FELIX: The New Readout System for the ATLAS Detector

Marco Trovato
on behalf of ATLAS TDAQ Collaboration

***Abstract*—After the current LHC shutdown (2019-2021), the ATLAS experiment at the LHC at CERN will be required to operate in an increasingly harsh collision environment. The LHC will deliver luminosities up to three times the original design value, with a commensurate increase in the number of interactions per bunch crossing. To maintain physics performance in this new regime, the ATLAS experiment will undergo a series of upgrades during the shutdown. A key goal of this upgrade is to improve the capacity and flexibility of the detector readout system. To this end, the Front-End Link eXchange (FELIX) system has been developed. FELIX acts 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 functions as a router between custom and radiation tolerant 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 also distributes the LHC bunch-crossing clock, as well as the fixed-latency trigger accepts and resets received from the TTC system, to the front-end electronics. FELIX uses commodity server technology in combination with FPGA-based PCIe I/O cards. FELIX servers run a software routing platform serving data to network clients. Commodity servers connected to FELIX systems via the same network run the new multi-threaded Software Readout Driver (SW ROD) infrastructure for event fragment building and buffering. In addition the SW ROD supports detector specific data processing, and serves the data, upon request, to the ATLAS High Level Trigger for Event Building and Selection. This contribution introduces the FELIX system, describes the firmware and software under development, and shows the integration test results prior to the commissioning of next year.**

***Index Terms*—ATLAS experiment, Data Acquisition, ATLAS Upgrade.**

ATL-DAQ-PROC-2019-036
13 December 2019

I. Introduction

THE Large Hadron Collider (LHC) accelerates and collides protons at 13 TeV center-of-mass energy. The ATLAS experiment [1] is one of the four major experiments built to detect the product of those collisions. A trigger and data acquisition system (TDAQ) is of utmost importance for selecting in real time only those collisions (also called events) worth investigating (about 1 out of 10^7).

The LHC is currently undergoing a major upgrade, named Phase-I upgrade, which is expected to increase both collision energy and peak luminosity during the Run 3 data taking period during 2021-2023. Another upgrade, named Phase-II upgrade, will follow during 2024-2026, before the Run 4 and Run 5 data taking periods (2026-2038). To cope with the change in luminosity, the ATLAS experiment will have to be upgraded. The Front End LInk eXchange (FELIX) is a new detector readout component. FELIX, being developed as part of the ATLAS TDAQ upgrade effort, acts as data router: it manages packets between the detector front-end electronics (FE) and commercial high bandwidth networks (Fig. 1). Communication with FEs happens via the GigaBit Transceiver (GBT) and Versatile Link architecture [2]. Communication with commercial networks happens via Remote Direct Memory Access (RDMA).

The GBT protocol, developed at CERN, provides a high-speed (4.8 Gb/s) radiation-hard optical link for data transmission [2]. In the default mode, where a forward error correction code is encoded, the GBT protocol features up to 40 different logical streams, also called e-links, thus allowing multiplexing data from several FEs into a single fiber. Detector control system (DCS) data and event data packets can be intertwined in a single e-link. The GBT frame consists of 4 bits for slow control and monitoring, 4 bits for the header, 80 bits for the payload, and 32 bits for forward error corrections. The header, which is either data (0101) or idles (0110) is used to achieve alignment at the receiver side.

The FELIX system is also in charge of distributing the input from the Timing, Trigger and Control (TTC) system [3]. The LHC clock and trigger information, after being recovered, are distributed to both on-detector electronics with low and fixed latency via GBT links, and to network endpoints.

In Run 3 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 [4], [5], [6]. Starting from Run 4 FELIX will readout the entire ATLAS detector [7].

Besides optimizing performance, FELIX reduces the reliance on custom hardware. By maximizing the use of commodity hardware, FELIX is easy to maintain and to upgrade. FELIX is also very modular, thus allowing the system to be scalable (e.g., different commercial off-the-shelf components can be easily swapped to resize the FELIX infrastructure). The FELIX system implements a switched network architecture which, again, makes the DAQ system easier to maintain and more scalable for future upgrades. FELIX is also detector independent.

In this contribution we introduce the FELIX hardware platform in Section II, the firmware design in Section III, and the software package in Section IV. Integration and reliability

Manuscript received MONTH DAY, 2019. (TO DO: Write the date on which you submitted your paper for review.)

