

FELIX: the New Detector Interface for the ATLAS Experiment

W. Wu

on behalf of the ATLAS TDAQ Collaboration

Abstract—During the next major shutdown (2019-2020), the ATLAS experiment at the LHC will adopt the Front-End Link eXchange (FELIX) system as the interface between the data acquisition, detector control and TTC (Timing, Trigger and Control) systems and new or updated trigger and detector front-end electronics. FELIX will function as a router between custom serial links from front-end ASICs and FPGAs to data collection and processing components via a commodity switched network. Links may aggregate many slower links or be a single high bandwidth link. FELIX will also forward the LHC bunch-crossing clock, fixed latency trigger accepts and resets received from the TTC system to front-end electronics. The FELIX system uses commodity server technology in combination with FPGA-based PCIe I/O cards. The FELIX servers will run a software routing platform serving data to network clients. Commodity servers connected to FELIX systems via the same network will run the new Software Readout Driver (SW ROD) infrastructure for event fragment building and buffering, with support for detector or trigger specific data processing, and will serve the data upon request to the ATLAS High Level Trigger for Event Building and Selection. This paper will cover the design and status of FELIX, the SW ROD, results of early performance testing and integration tests with several ATLAS front-ends.

Index Terms—ATLAS experiment, ATLAS Level-1 calorimeter trigger system, ATLAS Muon Spectrometer, data acquisition.

I. INTRODUCTION

THE Large Hadron Collider (LHC) will undergo a series of significant upgrades in the next ten years, which increase both collision energy and peak luminosity. As one of the four major experiments, the ATLAS experiment will also follow the same upgrade steps [1]. The Front End Link eXchange (FELIX) is a new detector readout component being developed as part of the ATLAS upgrade effort [2]. FELIX is designed to act as a data router, receiving packets from detector front-end electronics and sending them to programmable peers on a commodity high bandwidth network. In the ATLAS Run 3 upgrade, FELIX will be used by the Liquid Argon (LAr) Calorimeters, Level-1 Calorimeter trigger system, BIS 7/8 and the New Small Wheel (NSW) muon detectors, as shown in the Fig. 1 [3], [4]. In the ATLAS Run 4 upgrade, the FELIX approach will be used to interface with all ATLAS detector and trigger systems.

FELIX brings multiple improvements in both performance and maintenance of the full DAQ (data acquisition) chain.

W. Wu is with Brookhaven National Laboratory, P.O. Box 5000, Upton, NY 11973-5000, USA. Email: weihawu@bnl.gov.

Copyright 2018 CERN for the benefit of the ATLAS Collaboration. Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.

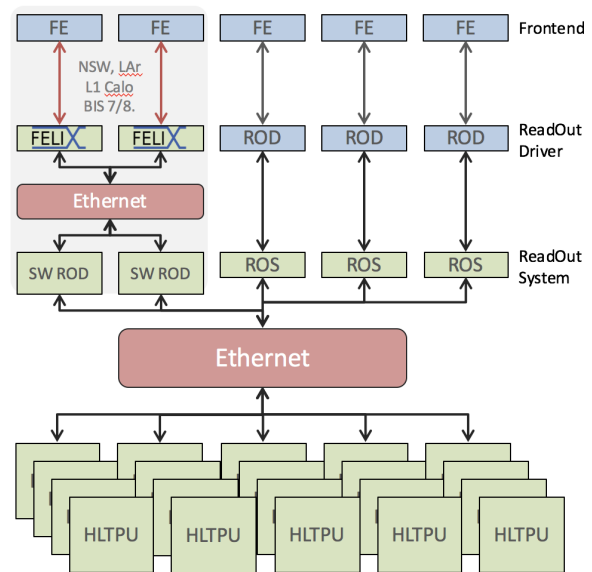


Fig. 1. ATLAS DAQ architecture in Run 3 upgrade

Since the FELIX system maximizes the use of commodity hardware, the DAQ system can reduce its reliance on custom hardware. Furthermore additional COTS (commercial off-the-shelf) components can be easily connected to resize the FELIX infrastructure as needed. The FELIX system implements a switched network architecture which makes the DAQ system easier to maintain and more scalable for future upgrades [5]. The FELIX architecture meets the following requirements:

