Eur Phys J C **33**, s01, s1041–s1043 (2004)
Digital Object Identifier (DOI) 10.1140/epjcd/s2004-03-1806-1

EPJ C direct
electronic only

The high-level trigger of ALICE

H. Tilsner[1], T. Alt[2], K. Aurbakken[2], G. Grastveit[2], H. Helstrup[3], V. Lindenstruth[1], C. Loizides[4], J. Nystrand[2], D. Roehrich[2], B. Skaali[5], T. Steinbeck[1], K. Ullaland[2], A. Vestbo[2], T. Vik[5], presented by H. Tilsner

[1] Kirchhoff Institute for Physics, University of Heidelberg, Germany
[2] Department of Physics, University of Bergen, Norway
[3] Bergen College, Norway
[4] Institute of Nuclear Physics, University of Frankfurt, Germany
[5] Department of Physics, University of Oslo, Norway

Received: 19 October 2003 / Accepted: 10 March 2004 /
Published Online: 31 March 2004 – © Springer-Verlag / Società Italiana di Fisica 2004

Abstract. One of the main tracking detectors of the forthcoming ALICE Experiment at the LHC is a cylindrical Time Projection Chamber (TPC) with an expected data volume of about 75 MByte per event. This data volume, in combination with the presumed maximum bandwidth of 1.2 GByte/s to the mass storage system, would limit the maximum event rate to 20 Hz. In order to achieve higher event rates, online data processing has to be applied. This implies either the detection and read-out of only those events which contain interesting physical signatures or an efficient compression of the data by modeling techniques. In order to cope with the anticipated data rate, massive parallel computing power is required. It will be provided in form of a clustered farm of SMP-nodes, based on off-the-shelf PCs, which are connected with a high bandwidth low overhead network. This High-Level Trigger (HLT) will be able to process a data rate of 25 GByte/s online. The front-end electronics of the individual sub-detectors is connected to the HLT via an optical link and a custom PCI card which is mounted in the clustered PCs. The PCI card is equipped with an FPGA necessary for the implementation of the PCI-bus protocol. Therefore, this FPGA can also be used to assist the host processor with first-level processing. The first-level processing done on the FPGA includes conventional cluster-finding for low multiplicity events and local track finding based on the Hough Transformation of the raw data for high multiplicity events.

PACS. 07.05.-t Computers in experimental physics – 07.05.Hd Data acquisition: hardware and software – 29.85.+c Computer data analysis

1 The HLT architecture

High-level trigger systems (HLT) are necessary in order to cope with the anticipated data rate in future experiments. The main tracking detector of ALICE will be a large Time Projection Chamber (TPC), which will produce a data volume of up to 75 MByte per event in central Pb-Pb collisions. With an anticipated event rate of about 200 Hz the data rate is approximately 15 GByte/s. In contrast, the maximum affordable bandwidth to the mass storage system is only about 1.2 GByte/s, which is one order of magnitude less than the data rate of the TPC only. Taking into account the rate of interesting physics triggers like jets, open charm, and e^+e^- and the net physics information contained in one raw event the set limit of 1.2 GByte/s is more than adequate in order to record all events. However, the key is in the selection of the relevant information. A detailed discussion about the physics applications of the HLT can be found in [1]. As a consequence, a system is needed which is able to reduce the data volume online by processing the events doing pattern recognition and simple event reconstruction in order to select interesting (sub)events or to compress data efficiently by modeling techniques [2], [3]. This is the task of the High-Level Trigger, a massive parallel computing system. The system will consist of a farm of clustered SMP nodes based on off-the-shelf PCs connected with a high bandwidth low overhead network.

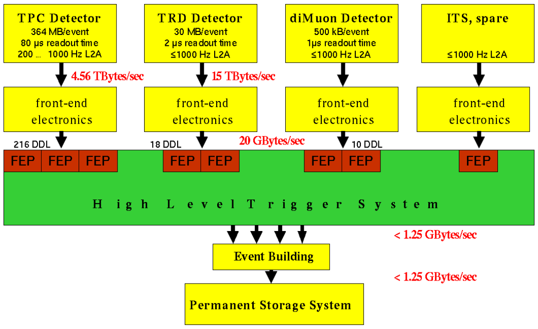

Fig. 1. Block diagram of the overall HLT architecture

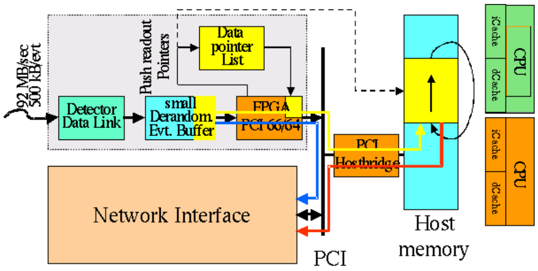

Fig. 2. Overview of the HLT Front End Processor architecture

The architecture of the HLT is determined by the detector hierarchy of ALICE, shown in Fig. 1. The independently operating detectors are synchronized by the trigger signal and ship their data via an optical data link to the so called *Front End Processors* (FEP). Overall, the HLT receives zero suppressed raw data from about 250 detector links, aggregating 25 GByte/s. The FEP are commercial off-the-shelf computers equipped with the *Read Out Receiver Card* (RORC), a PCI card on which a detector link mezzanine card is mounted and which subsequently can be mounted in any computer supporting PCI. The main memory of the computer functions as the event buffer, implementing the most cost effective memory. Figure 2 shows an overview of the architecture of the HLT Front End Processor.

