FELIX-Based Readout of the Single-Phase ProtoDUNE Detector

Andrea Borga, Eric Church, Frank Filthaut, Enrico Gamberini, Paul de Jong, Giovanna Lehmann Miotto, Frans Schreuder, Jörn Schumacher, Roland Sipos, Milo Vermeulen, Kevin Wierman, and Lynn Wood

***Abstract*— Large liquid argon (LAr) time projection chambers (TPCs) have been adopted for the Deep Underground Neutrino Experiment (DUNE) experiment's far detector, which will be composed of four 17-kton detectors situated 1.5 km underground at the Sanford Underground Research Facility. This represents a large increase in scale compared to existing experiments. Both single- and dual-phase technologies will be validated at CERN, in cryostats capable of accommodating full-size detector modules, and exposed to low-energy charged particle beams. This program, called ProtoDUNE, also allows for extensive tests of data acquisition strategies. The Front-End LInk eXchange (FELIX) readout system was initially developed within the ATLAS collaboration and is based on custom field-programmable gate array (FPGA)-based Peripheral Component Interconnect Express input/output cards, connected through point-to-point links to the detector front end and hosted in commodity servers. FELIX will be used in the single-phase ProtoDUNE setup to read the data coming from 2560 anode wires organized in a single anode plane assembly (APS) structure. With a continuous readout at a sampling rate of 2 MHz, the system must deal with an input rate of 96 Gb/s. An external trigger will preselect time windows of 5 ms with interesting activity expected inside the detector. Event building will occur for triggered events, at a target rate of 25 Hz; the readout system will form fragments from the data samples matching the time window, carry out lossless compression, and forward the data to event building nodes over 10-Gb/s Ethernet. This paper discusses the design and implementation of this readout system as well as the first operational experience.**

***Index Terms*— Data acquisition (DAQ), data collection, high energy physics instrumentation computing.**

I. Introduction

PROTODUNE-SP [1] is the single-phase Deep Underground Neutrino Experiment (DUNE) far detector prototype that is under construction and will be operated at the CERN Neutrino Platform (NP) starting in 2018. ProtoDUNE-SP represents a crucial part of the DUNE effort toward the construction of the first DUNE 10-kton fiducial (17 kton total) liquid argon (LAr) mass far detector module. With a total LAr mass of 0.77 kton, it represents the largest monolithic single-phase LAr time projection chamber (TPC) built to date. It is housed in an extension to the EHN1 hall in the North Area, where the CERN NP is providing a new dedicated charged-particle test beamline. ProtoDUNE-SP aims to take its first beam data in the second half of 2018.

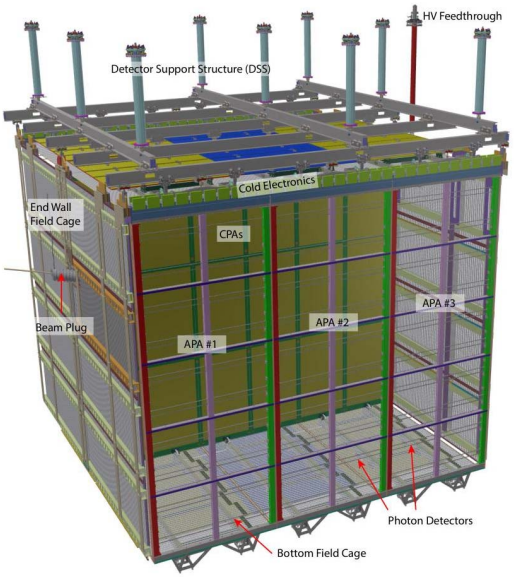

Fig. 1. Major components of the ProtoDUNE-SP TPC.

The ProtoDUNE-SP TPC, illustrated in Fig. 1, comprises two drift volumes, defined by a central cathode plane that is flanked by two anode planes, each at a distance of 3.6 m, and a field cage (FC) that surrounds the entire active volume. The active volume is 6 m high, 7 m wide, and 7.2 m deep (along the drift direction). Each anode plane is constructed of three adjacent anode plane assemblies (APAs) that are each 6 m high by 2.3 m wide in the installed position. Each APA consists of a frame that holds three parallel planes of induction and collection wires on each of its two faces for a total of 2560 channels; the wires of each plane are oriented at different angles with respect to those making up the other planes of the same face, to enable 3-D reconstruction.