- FELIX should be detector independent.
- FELIX must support the CERN standard GBT protocol with all its configuration options to connect to FE (Front-End) units having radiation hardness concerns [6].
- FELIX must distribute TTC (Timing, Trigger and Control) signals via fixed latency optical links.
- FELIX must route data from different GBTx E-links to configurable network end-points. E-links are low bandwidth (80 to 320 Mb/s) serial electrical links that are aggregated into a single high speed (4.8 Gb/s) GBT optical link.
- For the ATLAS Run 4 upgrade, FELIX should also support fast calibration operations for FE units, by implementing a mechanism to send control commands and distribute data packets simultaneously at high throughput, with a synchronisation mechanism that does not involve network traffic.

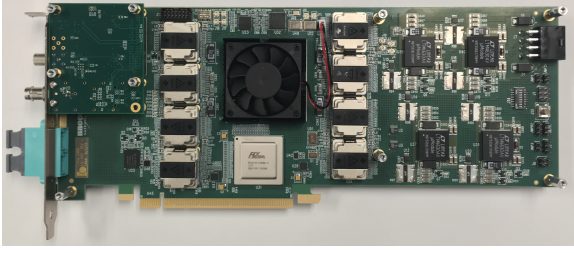


Fig. 2. FELIX final prototype FPGA PCIe card (FLX-712)

In this paper we introduce the FELIX hardware platform in Section II, the firmware design in Section III and software features in Section IV. The status of integration activities with several ATLAS front-end units is described in Section V.

II. THE FELIX INTERFACE CARD

The FELIX hardware platform has been developed for the final implementation in the ATLAS Run 3 upgrade. It is a standard height PCIe Gen3 card. The latest version is named as the FLX-712, as shown in Fig. 2. It is based on a Xilinx Kintex UltraScale FPGA (XCKU115-FLVF-1924) capable of supporting 48 bi-directional high-speed optical links via on-board MiniPOD transceivers, with a 16-lane PCIe Gen3 interface. In comparison to the previous version (FLX-711), the FLX-712 no longer hosts the unneeded DDR4 SODIMM connectors [7]. This eases PCB routing and also makes the board shorter. Since the FPGA has two Super Logic Regions (SLRs), two 8-lane PCIe endpoints are implemented in separate SLRs to achieve a balanced placement and routing that allows more channels to be serviced and easier timing closure.

Fig. 3 shows the functional block diagram of the FLX-712. Since the Xilinx UltraScale FPGA supports at most 8-lane PCI Express, a PCIe switch (PEX8732) is used to connect two 8-lane endpoints to the 16-lane PCIe slot. This approach ensures that it is possible to achieve the required nominal bandwidth of 128 Gb/s. There are four transmitter MiniPODs and four receiver MiniPODs on board; each one has 12 high-speed Rx or Tx links connected to FPGA GTH transceivers [8]. The speed of these 48 optical links can be up to 14 Gb/s, which is limited by the MiniPODs. An on-board jitter cleaner chip (Si5345) is used to provide a low jitter reference clock, at an integer multiple of the BC (bunch-crossing) clock, for the GTH transceivers. The Front-end optical links can connect to the FLX-712 via two optical multi-fiber (MTP) couplers. The MTPs can each be either MTP-24 (12 pairs) or MTP-48 (24 pairs) according to the application.

All of the hardware features of FLX-712 have been successfully verified. To test the PCIe interface, two Wupper DMA engines were implemented in the FPGA. Counter patterns were then used to test the throughput to the host server. The total measured throughput of these two 8-lane PCIe Gen3 endpoints can be up to 101.7 Gb/s, in agreement with the PCIe specification. To test the optical links, the Xilinx IBERT IP was used to perform BER (Bit Error Rate) and eye diagram tests at line rates of 12.8 Gb/s and 9.6 Gb/s [8] [9]. The results show that the BER is smaller than 10^{-15} for all of the 48

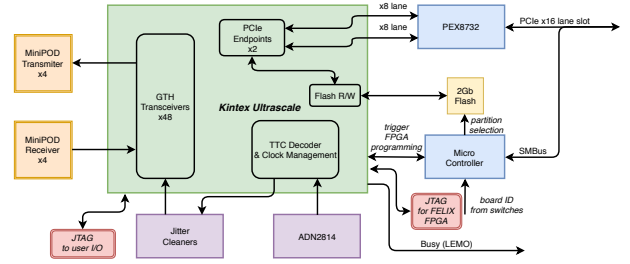


Fig. 3. Block diagram of the FELIX final prototype FPGA card

optical links. A typical eye diagram at 12.8 Gb/s is shown in Fig. 4.