Marco Trovato is with the Argonne National Laboratory, 9700 S Cass Ave, Lemont, IL 60439, USA (e-mail: mtrovato@anl.gov).

978-1-7281-4164-0/19/$31.00 ©2019 IEEE

tests are described in Section V, right before the conclusions in Section VI.

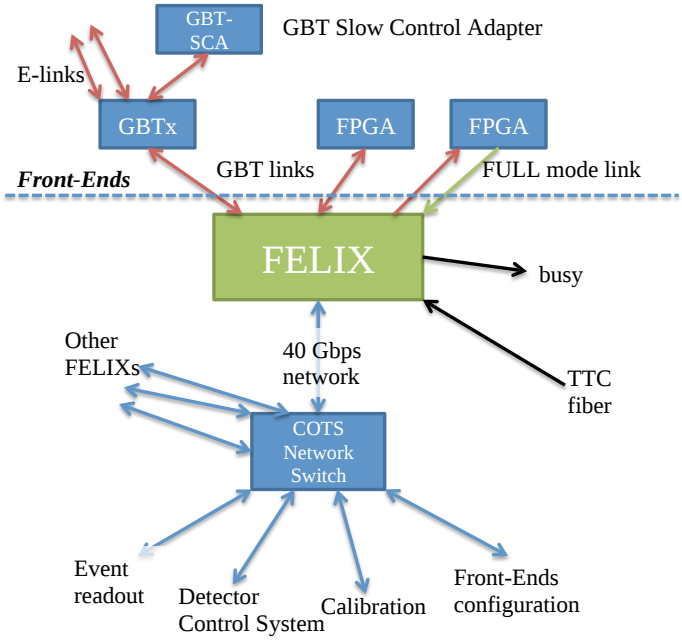

Fig. 1. The FELIX system acting as router between front-end electronics and commercial networks. FELIX is part of the ATLAS trigger and DAQ system.

II. Hardware

FELIX is a commercial server plus up to two PCIe cards. The final prototype of the PCIe card, named FLX-712 (Fig. 2), feature 16 Gen-3 PCIe lanes, 48 TX and RX optical links. The on-board FPGA is a Kintex Ultrascale XCKU115FLVF1924-2E [8]. The card hosts a custom form factor Timing Mezzanine Card (TMC), which can be assembled in three different configurations to allow interfacing with the TTC, 28 TTC-PON or the White Rabbit [3] systems. In the case of ATLAS Phase-I the TTC configuration is used. A custom made stiffener bar has been produced to strengthen the card. The dimensions of the card are 270 mm $\times$ 110 mm $\times$ 1.58 mm, according to the PCIe Card Electromechanical Specifications.

As shown in Figure 3, FLX-12 hosts 4 MiniPOD transmitters and 4 MiniPOD receivers. Each MiniPOD has 12 high-speed links connected with FPGA GTH transceivers [9]. The speed of the 48 optical links can be up to 14 Gb/s, limited by the MiniPODs. The TTC clock from the data recovery chip (ADN2814) goes through either a Si5345 or LMK03200 jitter cleaner. The cleaned 240 MHz clock is used as a reference clock for the GTH transceivers. Two of the hardcore PCIe end-points within the FPGA are used, with the PEX8732 PCIe switch used to connect them to a 16-lane slot. This approach ensures the possibility to achieve the required nominal bandwidth of 128 Gb/s. The FE optical links can connect to the FLX-712 via two optical multi-fiber (MTP) couplers. The MTPs are either MTP-24 (12 pairs) or MTP-48 (24 pairs), depending on the application.

An on-board 2 Gb FLASH memory can store 4 different firmware bit files. An on-board microcontroller, which is accessible by the host via either SMBus or FPGA and PCIe interface, can be used to select one of the four FLASH memory partitions, and trigger FPGA programming from the image stored in the selected partition. On power-up the FPGA is programmed with the image from the partition that is selected by two on-board jumpers. A direct JTAG connection for FPGA programming is also available. The FLASH memory can be loaded with bit files either via the PCIe interface or JTAG.

A single +12V power line is supplied to the card via an on-board 8-pin connector (Fig. 2). The FLX-712 uses 4 dual 18A regulators (LTM4630A). In order to meet the power sequence requirements of Xilinx FPGAs they are enabled in 3 stages, as per the AC and DC characteristics indicated in the Xilinx documentation [8].

Link speed testing has been performed on 17 FLX-712 prototypes of the latest versions. The average bit error rate is lower than 1×10^{-15}.