The readout of the TPC wires, prior to being received by the data acquisition (DAQ) system, consists of cold electronics (CE) mounted on the APAs inside the cryostat and the warm

Manuscript received December 11, 2018; revised February 19, 2019; accepted March 5, 2019. Date of publication March 18, 2019; date of current version July 16, 2019.

A. Borga, F. Schreuder, and M. Vermeulen are with Nikhef, 1098 XG Amsterdam, The Netherlands.

E. Church, K. Wierman, and L. Wood are with the Pacific Northwest National Laboratory, Richland, WA 99354 USA.

F. Filthaut is with Department of High Energy Physics, Radboud University, 6525 AJ Nijmegen, The Netherlands, and also with Nikhef, 1098 XG Amsterdam, The Netherlands (e-mail: f.filthaut@science.ru.nl).

E. Gamberini, G. Lehmann Miotto, J. Schumacher, and R. Sipos are with CERN, Geneva, Switzerland.

P. de Jong is with Nikhef, 1098 XG Amsterdam, The Netherlands, and also with the Institute of Physics, University of Amsterdam, 1012 WX Amsterdam, The Netherlands.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2019.2904660

electronics outside the cryostat on the flange. CE data are received on the warm interface boards (WIBs) which are situated on the top of the flanges. Each WIB multiplexes the data to four 4.8 Gb/s (or two 9.6 Gb/s) lines that are sent over optical fibers to the DAQ. Two systems are used to receive data from the WIBs. The baseline system uses reconfigurable computing elements (RCEs) [2] which are housed in industry standard Advanced Telecommunications Computing Architecture shelves and are used to read out 5 of 6 APAs. The alternative system described here allows the collaboration to explore less costly solutions by making use of recent advances in commodity computing; it is based on the Front-End LInk eXchange (FELIX) [3] technology and is used to receive the data from the remaining APA.

II. FELIX-Based Readout

The main driver of the FELIX concept is the firm belief that a thin interface (featuring a minimal amount of custom hardware and software) managing the interaction with detector front-end links and injecting data into commodity servers at an early stage of the DAQ chain provides the flexibility that is required for the optimization and maintenance of long-term and long-lifetime systems.

The FELIX input/output (I/O) card is built around the Xilinx Ultrascale XCKU115 field-programmable gate array (FPGA). It accommodates up to 48 optical fibers and provides a simple point-to-point interface to the detector front end, receiving data from the front end using the widely used 8b/10b encoded serial protocol at 9.6 Gb/s (it also features a 4.8 Gb/s link to the front end, which however is not used in the present application). Encoding protocols with smaller overhead exist, such as 64b/66b, but are not readily available for FPGAs used in front-end electronics. Using the Peripheral Component Interconnect Express (PCIe) format allows all data to be transferred to the host memory. This solution leverages the fast evolution of multicore server performance, the possibility of using the large available host memory and an optimal choice of high-performance networking for data dispatching. It is a more general trend in high energy physics DAQ to move toward PCIe-based solutions. For instance, the ALICE and LHCb experiments are developing general readout PCIe boards for their upcoming upgrades [4], [5].

A. Topology of the Readout System

The FELIX I/O card interfaces with its host PC through 16-lane PCIe Gen3. It transfers the incoming WIB data directly into the host PC's memory using a continuous direct memory access (DMA) transfer accomplished through the Wupper [6] engine. The host PC runs a software process, called felixcore [7], which publishes selected data to any client subscribing to it, based on link identifiers. In ProtoDUNE, the clients to the felixcore application are the BoardReader processes, which are part of the artdaq [12] framework used for the ProtoDUNE DAQ dataflow system.

From a hardware point of view, the FELIX and BoardReader hosts in use are based on a dual-socket Intel Xeon Processors (E5-2620 v4 2.1 GHz), equipped with a Mellanox Technologies MT28800 Family [ConnectX-5 Ex] 2 × 100-Gb/s NIC.