The RORC is the interface between the front end electronics on the detectors and the HLT computing cluster. These custom PCI cards receive the raw data coming from the detectors and forwards them to the main memory of the host computer. This data flow is easily implemented by first reserving a large memory block in the host and then filling it with direct memory access (DMA) transfers. These transfers are initiated by a DMA engine which is implemented on the RORC. The availability of free buffer space can be implemented by pointer lists, stored in small FIFO memories on the PCI card. All control logic of the RORC is implemented in an FPGA which is also responsible to handle the PCI bus protocol. Despite its functionality as a PCI device, the FPGA is also used to assist the host processor with first level processing while the data is being transfered to the memory of the corresponding node.

Data processing on the HLT computation cluster is parallel. Each processing stage will perform its task on a well-identified subset of data and all processors run almost independently of each other. The data transport between the different analysis processes running on different nodes is handled and orchestrated by a software framework which is based on the publisher-subscriber principle [4]. A subscribing process informs a particular (data-)publishing process that it is interested in its data. From this point on the publisher will announce any new data (e.g. cluster or track parameters) to the interested subscribers. In order to be as efficient as possible the large data payload itself will not be communicated between different processes. Instead, only a descriptor of the data including a reference to the actual data in a shared memory segment will be sent while the data itself remains untouched.

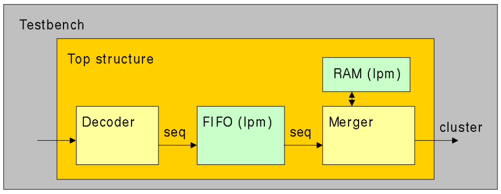

Fig. 3. Block diagram of the FPGA cluster finder

2 Online event reconstruction

In order to derive a trigger decision the HLT must be able to perform a detailed analysis of the incoming events at the anticipated event rate of 200 Hz. The most crucial part in this analysis is fast pattern recognition of the TPC, which delivers by far the most data. Depending on the event multiplicity, which is expected to be in the range of $dN/dy = 1000$ to 8000, this is realized in two different ways. For events with low multiplicity ($dN/dy < 4000$) a sequential feature extraction, consisting of cluster finding followed by a track finder, is foreseen. This scheme is working on space points. For high multiplicity events this approach can not be used since there is a large fraction of overlapping clusters which leads to a reduced tracking efficiency. For those events an iterative feature extraction scheme working directly on raw data is planned. This includes a Hough Transform as tracklet finder followed by a detailed analysis and deconvolution of the charge clusters, using the found tracklets as starting point.

The time needed to analyze one event with dN/dy of 4000 is estimated to be about 12 s. In order to be able to run at an anticipated event rate of 200 Hz to date 2400 CPUs are necessary. The estimated computing requirements for high multiplicity events are two orders of magnitude higher, resulting in a rather unrealistic number of CPUs. This problem can be solved by using the FPGA for initial data processing. The incoming raw data are searched for clusters or track candidates by the FPGA cluster finder or Hough Transform respectively. The results are transfered via the PCI bus to the HLT computing nodes memory.

2.1 FPGA cluster finder

The FPGA implementation of the cluster finder is shown in Fig. 3. The *Decoder* processes the incoming zero suppressed raw data. The input data stream consists of time ordered sequences of 10-bit ADC values. Furthermore, the *Decoder* accumulates the total charge of the sequences and calculates the center of gravity in time direction. Since the individual ADC values are not needed after the calculations they are not stored. The second processing unit, the *Merger*, reads the sequences and merges sequences on adjacent pads. The output of the *Merger* is the coordinate of the clusters. A detailed description of the FPGA cluster finder can be found in [6].