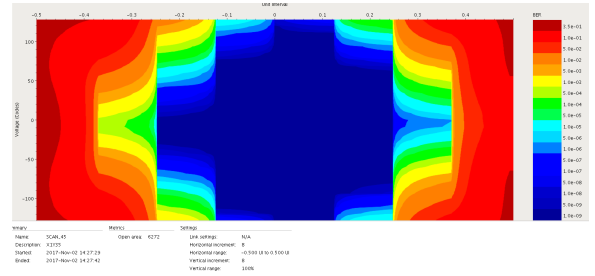


Fig. 4. The eye diagram for one optical transceiver channel at 12.8 Gb/s

III. FELIX FIRMWARE

The FELIX firmware supports two modes: GBT mode and FULL mode. GBT mode uses GigaBit Transceiver (GBT) architecture and a protocol developed by CERN providing a bi-directional high-speed (4.8 Gb/s) radiation-hard optical link [6]. FULL mode uses a customized light-weight protocol for the front-end path, providing a higher maximum payload at a line rate of 9.6 Gb/s. As FULL mode uses 8b/10b encoding, a maximum user payload of 7.68 Gb/s can be achieved. The main functional blocks of the FELIX firmware, shown in Fig. 5, consist of a GBT wrapper, Central Router, PCIe Direct Memory Access (DMA) engine and other modules. Two sets of firmware modules are instantiated in the top level design to have a balanced structure and to ease FPGA net routing.

In addition to routing front-end data streams, FELIX also distributes TTC information to front-end electronics from the TTC system. The TTC decoder firmware module is based on the TTC firmware from the CERN GLIB project [10]. It receives the clock and serial TTC data from a TTC optical fiber via a clock and data recovery chip (ADN2814). The serial TTC data contains two interleaved data streams: the A-channel, reserved for the Level-1 Accept, and the B-channel which carries other commands such as BCR (Bunch Counter Reset). The generated 40.08 MHz TTC clock from the MMCM is distributed via a dedicated clock net to the rest of FPGA fabric. Due to the low jitter requirement of the high-speed GTH transceivers, their reference clock is provided by the on-board jitter cleaner (Si5345) which multiplies the frequency and cleans the jitter.

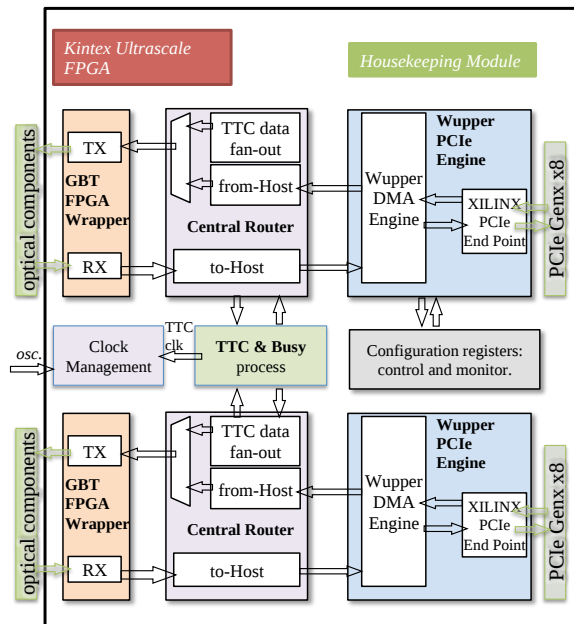


Fig. 5. Block diagram of the FLX-712 firmware

The FELIX GBT wrapper is based on the CERN GBT-FPGA firmware with several performance improvements [11]. It encapsulates the Forward Error Correction (FEC) encoder/decoder, a scrambler/descrambler and a gearbox architecture. To decrease the latency, the frequency of the FEC encoder/decoder and scrambler/descrambler clock domain was increased to 240 MHz [12]. The GBT protocol supports GBT frame-encoding mode and wide-bus mode [11]. The wide-bus mode is not radiation tolerant, as the FEC encoder and decoder are sacrificed in the to-host direction in favor of a higher user payload.