A Xilinx VC-709 evaluation board [10] is also used for debugging the firmware and performing small-scale tests.

The current recommended hardware platform for the FELIX system is based on the Supermicro® X10SRA-F motherboard [11]. The system is populated with at least 32 GB of DDR4 RAM and an Intel® Broadwell ™ E5 family CPU (v4) with at least six real cores. Current Network Interface card used for testing is a Mellanox ConnectX-5 operated in RDMA mode at either 2 $\times$ 25 Gb/s or 2 $\times$ 100 Gb/s.

FLX-712 or HiTech Global HTG-710 [12] evaluation board are used for emulating the FE data, which is needed to test the FELIX system prior to the commissioning.

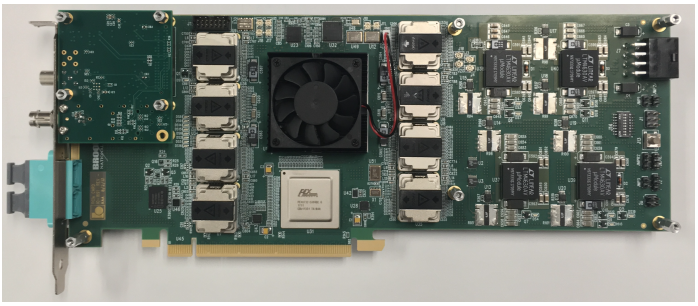

Fig. 2. The final FELIX prototype for Phase-I (FLX-712) with fibres assembled.

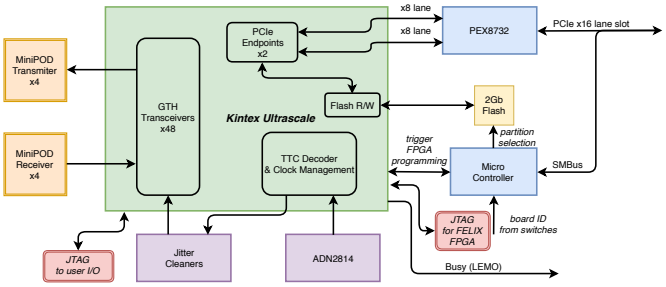

Fig. 3. Diagram of FLX-712 final prototype.

III. Firmware

FELIX firmware supports two modes: GBT mode and FULL mode. GBT mode uses the GigaBit Transceiver architecture and the protocol described in Section I. The high-speed communication at 4.8 Gb/s is bi-directional. While FULL mode is equivalent to the GBT for the path towards the FE, it uses a customized light-weight protocol for the path from the FE: that bandwidth is 9.6 Gb/s, double with respect to the GBT mode. In order to ensure DC-balancing, data is scrambled in GBT mode or 8b/10b encoded in FULL mode. The maximum payload bandwidths for the GBT and FULL modes are 3.2 Gb/s and 7.68 Gb/s respectively. If GBT wide-mode is used (e.g., no forward error correction) the maximum payload bandwidth for GBT is 4.48 Gb/s. The GBT wide-mode is non-radiation tolerant.

The main modules of the firmware are the GBT wrapper, which is an optimized version of GBT-FPGA core [13], the Central Router for internal management of e-links into physical links, the TTC decoder and the Wupper [15], which is the PCIe engine with DMA interface. A block diagram of the firmware for FLX-712 is shown in Fig. 4. Given the two PCIe end-points, two distinct instances of Wupper PCIe engines are instantiated in the design: half of the channels are connected to each engine. This approach introduces the concept of logical devices co-existing in one single physical device. Central router and GBT wrapper modules are also duplicated to ease the FPGA net routing.

As described earlier, the FELIX GBT wrapper encapsulates the forward error correction encoding and decoding, and scrambler architecture from the CERN GBT-FPGA project. To decrease the latency, firmware blocks operate in a 240 MHz domain. Logic to ensure a smooth time domain crossing between the 40 MHz TTC clock and the 240 MHz clock has been placed. The scrambler and descrambler are enabled once per six 240 MHz clocks. Two multiplexers are added to improve modularity: one for GBT encoding, another for GBT decoding. With these multiplexers, GBT modes for TX and RX can be changed independently. A tailored RX finite state machine takes care of the GBT alignment automatically. The design details are described in [14].