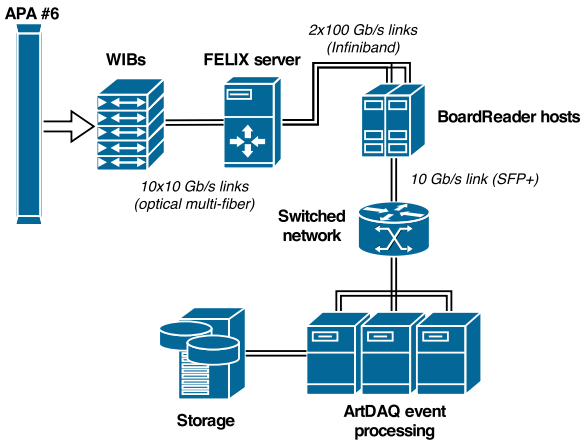

Fig. 2. Overview of the FELIX DAQ chain for ProtoDUNE-SP.

A single FELIX I/O card receives data from a whole APA over ten links and is hosted in one server. Final performance benchmarking is still needed to establish whether a single host for the BoardReader processes receiving data from the FELIX will be sufficient, instead of requiring a second host as will be assumed in the following. The output of selected data toward the DAQ event builder is carried out over 10-Gb/s Ethernet.

B. WIB Data Volume and Structure

In the ProtoDUNE-SP context, the readout system must be able to cope with the bandwidth of data transfers from the WIBs. The WIBs will send data to FELIX at a 2-MHz frame rate per optical link. Each frame contains 120 32-bit words, leading to a payload rate of 7.68 Gb/s; the additional overhead from 8b/10b encoding leads to a total transfer rate of 9.6 Gb/s. Each link represents 256 channels; FELIX, therefore, needs to read from ten input links, corresponding to a total payload rate of 76.8 Gb/s. These data are split and sent to two BoardReader hosts via two 100-Gb/s Infiniband [10] links, which were chosen to have a large safety margin (in contrast to the alternative of using two 40-Gb/s links).

The readout system must buffer the incoming data until a data request is received from the event building farm and then transfer the data contained in a 5-ms time window, centered around the trigger timestamp. The DAQ system is designed for a target trigger rate of 25 Hz. An overview of the described readout system is seen in Fig. 2.

In the case of FELIX, the WIBs combine data from two front-end motherboards (FEMBs) into one frame. Since each FEMB sends out 128 channels at a rate of 2 MHz, FELIX frames contain 256 channel values in total, divided over four blocks (two for each FEMB) that are each assembled by a COLd DAta Transmission application specific integrated circuit (COLDATA). The channel values each take up 12 bits but are cut up and rearranged to be byte aligned. This configuration ensures that if an individual FEMB should fail, the WIB will continue sending fixed-size fragments toward the FELIX.

Apart from the analog-to-digital converted (ADC) values, a WIB frame (Fig. 3) contains a WIB header as well as four COLDATA headers. As the name suggests, the information in the WIB header is added by the WIB. It contains identifier data

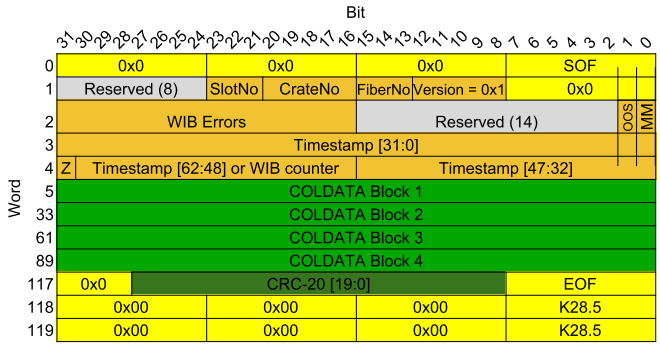

Fig. 3. Simplified schematic of a WIB frame as it is sent to FELIX. The main payload is contained within the four COLDATA blocks. Words 1 through 116 are passed on to the FELIX host, decreasing the frame size to 464 bytes.