PCIe firmware, called Wupper, was designed to provide a simple Direct Memory Access (DMA) interface for the Xilinx PCIe Gen3 hard block [13]. It transfers data between a 256-bit wide user logic FIFO and the host server memory, according to the addresses specified in DMA descriptors. Up to eight descriptors can be queued to be processed sequentially. Since the Xilinx PCIe Gen3 hard block only supports a maximum of eight lanes, the FPGA implements two 8-lane PCIe endpoints with separate DMA engines. The block diagram of the Wupper design is shown in Fig. 6. Its functional blocks can be categorized into two groups: DMA control and DMA write/read. The DMA control parses and monitors received descriptors. The DMA write/read blocks process the data streams for both directions. If the received descriptor is a to-host descriptor, the payload data is read from the user logic FIFO and added after the header information. If the descriptor is a from-host descriptor, the header of received data is removed and the length is checked; then the payload is shifted into the FIFO.

IV. FELIX SOFTWARE

The FELIX software suite has different layers: for example, low-level software tools, test software and production software. Access to the FELIX hardware level is controlled

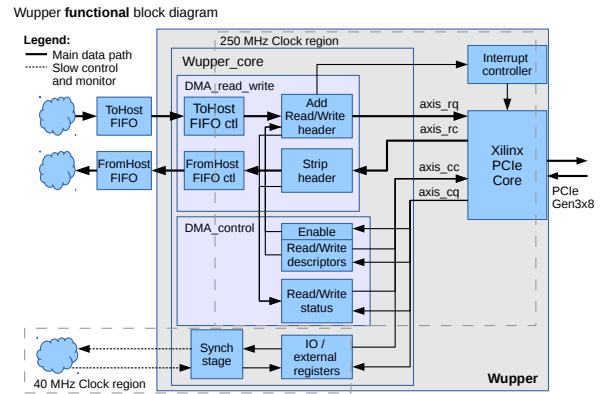


Fig. 6. Structure of the Wupper PCIe engine

via two device drivers: flx and cmem_rcc. The flx driver is a conventional character driver for PCIe interface cards. Its main function is to provide virtual addresses for the registers of a FLX-712 card that can be used directly by user processes for access to the hardware. This design avoids the overhead of a context switch per IO transaction and is therefore essential for the performance of FELIX. The cmem_rcc driver, from the ATLAS TDAQ project, allows the application software to allocate large buffers of contiguous memory. For use with FELIX, it has been tested for buffers of up to 16 GByte and the allocation time of large buffers has been reduced.

The felixcore application handles the data between the front-ends using the FLX-712 card and a dedicated library called NetIO. Its functional architecture is shown in Fig. 7. It does not perform any content analysis or manipulation of the data, other than that which is needed for decoding and transport. The DMA engine transfers a data stream into a contiguous circular buffer which is allocated using the cmem_rcc driver in the memory of the host server. Continuous DMA enables data transfer at full speed and does not require the DMA to be re-set for each transfer. Data blocks retrieved from the circular buffer are inspected for integrity while extracting the E-link identifier and sequence number. The block is then copied to a selected worker thread based on the E-link identifier. The worker threads recombine the data stream for each E-link if any splitting for transport proved necessary. Once the data reconstruction is complete, the data are appended with a FELIX header and published to the network through NetIO.