The Central Router routes and formats e-link data between the GBT interface and the PCIe DMA engine in both directions: to-Host direction, from the GBT interface to PCIe engine, and from-Host direction, from the PCIe engine to the GBT interface. Each central router instance handles 12 physical links. The data is organized in e-groups of 16 bits each. Each e-group is composed of up to 8 e-links. The e-groups are synchronous with the TTC clock: data is transferred every 40 MHz clock. There are four possible e-link data widths: 2, 4, 8 and 16 bits, corresponding to data rates of 80, 160, 320 and 640 Mb/s. The e-links can be dynamically activated and their widths can be dynamically changed. Different data formats are supported (e.g., 8b/10b). The two directions are completely independent.

Wupper is designed to provide a simple Direct Memory Access (DMA) interface for the Xilinx Virtex-7/Kintex Ultrascale PCIe Gen-3 hard block. In the FPGA Wupper provides an interface to a standard FIFO. This FIFO has the same width as the Xilinx AXI4-Stream interface (256 bits) and runs at 250 MHz. The Central Router reads from or writes to the FIFO. Wupper handles the transfer into server memory, according to the addresses specified in the DMA descriptors. These descriptors, which can be queued up to a maximum of 8, will be processed sequentially. The DMA descriptors, with an address, a read/write flag, the transfer size (number of 32-bit words) and an enable line, are mapped as normal PCIe memory or IO registers. Besides the descriptors and the enable line, a status register for every descriptor is provided in the register map. The Wupper core is divided in two parts: DMA control and DMA read/write. The DMA control parses and monitors received descriptors. The DMA write/read blocks process the data streams for both directions. The achievable throughput of a single Wupper instance is 64 Gb/s, as per the specifics of an 8-lane Gen-3 PCIe interface. Given the two Wupper instances, or the 16-lane PCIe interface of the FLX-712, a maximum throughput of 128 Gb/s can be achieved. See the Wupper documentation [15] for more details.

FELIX also distributes TTC information to FEs and network end-points from the TTC system. The TTC decoder firmware module is based on the TTC firmware from the CERN GLIB project [16]. It receives the clock and the serial TTC data stream from a TTC optical fiber via the ADN2814 chip. The serial data stream, encoded by a 160.32 MBaud biphase mark code, contains two interleaved data streams: the A channel, reserved for the Level-1 Accept, and the B channel, which carries the synchronous commands, such as bunch crossing reset, and asynchronous commands. The correct alignment of the LHC bunch crossing clock must be deduced from the A and B-channel streams. A state machine samples the serial data stream with a 160.32 MHz clock extracted by the ADN2814. It separates the A channel and B channel information, decodes the B-channel extracting the broadcast commands and the data, and provides a 40.08 MHz clock aligned to the LHC bunch crossing clock. This clock is used to choose the correct phase of a 40 MHz clock generated by an MMCM from the 160 MHz recovered clock. The generated clock is used for distribution, as well as to the rest to the rest of the FPGA fabric.

More details about the firmware (e.g., BUSY, XOFF) can be found in a separate document [17].

IV. Software

Access to the FELIX hardware level is controlled by a device driver named flx. The flx driver is a conventional character driver for PCIe cards. Its main function is to provide virtual addresses for the registers that can be used directly by user processes to access the FLX-712 card. This design avoids the overhead of a context switch per IO transaction, therefore being essential for the high FELIX performances. An additional driver, cmem_rcc, from the ATLAS TDAQ project, allows the software to allocate large buffers of contiguous memory. Access to the cmem_rcc driver happens via the cmem_rcc library. 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. The