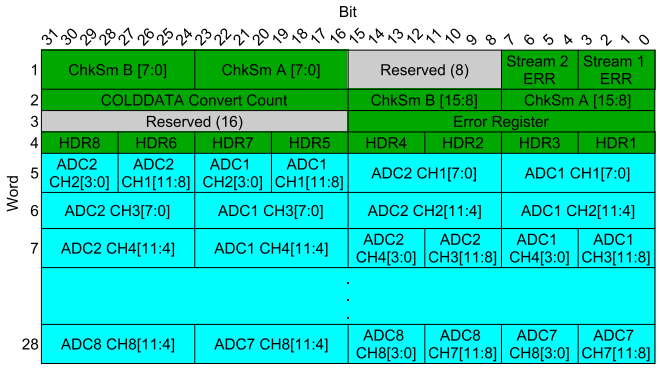

Fig. 4. Simplified schematic of a COLDATA block. One block contains 64 front-end channel values.

such as the APA number, WIB number, and WIB output fiber number that the data originated from. Together, these uniquely identify the origin of the data encompassed by the frame. In addition, there are several error fields available for the WIB to pass error flags along. Finally, the WIB header contains a 63-bit timestamp, which is generated from a detector wide timing system and increments every 20 ns. From one frame to the next, this timestamp is expected to increment by 25 such intervals, given the 2-MHz frame rate.

The channel values in each frame are divided into four COLDATA blocks, in which each contains 64 values and has a COLDATA header each. One such block is shown schematically in Fig. 4. The COLDATA header contains additional error fields as well as a COLDATA convert count, which is expected to increment between consecutive frames from the same source. These counters are only supposed to be identical when coming from the same FEMB and can, therefore, differ within a frame, which contains data from two separate FEMBs. (The checksums also displayed in Fig. 4 are not meaningful anymore by the time they are sent from the WIB.)

Each frame contains a CRC-20 checksum generated by the WIB. It is used by FELIX to verify the frame's data integrity and is then discarded before the frame is passed to the FELIX host PC, with a flag set in the frame trailer to indicate an error.

C. FELIX Firmware

The FELIX firmware design has been described in [13] and a simplified version is shown schematically in Fig. 5. In the ProtoDUNE context, FELIX operates in a FULL mode, which

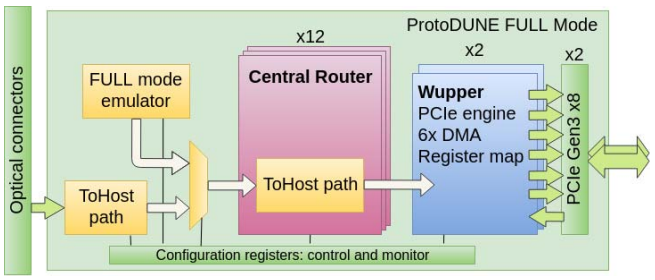

Fig. 5. Main FELIX firmware logical blocks in FULL mode.

features unidirectional links from the front-end systems that allow for a data stream of 9.6 Gb/s per link. The central router communicates the incoming data to the Wupper PCIe engine according to a user-defined configuration. The Wupper, in turn, handles the DMA toward the host. For testing purposes, a FULL mode emulator can be used. It typically cycles over a small predefined buffer and sends its contents to the central router at a rate representative of the final system. In order to be able to sustain the high rate of incoming frames (2 MHz) and the high throughput requirements, the firmware is being modified in several parts, specifically for ProtoDUNE.

1) Increasing the packing factor and DMA payload size lowers the rate of memory copies within the host, where several DMA payloads must be stitched back together. A larger packing factor also increases the space and time requirements from the host memory access perspective, inducing backpressure on the firmware buffers. As a tradeoff, a frame packing factor of 6 is currently used as the baseline.
2) Data from each individual link are DMA'd into different memory areas on the host.

These two changes allow for considerable simplification of the felixcore software running on the FELIX host. For instance, all block processing pipelines involving substantial memory copies were removed from the original felixcore implementation, having opted instead to publish directly from the DMA buffer. A more detailed description of the FELIX-specific felixcore software can be found in Section III.

D. Felixcore Application

The felixcore application is in charge of routing data from the detector to networked software clients [7]. The generic ATLAS version has been designed with the aim of being unaware of the data that it routes, supporting bidirectional traffic between the detector front end and software back end, and allowing for a flexible demultiplexing of data aggregated on a single physical link. It supports two main networking communications standards (transmission control protocol/internet protocol and Infiniband), integrated into the NetIO messaging layer.