NetIO is implemented as a generic message-based networking library that is tuned for typical use cases in DAQ systems. It offers four different communication modes: low-latency point-to-point communication, high-throughput point-to-point communication, low-latency publish/subscribe communication and high-throughput publish/subscribe communication. NetIO has a backend system to support different network technologies and API's. At this time, two different backends exist. The first backend uses POSIX sockets to establish reliable connections to endpoints. Typically this backend is used for TCP/IP connections in Ethernet networks. The second backend uses libfabric for communication and is used for Infiniband

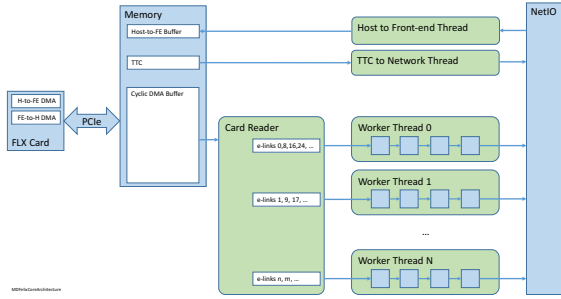


Fig. 7. The felixcore application architecture

and similar network technologies [14]. Libfabric is a network API that is provided by the OpenFabrics Working Group. There are six different user-level sockets in NetIO, of which four are point-to-point sockets (one send socket and one receive socket, each in a high-throughput and a low-latency version), and two publish/subscribe sockets (one publish and one subscribe socket). The publish/subscribe sockets internally use the point-to-point sockets for data communication.

A number of benchmarks have been carried out to evaluate the performance of felixcore application and NetIO. These tests were run with a host server as the FELIX and another host as the data receiver. A 40 GbE connection was available between the hosts. In the GBT mode performance test, two FLX-712 cards were used to support 48 GBT links. The FLX cards were configured to the most demanding workload for the ATLAS Run 3 upgrade, with 8 E-links per GBT link and a chunk size of 40 Bytes. As shown in Fig. 8, the system is comfortably able to transfer the full load at above the ATLAS L1 Accept rate of 100 kHz. Benchmarking for the FULL mode case also indicates that it will be possible to handle data at the L1 Accept rate.

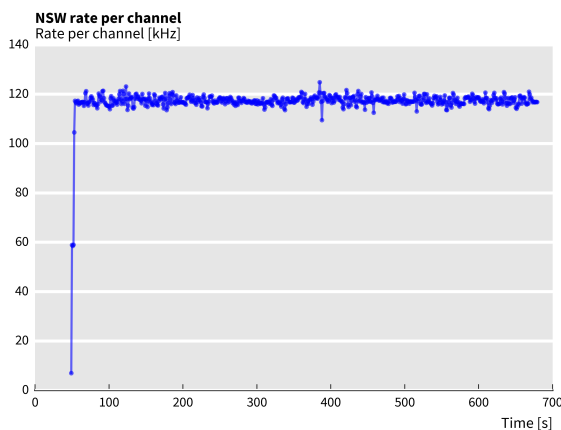


Fig. 8. Felixcore performance for GBT mode

A Software ROD (ReadOut Driver) is an application running on a commodity server which receives data from one or more FELIX systems and performs flexible data aggregation and formatting tasks. Incoming data packets associated with a given ATLAS event are automatically logically aggregated into a larger event fragment for further processing. The data are

finally formatted to match common ATLAS specification, as produced by existing readout system, for consumption by High Level Trigger (HLT) on request. Benchmarks for the current aggregation algorithms, including realistic simulation of the cost of subdetector processing and HLT request handling, were carried out with simulated input data from multiple FELIX cards, each with 192 E-links and realistic packet sizes. The test results are shown in the Fig. 9. The algorithm is shown to be able to handle input from multiple FELIX cards, with the performance able dependent on host CPU speed and number of cores. The 1%, 50% and 100% in the plot refer to the fraction of the events arriving at the software ROD which the HLT then samples.

