPC, a large 3-dimensional tracking device with fast readout for ultra-high multiplicity events NIM Phys. Res., A 622 316–367. 55 p
[12] Rohr D et al. 2017 GPU-accelerated track reconstruction in the ALICE High Level Trigger JPCS
[13] Alme J et al. 2013 RCU2 - The ALICE TPC readout electronics consolidation for Run2 JINST 8 C12032
[14] Alt T 2017 An FPGA based pre-processor for the ALICE High Level Trigger Ph.D. thesis Goethe-University Frankfurt