For verification purposes the system was simulated using a test bench. Simulated raw data was sent to the *Decoder*. Found clusters of the *Merger* circuit were compared

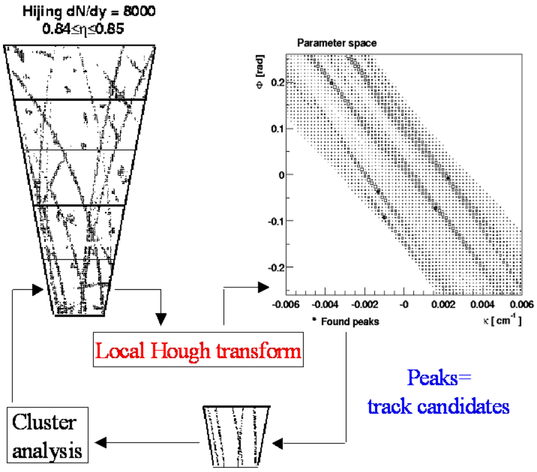

Fig. 4. Online data analysis for high multiplicity events using Hough Transform

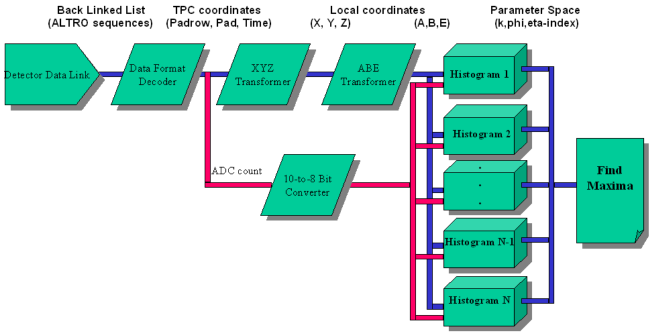

Fig. 5. Data flow diagram of the planned Hough Transform co-processor

to results obtained by a C++ code running the behavioral model of the cluster finder algorithm on the same data set. The found clusters showed a very good agreement.

2.2 The Hough transform

The Hough Transform is a standard tool in image analysis which allows recognition of global patterns in image space by recognition of local patterns in a transformed parameter space. The basic idea is to find curves that can be parameterized in a suitable parameter space. In its original form one determines a curve in parameter space for a given signal, corresponding to all possible tracks with a given parametric form it could possibly belong to [7]. All such curves, belonging to the different signals, are drawn in parameter space which is discretized and the entries are stored in a histogram. In this histogram one track is represented by a peak at the corresponding parameters.

In ALICE, the local track model is a helix. In order to simplify the transformation the detector volume is divided into sub-volumes in pseudo rapidity η. If the analysis is restricted to tracks originating from a common interaction point the circular track in the η sub-volume is characterized by the emission angle with the beam axis and the curvature [8]. After the transformation into this parameter space, each active pixel (ADC value above threshold) is represented by a sinusoidal line extending over the whole range in parameter space. All corresponding bins in the histogram are incremented with the ADC value of the transformed pixel. The superposition of all these point transformations produce a maximum at the circle parameters of the track. The track recognition is now done by searching for local maxima in the parameter space, see Fig. 4. Once the track parameters are known, cluster finding on the raw data can be performed by a straightforward unfolding of the clusters [9].

The data flow diagram of the Hough Transform co-processor is shown in Fig. 5. In the *Data Format Decoder* the data will be decoded into a quadruple of pad-row, pad, time, and ADC value. Before further processing the ADC values will be converted nonlinearly from a 10-bit resolution to 8-bit, resulting in a constant relative accuracy over the whole dynamic range and a reduction of the event size. The conversion is accomplished with a fixed look up table containing 1024 entries. In the first transformation (XYZ), the pad-row, pad, and time coordinates are transformed into local Cartesian coordinates, using two additional look up tables. To optimize for hardware implementation the data is transformed into another coordinate system (ABE), reducing the number of trigonometric calculations needed to fill the histograms. Finally, the histograms are searched for peaks above a given threshold. These peaks represent track candidates and for each of these candidates a triplet representing curvature, emission angle, and an index of the pseudo-rapidity interval is send via the PCI bus to the host PC for further processing. At present, a behavioral model of the Hough Transform exists. The model was verified with simulated event data and the results compared to the results of the online C++ code.

Acknowledgements. This work was supported by the German Federal Ministry of Education and Research (06HD955I).

References

1. *ALICE High-Level Trigger Conceptual Design*, http://www.kip.uni-heidelberg.de/ti/L3/concept.html
2. V. Lindenstruth et al.: Proceedings of the 13th IEEE-NPSS Real Time Conference, Montreal, Canada, 2003
3. J. Berger et al.: Nucl. Instr. Meth A **489**, 406 (2002)
4. T. Steinbeck et al.: Proceedings of CHEO03, La Jolla, California, 2003
5. G. Grastveit et al.: Proceedings of CHEP03, La Jolla, California, 2003
6. G. Grastveit: *VHDL-implementation of the Cluster Finder algorithm for use in ALICE* Master Thesis University of Bergen, Norway, 2003
7. P.V.C. Hough: International Conference on High Energy Accelerators and Instrumentation, CERN, 1959
8. D. Brinkman et al.: Nucl. Instr. Meth. A **354**, 419 (1995)
9. U. Frankenfenld et al.: Proceedings of the 12th IEEE-NPSS Real Time Conference, Valencia, Spain, 2002