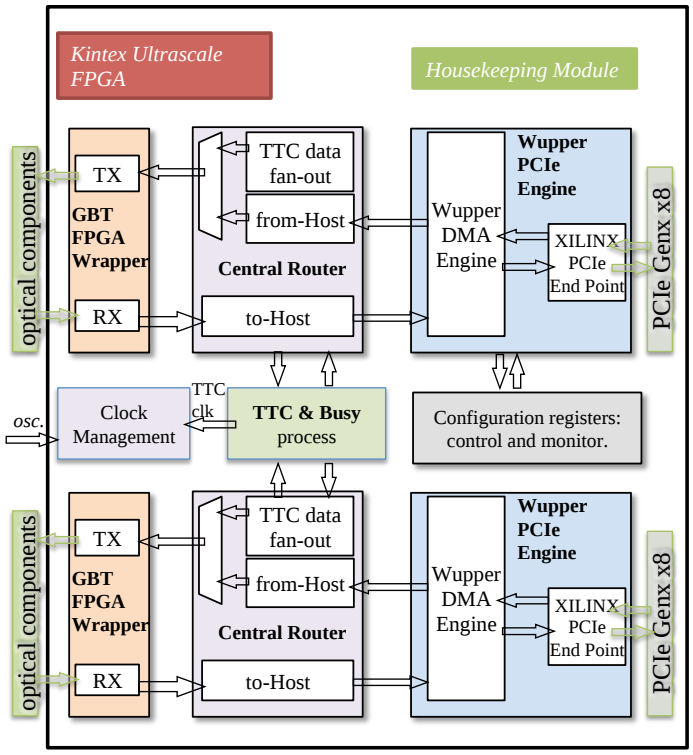

Fig. 4. *FELIX firmware diagram.*

RDMA technology is being utilized to maximize efficiency and throughput.

The FELIX software suite comprises both high and low level tools. At a lower level, the suite provides a number of tools, both command line and GUI based, to facilitate system configuration and testing. At the highest level, the FelixCore application is responsible for communication and bulk dataflow in a full slice system.

All communications with the flx driver and the registers of the FLX cards is via a low-level, object oriented library, called libFlxCard.so. The library provides an interface to the functions of the firmware, including support for DMA operations from and to the card, interrupts from the card, communication with on-board peripherals via I^2C, GBT configuration, interrupt handling, as well as card register and bit-field access. A number of specialized applications in the flxcard package allows to configure a FLX card, monitor its status, test DMA transactions or communicate with I^2C. While the software tools in the flxcard package are mostly for the low-level configuration, monitoring and testing of components in the hardware of the various cards available, the command-line software tools in the ftools package are more geared towards testing and debugging the interface of the FLX-card to the GBT links, as well as controlling and testing the functionality of the central router part of the FLX-card design. In order to allow configuration of e-links FLX-card provides, a set of registers to configure the central router component of the FLX-card accordingly. Such a configuration is matched to the configuration of the front-end GBT links. To provide a visual representation of the central router configuration and simplify any modification to it, a user-friendly graphical tool was developed, called elinkconfig.

The FELIX Core application is the central process of a FELIX system. It handles the communication between one or more FELIX cards on one side and a set of NetIO-library clients [18] on the other side (Fig. 5). FELIX Core is a stateless system, to be run all the time, for DAQ and DCS purposes. It runs on a PC, running the Linux operating system, in which the FELIX card(s) and network Card(s) are installed. FELIX Core has the following functions:

- Packet forwarding from the FEs to the DAQ system: read and decode data packets from the FELIX card and forward them as messages to NetIO clients, based on dynamic routing rules;
- packet forwarding from the DAQ system to the FEs: receive messages from NetIO clients and forward them as packets to the e-links of the FELIX Card, as supplied in the message header;
- configure the FELIX card, e-link configuration and operational parameters based on the input of a configuration file;
- recover from host failures. Using a publish/subscribe system host failures of FELIX clients can be handled transparently;
- advertise e-link meta information to NetIO clients before they actually connect;
- gather statistics and performance metrics and make those available;
- report operational status information, such as the status of the detector links, and warn in the case of hardware failure;
- offset e-links to distinguish when multiple FELIX Core applications are run to read out multiple cards;
- handle the FELIX card, which internally is seen as two cards, by a single FELIX Core instance;
- handle multiple FELIX Cards by the same FELIX Core instance.

NetIO is implemented as a generic message-based networking library that is tuned for typical use cases in DAQ systems. It offers 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 APIs: POSIX or Infiniband.

More details about the FELIX software can be found in [19].

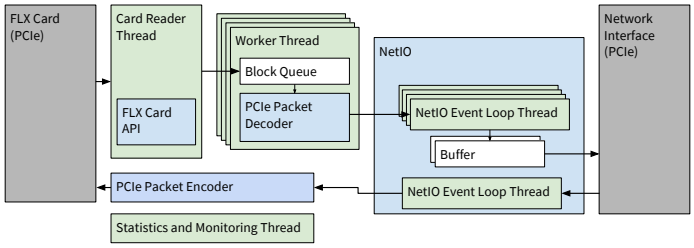

Fig. 5. *FelixCore Architecture.*

V. Integration and Performances

As mentioned in Section I, during Run 3 FELIX will serve as a readout system for a number of ATLAS sub-detectors. Starting from Run 4 FELIX will read out the whole ATLAS detector. Integration tests of the FELIX system with the New Small Wheel FEs, Liquid Argon Digitizer Boards, and Global Feature Extractor module of the level 1 calorimeter, have been described elsewhere [20]. The FELIX system is also being integrated with the Tile Calorimeter and Inner Tracker for Phase-II. In this document we focus on the latest tests aiming to characterize the performance of the system.