In ProtoDUNE, only unidirectional traffic is used (from the detector to host memory) and the data fragments have a fixed size. Focusing on these two features, a simplified felixcore version was implemented that resembles the scatter-gather computing technique. Generic input link identifier threads and data copy pipelines were removed, and a dedicated network publisher thread was introduced for each physical link.

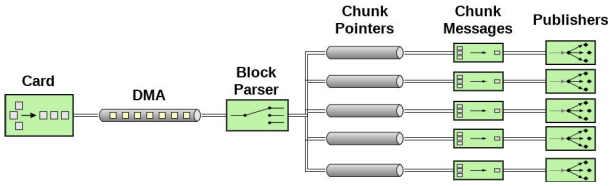

Fig. 6. Overview of the felixcore publisher application.

The card-reader thread is responsible for the DMA payload (block) parsing to extract the location of the fragments for the new network messages. From these fragment pointers, the publishers assemble network messages and send these messages to the subscribed clients. The data handling is shown schematically in Fig. 6. In order to completely CPU offload, the required network operations and to utilize the zero-copy remote direct memory access feature of the network interfaces, the Infiniband back end is used.

Thanks to the firmware modifications for reducing processing needs and this highly use-case specific implementation of the publisher application, the cards DMA buffer parser and the publisher threads are able to keep up with the continuous stream of aggregated WIB frames without problems. If congestion occurs downstream of the felixcore application such that the capacity of buffers which allow for the absorption of instantaneous congestion is exceeded, data are dropped. This may cause data loss and is flagged by error messages at the felixcore and BoardReader applications. In addition, empty fragments may be sent to the event builder if required. It should nevertheless be noted that the system is very predictable due to the fixed rate of incoming data and that this type of congestion does not occur until well beyond the design trigger rate.

III. DAQ Software Layer

The ProtoDUNE use case has challenging requirements for the software downstream of the FELIX readout, since it has to support the full data rate, perform trigger matching and lossless data compression. The experience from previous experiments such as MicroBooNE [9] suggests that for a sufficiently low electronics noise level, a compression factor of 4 is feasible for the ProtoDUNE-SP data. Nevertheless, general-purpose compression algorithms are less efficient, especially if data are not reorganized prior to compression.

A. BoardReader Implementation

The BoardReader implementation for the FELIX-based readout integrates the NetIO [11] messaging layer, which subscribes to the felixcore application, into the artdaq [12] framework. The latter is used as ProtoDUNE's DAQ software framework. In particular, each BoardReader subscribes to data from one WIB link. Its main tasks are to:

1) store all incoming data into a circular buffer;
2) receive trigger information (timestamp and event identifier);
3) form a DAQ fragment by matching the data in the circular buffer and the timestamp of the trigger (5 ms of data around the trigger);

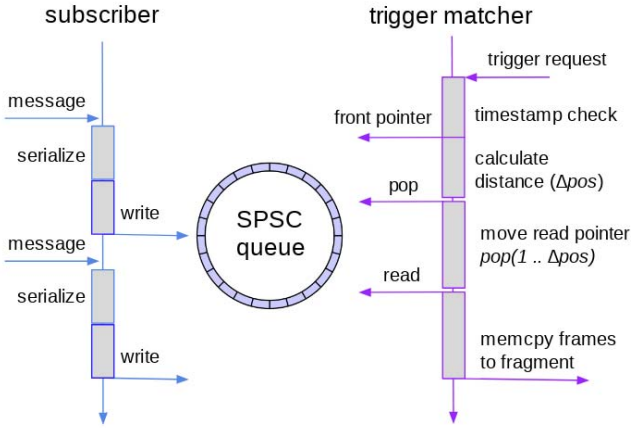

Fig. 7. Logic of producer and consumer threads and their utilization of the queue.

4) compress the content of the DAQ fragment (as discussed in more detail below);
5) pass on the DAQ fragment to the event builder.