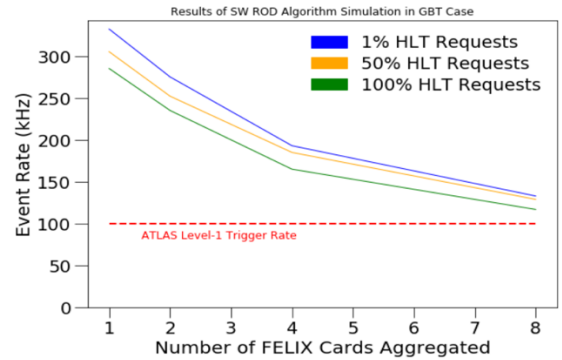


Fig. 9. Software ROD performance

V. INTEGRATION TESTS WITH DIFFERENT FRONT-ENDS

For the upcoming ATLAS Run 3 upgrade in 2019, FELIX will be implemented to interface with several detector front-ends, such as the Muon Spectrometer's New Small Wheel (NSW), Liquid Argon Calorimeter (LAR) Trigger Digitizer Board (LDPB) and the Level-1 Calorimeter Trigger (L1Calo) system [3] [15]. For the Run 4 upgrade of HL-LHC (High-Luminosity LHC), the plan is to adopt FELIX to interface with all the detector front-ends.

A. Integration Test with New Small Wheel Front-ends

In the NSW integration tests, FELIX successfully distributed TTC information to front-end electronics, including the bunch crossing clock and L1A trigger signal. The dataflow to and from the front-ends has been demonstrated. FELIX can also trigger a front-end test pulse from a test application, and successfully configure ASICs and FPGAs via the GBT-SCA's GPIO, I²C, SPI and JTAG interfaces [16]. Other highlights also include the ability to read out ADC monitoring data and configure the GBTx on the L1DDC board [17]. Taken together these tests provide a robust demonstration of the functionality of the IC and SCA links in the GBT frame [6].

B. Integration Test with Liquid Argon Calorimeter LTDB

In the LAR (Liquid Argon Calorimeter) Run 3 upgrade, the LAR Trigger Digitizer Board (LTDB) digitizes input analog signals, and transmits them to the back-end system [4]. There

are five GBTx and five GBT-SCA chips on the LTDB prototype. And five GBT links in total from FELIX are connected to the LTDB. Part of the connection scheme (one GBT link) is shown in Fig. 10. GBT-SCA chips are used to control the power rails, I²C buses and also perform on-board temperature measurement [16]. Besides the interface to EC links with the GBT-SCA chip, each GBTx on the LTDB provides the recovered 40 MHz TTC clock from a FELIX GBT link to the ASICs of the NEVIS ADC and serializers LOCx2, and also sends the BCR (Bunch Counter Reset) signal to the LOCx2 ASIC [18] [19].

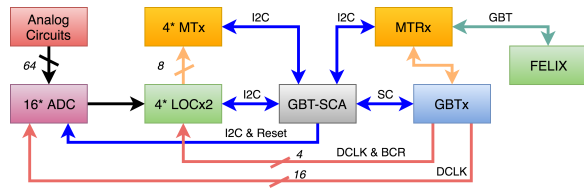


Fig. 10. GBTx and GBT-SCA connections in the LTDB prototype

C. Integration Test with gFEX

The Global Feature Extractor (gFEX) is one of several modules that will be deployed in the Level-1 Calorimeter (L1Calo) trigger system in the ATLAS Run 3 upgrade [20]. In the integration test of gFEX and FELIX, gFEX needs to recover the TTC clock from a FELIX GBT link at 4.8 Gb/s, and also receive TTC signals such as Level-1 trigger accept and BCR. As for the to-host path, gFEX needs to send data to FELIX using FULL mode optical links at 9.6 Gb/s. A block diagram of the test setup is shown in Fig. 11. The test results show that gFEX recovers a stable TTC clock and receives the TTC information correctly. The latency of TTC signal transmission (from TTC system to gFEX through FELIX) is fixed and does not change under conditions such as transceiver reset, fiber reconnection, TTC system power cycling and FELIX & gFEX power cycling. The FULL mode links from gFEX to FELIX have been tested with the PRBS-31 (Pseudo Random Bit Sequence) data pattern. No error was observed and the BER (Bit Error Rate) is smaller than 10^{-15} .