Full chain tests have been performed in both GBT and FULL modes. In both cases the following devices have been used: emulators, the FELIX system, a commodity server running the application named "software ROD" (SWROD), and a TTC system, producing TTC signals upon stimulation by a Rasperry Pi. A SWROD is an application running on commodity servers, which receive data from multiple FELIX systems, perform flexible data aggregation, and format tasks for further processing.

The GBT mode test was performed in the worst case scenario envisioned for Run 3: 384 e-links activated, streams of 40 bytes emulated and sent to the FELIX system at an average trigger rate of 100 kHz, which is the maximum trigger rate expected during Run 3 data taking. The total processed throughput is observed to be 12.5 Gb/s and is in line with expectations (Fig. 6). The trigger rate was also successfully tested in more elevated conditions (i.e., trigger rate of 150 kHz), thus providing a 50% safety margin (Fig. 7).

The FULL mode test was performed with 12 links, streams of 5 kB at an average trigger rate of 120 kHz. These conditions already exceed the requirement during Run 3. A total throughput of 56.7 Gb/s is observed, which agrees with expectations. As for the GBT tests, the trigger rate was also increased to show the robustness and reliability of the system. At about 200 kHz of average trigger rate (i.e., two times more than required), the FELIX XOFF signal is asserted to throttle the transmission of data from the FE to FELIX and prevent buffer overflows.

VI. Conclusions

FELIX is a router between front-end serial links and commodity networks: it separates data from data processing. FELIX also distributes the LHC bunch-crossing clock and trigger accept to the FEs. FELIX takes advantage of the latest progress in technology to simplify the ATLAS readout. After the Phase-I upgrade FELIX will be employed to read out a number of ATLAS sub-detectors. After the Phase-II upgrade FELIX will read out the entire ATLAS detector. FELIX supports GBT and FULL modes. Two hardware platforms can be used: the commercial VC709 for debugging and the custom FLX-712, final FELIX prototype, which will be used during the upcoming data taking. Integration and testing with the final FELIX prototype, the complete firmware and software infrastructure have proven that the system complies with readout requirements of the next data taking period. More tests are ongoing to increase the overall reliability and perform the final benchmarking. The

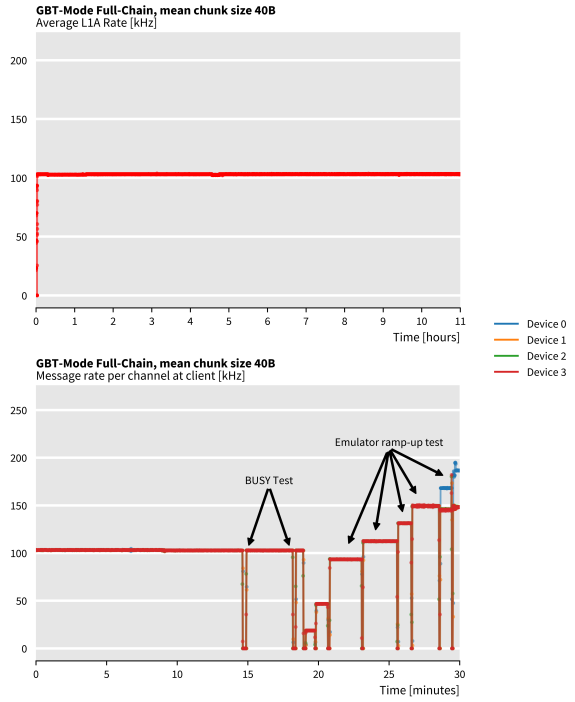

Fig. 6. GBT chain test of four logical FELIX devices. 40 B data streams at an average trigger rate ranging from 100 to 200 kHz are emulated and sent to the FELIX system. 432 e-links are activated. The upper plot shows the throughput processed by FELIX at 100 kHz trigger rate. The lower plot shows the throughput during the manual assertion of the BUSY signal or at trigger rates larger than 100 kHz.