In order to achieve the required 25-Hz readout rate as outlined in Section II-B, particular care has been put into the implementation of the customized part of the BoardReader application, avoiding as much as possible dynamic memory allocation. Internal elements are accessed through unique pointers and the access functionalities strictly avoid copies. Every link has dedicated subscriber threads that populate single-producer single-consumer (SPSC) queues, using the lock-free implementation from the Folly [8] library. An SPSC queue can be used for communication with a thread which services a hardware device (wait-free property is required), or when there are naturally only one producer and one consumer. As our solution has a single extractor (trigger-matcher) for each circular buffer, using a lock-free implementation is a straightforward choice. The thread flow and the utilization of the circular buffer are shown in Fig. 7.

The solution relies heavily on the C++11 standard library. Move semantics are used to avoid memory copies and to directly stream the data from the network interface card's ring buffer to the queues. This is denoted by "write" in Fig. 7. The use of condition variables and of a dedicated class that implements the pausable threads ensure the proper synchronization of the trigger matching threads, and the compliance with the internal state machine of artdaq. The implementation also focuses on flexibility, as the topology of the queues and links is scalable by the BoardReader configuration.

Using the described system, data output rates of nearly 50 Hz were reached during stress testing. This fits the design requirements comfortably, which are to output 9.28 Gb/s in the form of 5-ms trigger windows at a rate of 25 Hz. Nonetheless, there is still room for improvement, particularly in the context of extrapolating to the DUNE requirements.

B. Compression

Two constraints exist for lossless compression within the FELIX system. First, the compression factor must be effective. The target compression factor for the ProtoDUNE-SP data was

set to 4 and is incorporated in storage hardware projections. Second, the compression algorithm should keep up with the trigger rate. At 25 Hz, each BoardReader process has only 40 ms to handle its current batch of data. Since the data compression stage is the computationally most demanding one, a fast solution is essential.

The speed of the compression procedure is optimized using dedicated Intel QuickAssist Technology (QAT) [14] hardware. This employs the DEFLATE [15] algorithm by default, which consists of a sliding window compression and a Huffman compression stage. In the former, repeated bit strings are replaced with references to a previous occurrence. The latter then replaces frequently occurring bytes with shorter bit strings. This allows a reduction of the time required for the compression of one fragment's data to approximately 4 ms. For unmodified, frame-by-frame input data, the achieved compression factor is well below the desired factor of 4. This is due in part to the noncontiguous storage of ADC data for individual channels in subsequent frames, but more importantly to the fact that part of the ADC values is cut up in the frame data, as mentioned in Section II-B. The BoardReader process will, therefore, feature a data reordering stage, where the ADC values are reconstructed in 16-bit words and reordered so that the ADC values for all 10 000 subsequent digitization time slices are contiguous in memory. Using a software reordering stage and QAT hardware-accelerated compression stage has proven to be much faster than a software implementation of DEFLATE working directly on the data. The QAT hardware supports other compression algorithms; however, DEFLATE was chosen because of its favorable balance between compression factor and compression time. Using a simulated electronics noise of approximately four ADC counts [9] on average, a compression factor of 3.7 is reached.

IV. Conclusion

The ProtoDUNE-SP detector is a 770-ton LAr detector intended to validate the single-phase LAr TPC technology at the full scale of the DUNE experiment and expects to receive the beam from the CERN Super Proton Synchrotron accelerator in the second half of 2018. One of its six APAs, representing 2560 anode wires, will be read out using the FELIX system.

The FELIX readout system is based on the concept of having a thin interface between the front end of a detector and commodity hardware. The current FELIX I/O card receives 96 Gb/s of data over ten links and uses 16-lane PCIe Gen3 to copy it to the FELIX host PC's memory.

The input data rate can be sustained using a firmware and software modified from its original version used in the ATLAS experiment, with a separate DMA transfer for each input link's data, and using larger block size matched to the input frame size. This is combined with the use of scatter-gather techniques to send the data to BoardReader processes running on separate hosts.

These BoardReader processes must perform trigger matching and lossless data compression by a factor of 4. The requirements on the compression, which is the most computationally demanding step, have largely been met by reformatting the data in software and subsequently carrying out a hardware-accelerated compression. They are subsequently packaged in the artdaq fragment format and forwarded to the artdaq-based event building framework.