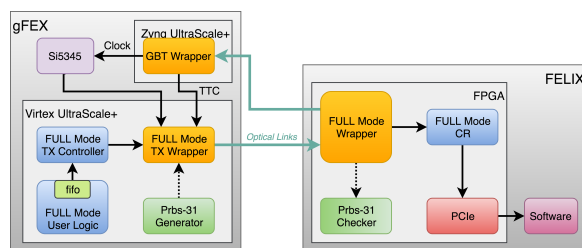


Fig. 11. Block diagram of gFEX and FELIX integration test

VI. CONCLUSION

FELIX is a readout system that interfaces custom links from front-end electronics to standard commercial networks in the ATLAS upgrade. FELIX also distributes the LHC bunch-crossing clock, trigger accepts and resets received from

the TTC system to detector front-ends through fixed latency optical links. It supports the CERN standard 4.8 Gb/s GBT protocol and a customized lightweight FULL mode which has a higher throughput of 9.6 Gb/s. The results of integration and performance tests with ATLAS front-end systems to date indicate that FELIX is on course to be ready for deployment in 2019.

REFERENCES

- [1] ATLAS Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider”, 2008 JINST **3** S08003
- [2] J. Anderson *et al.*, “A new approach to frontend electronics interfacing in the ATLAS experiment”, 2016 JINST **11** C01055
- [3] R.-M. Coliban *et al.*, “The Read Out Controller for the ATLAS New Small Wheel”, 2016 JINST **11** C02069
- [4] Aleksa, M. *et al.*, “ATLAS Liquid Argon Calorimeter Phase-I Upgrade Technical Design Report”, CERN-LHCC-2013-017, ATLAS-TDR-022
- [5] J. Anderson *et al.*, “FELIX: a High-Throughput Network Approach for Interfacing to Front End Electronics for ATLAS Upgrades”, 2015 J. Phys.: Conf. Ser. **664** 082050
- [6] CERN GBT Project: GBTX Manual
- [7] J. Anderson *et al.*, “FELIX: a PCIe based high-throughput approach for interfacing front-end and trigger electronics in the ATLAS Upgrade framework”, 2016 JINST **11** C12023
- [8] Xilinx, UltraScale Architecture GTH Transceivers, http://www.xilinx.com/support/documentation/user_guides/ug576-ultrascale-gth-transceivers.pdf
- [9] Xilinx, IBERT for UltraScale GTH Transceivers v1.3, http://www.xilinx.com/support/documentation/ip_documentation/ibert_ultrascale_gth/v1_3/pg173-ibert-ultrascale-gth.pdf
- [10] P Vichoudis *et al.*, “The Gigabit Link Interface Board (GLIB), a flexible system for the evaluation and use of GBT-based optical links”, 2010 JINST **5** C11007
- [11] M. Barros Marin *et al.*, “The GBT-FPGA core: features and challenges”, 2015 JINST **10** C03021
- [12] K. Chen *et al.*, “Optimization on fixed low latency implementation of the GBT core in FPGA”, 2017 JINST **12** P07011
- [13] A. Borga *et al.*, PCIe Gen3x8 DMA for virtex7, <http://stackoverflow.com/>
- [14] Libfabric, OpenFabrics, <https://ofwg.github.io/libfabric>
- [15] H. Chen *et al.*, “The prototype design of gFEX - a component of the l1calo trigger for the ATLAS phase-I upgrade”, 2016 *IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD)*, Strasbourg, 2016, pp. 1-5
- [16] A. Caratelli *et al.*, “The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments”, 2015 JINST **10** C03034
- [17] P. Gkoutoumis “Level-1 data driver card of the ATLAS new small wheel upgrade compatible with the phase II 1 MHz readout scheme”, 2016 *5th International Conference on Modern Circuits and Systems Technologies (MOCAS)*, Thessaloniki, 2016, pp. 1-4
- [18] J Kuppambatti *et al.*, “A radiation-hard dual channel 4-bit pipeline for a 12-bit 40 MS/s ADC prototype with extended dynamic range for the ATLAS Liquid Argon Calorimeter readout electronics upgrade at the CERN LHC”, 2013 JINST **8** P09008
- [19] L. Xiao *et al.*, “LOCx2, a low-latency, low-overhead, 2x 5.12-Gbps transmitter ASIC for the ATLAS Liquid Argon Calorimeter trigger upgrade”, 2016 JINST **11** C02013
- [20] S. Tang *et al.*, “gFEX, the ATLAS Calorimeter Level-1 real time processor”, 2015 *IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, San Diego, CA, 2015, pp. 1-5