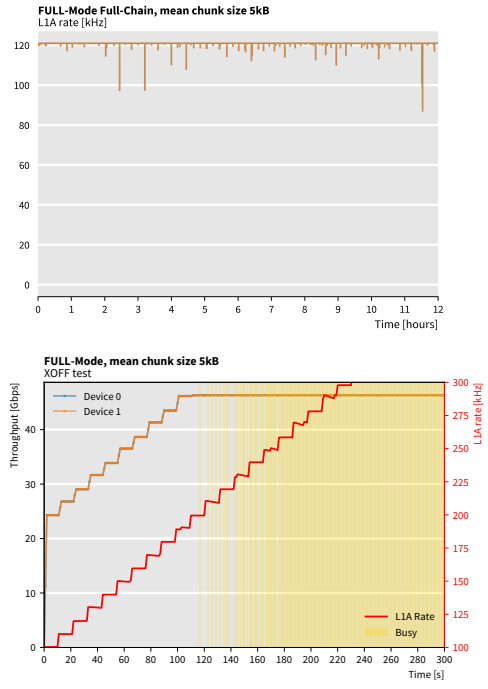

Fig. 7. FULL mode chain test of four logical FELIX devices. 5 kB data streams at an average trigger rate ranging from 100 to 300 kHz are emulated and sent to the FELIX system. 12 links are used. The upper plot shows the throughput processed by FELIX at 120 kHz trigger rate. The lower plot shows the throughput during trigger ramp-up at values larger than 100 kHz. For trigger rates larger than 200 kHz the XOFF signal is engaged.

procurement of the final system was in 2018-2019, the system installation will start in the first quarter of 2020.

REFERENCES

[1] ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider", 2008 JINST **3** S08003

[2] CERN GBT Project: GBTX Manual

[3] RD12 Project collaboration, B. G. Taylor, "TTC distribution for LHC detectors", IEEE Trans. Nucl. Sci. **45** (1998) 821.

[4] ATLAS Collaboration, "Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System", CERN-LHCC-2013-018

[5] ATLAS Collaboration, "ATLAS Liquid Argon Calorimeter Phase-I Upgrade Technical Design Report", CERN-LHCC-2013-017

[6] ATLAS Collaboration, "New Small Wheel Technical Design Report", CERN-LHCC-2013-006

[7] ATLAS Collaboration, "Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System", CERN-LHCC-2017-020

[8] Xilinx, UtraScale Architecture and Product Data Sheet: Overview, https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf

[9] Xilinx, UltraScale Architecture GTH Transceivers, http://www.xilinx.com/support/documentation/userguides/ug576-ultrascale-gth-transceivers.pdf

[10] VC709 Evaluation Board for the Virtex-7 FPGA, User Guide, https://www.xilinx.com/support/documentation/boards_and_kits/vc709/ug887-vc709-eval-board-v7-fpga.pdf

[11] Supermicro, Supermicro X10SRA-F Motherboard Model Specification, 2016, http://www.supermicro.nl/products/motherboard/Xeon/C600/X10SRA-F.cfm

[12] HiTech Global, HiTech Global HTG-710, http://www.hitechglobal.com/Boards/PCIE-CXP.htm

[13] M. Barros Marin *et al.*, "The GBT-FPGA core: features and challenges", 2015 JINST **10** C03021

[14] K. Chen *et al.*, "Optimization on fixed low latency implementation of the GBT core in FPGA", 2017 JINST **12** P07011

[15] OpenCores, "Wupper: PCIe DMA Engine for Xilinx FPGAs", https://opencores.org/projects/virtex7_pcie_dma

[16] 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

[17] G. Unel on behalf of the ATLAS TDAQ Collaboration, "FELIX: the New Detector Readout System for the ATLAS Experiment", PoS TWEPP2018 (2019) 140

[18] J. Schumacher on behalf of the ATLAS TDAQ Collaboration, "Utilizing HPC Network Technologies in High Energy Physics Experiments", doi:10.1109/HOTI.2017.25

[19] Jörn Schumacher on behalf of the ATLAS TDAQ Collaboration, "Interfacing Detector and Collecting Data for Large-Scale Experiments in High Energy Physics using COTS Technology", CERN-THESIS-2017-062

[20] W. Wu on behalf of the ATLAS TDAQ Collaboration, "FELIX: the New Detector Interface for the ATLAS Experiment", IEEE Trans.Nucl.Sci. **66** (2019) no.7, 986-992