The present system meets the design goals of the ProtoDUNE experiment; improvements needed to meet the requirements of the future DUNE experiment are under investigation.

Acknowledgment

The authors would like to thank their colleagues in the DAQ group of the single-phase ProtoDUNE setup, within which this project is embedded, for their support in integrating the FELIX-based readout and the CERN openlab team for providing support for their hardware-accelerated compression studies. They would also like to thank the assistance and support from the FELIX developers team on the ATLAS experiment, without whom this project would not have been possible.

References

[1] B. Abi *et al.* (2017). "The single-phase protodune technical design report." [Online]. Available: https://arxiv.org/abs/1706.07081

[2] R. Herbst *et al.*, "Design of the SLAC RCE platform: A general purpose ATCA based data acquisition system," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, Nov. 2016, pp. 1–4. doi: 10.1109/NSSMIC.2014.7431254.

[3] J. Anderson *et al.*, "FELIX: A PCIe based high-throughput approach for interfacing front-end and trigger electronics in the ATLAS upgrade framework," *J. Instrum.*, vol. 11, no. 12, Dec. 2016, Art. no. C12023. doi: 10.1088/1748-0221/11/12/C12023.

[4] P. Durante, N. Neufeld, R. Schwemmer, U. Marconi, G. Balbi, and I. Lax, "100 Gbps PCI-express readout for the LHCb upgrade," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 4, pp. 1752–1757, Aug. 2015.

[5] A. Borga *et al.*, "The C-RORC PCIe card and its application in the ALICE and ATLAS experiments," *J. Instrum.*, vol. 10, no. 2, Feb. 2015, Art. no. C02022. doi: 10.1088/1748-0221/10/02/C02022.

[6] *OpenCores Project*. Accessed: Jun. 20, 2018. [Online]. Available: https://opencores.org/project/virtex7_pcie_dma/overview

[7] J. Schumacher, "Improving packet processing performance in the ATLAS FELIX project: Analysis and optimization of a memory-bounded algorithm," in *Proc. 9th ACM Int. Conf. Distrib. Event-Based Syst.*, 2015, pp. 174–180. doi: 10.1145/2675743.2771824.

[8] *Facebook Open-Source Library*. Accessed: Mar. 13, 2018. [Online]. Available: https://github.com/facebook/folly

[9] R. Acciarri *et al.*, "Noise characterization and filtering in the microboone liquid argon TPC," *J. Instrum.*, vol. 12, no. 8, 2017, Art. no. P08003. doi: 10.1088/1748-0221/12/08/P08003.

[10] "Introduction to infiniband," Mellanox Technologies, Sunnyvale, CA, USA, White Paper 2003WP, 2003. Accessed: Jun. 20, 2018. [Online]. Available: https://www.mellanox.com/pdf/white papers/ IB_Intro_WP_190.pdf

[11] J. Schumacher, C. Plessl, and W. Vandelli, "High-throughput and low-latency network communication with NetIO," *J. Phys., Conf. Ser.*, vol. 898, no. 8, 2017, Art. no. 082003. doi: 10.1088/1742-6596/898/8/082003.

[12] K. Biery, C. Green, J. Kowalkowski, M. Paterno, and R. Rechenmacher, "Artdaq: An event-building, filtering, and processing framework," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 5, pp. 3764–3771, Oct. 2013.

[13] J. Anderson *et al.*, "A new approach to front-end electronics interfacing in the ATLAS experiment," *J. Instrum.*, vol. 11, no. 1, Jan. 2016, Art. no. C01055. doi: 10.1088/1748-0221/11/01/C01055.

[14] Intel. (2018). *Intel QuickAssist Technology (Intel QAT)*. Accessed: Jan. 31, 2018. [Online]. Available: https://www.intel.com/content/www/us/en/architecture-and-technology/intel-quick-assist-technology-overview.html.

[15] P. Deutsch, *DEFLATE Compressed Data Format Specification Version 1.3*, document RFC 1951, May 1996. Accessed: Nov. 30, 2018. [Online]. Available: https://tools.ietf.org/html/rfc1951