

# 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.

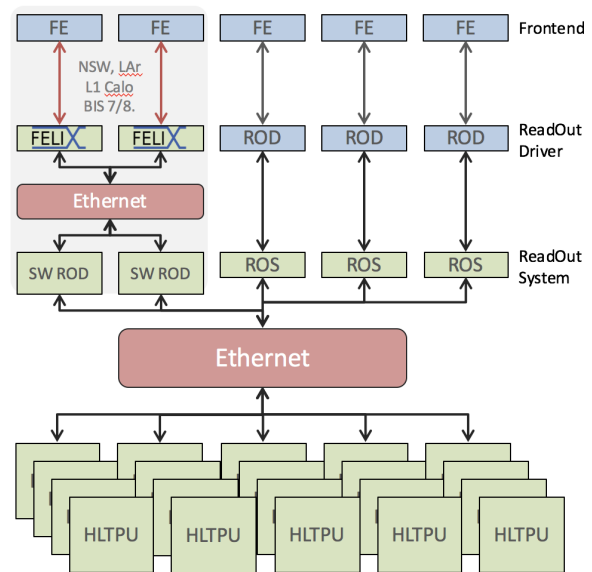


Fig. 1. ATLAS DAQ architecture in Run 3 upgrade

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.

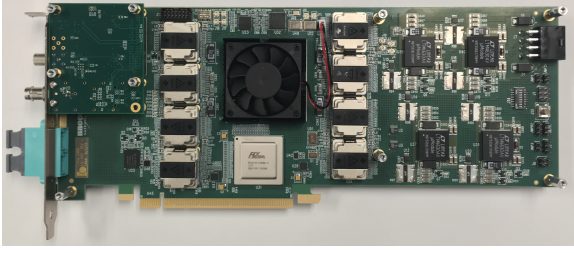


Fig. 2. FELIX final prototype FPGA PCIe card (FLX-712)

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 (see Section III-F) 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

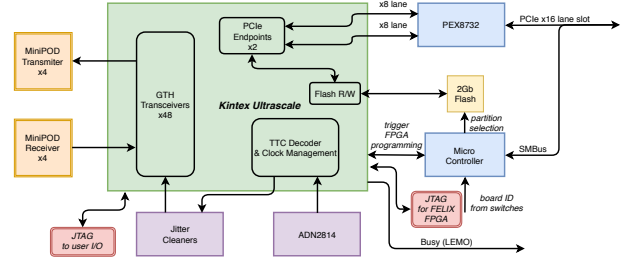


Fig. 3. Block diagram of the FELIX final prototype FPGA card

of the 48 optical links. A typical eye diagram at 12.8 Gb/s is shown in Fig. 4.

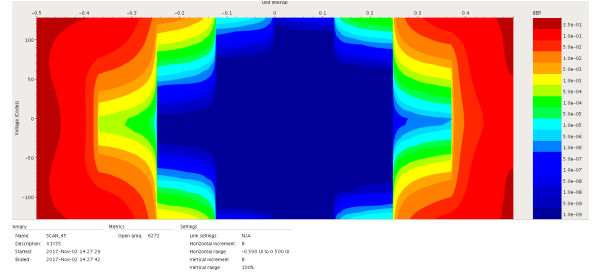


Fig. 4. The eye diagram for one optical transceiver channel at 12.8 Gb/s

In order to ease the update of FELIX firmware, an on-board parallel flash can store four different firmware bitfiles in separate partitions. These stored bitfiles can be updated and verified by FELIX software tools via the PCIe interface. A micro-controller (ATMEGA324A) [10] is used to control the reconfiguration of the FPGA from selectable bitfiles stored in the flash memory. Software tools in the host server communicate with the micro-controller via the System Management Bus. The micro-controller reads the status of on-board switches and uses it as the I<sup>2</sup>C slave address, which can be used as the board ID. The flash partition selection can be controlled by the FPGA, the micro-controller and by jumpers. FPGA firmware has highest priority, and the jumpers have lowest priority.

A mezzanine card has been developed to receive the TTC information. It is connected to the FLX-712 via a Samtec SEARAY connector, as shown in Fig. 5. It can be populated to interface to the LHC legacy TTC, TTC-PON or White Rabbit systems. In the configuration for the legacy TTC system, an on-board clock and data recovery ASIC (ADN2814) is used to recover the 160 MHz LHC TTC clock and data.

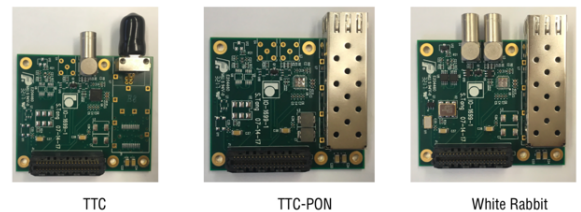


Fig. 5. FELIX timing mezzanine card for the FLX-712

The FPGA is equipped with a standard axial fan heat-sink.

It is estimated that the whole FLX-712 card will consume less than 64 W. The air flow in the server should be sufficient and no exotic cooling appears to be required (the card consumes less power than a GPU). The temperature of FLX-712 card installed in a host server can also be checked with software tools. For a project with 46 links of about 5 Gb/s, the temperature of FPGA internal diode, PCIe switch and MiniPODs are about 66°C, 55°C and 45°C respectively.

### 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. 6, 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.

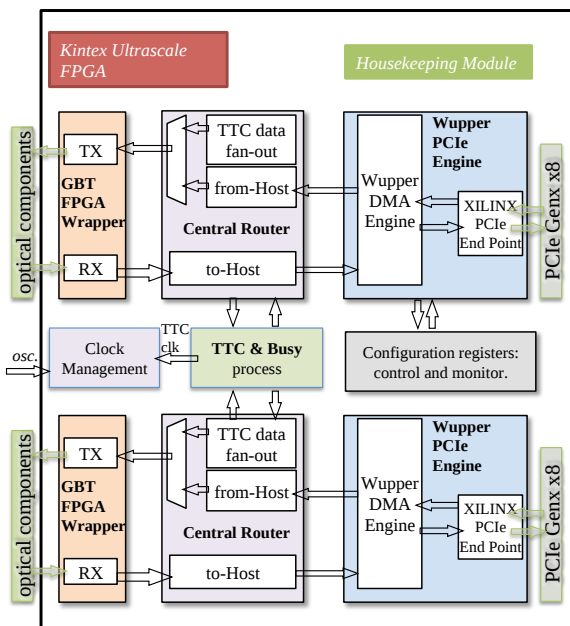


Fig. 6. Block diagram of the FLX-712 firmware

#### A. TTC Decoder

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 [11]. 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 A and B channels are interleaved bit-by-bit; the B-channel is further encoded by a Hamming code. The correct alignment of the 40.08 MHz LHC bunch crossing clock must be deduced from the A and B-channel streams. A state machine is used to sample these two data streams with the 160.32 MHz recovered clock from the ADN2814. It also separates the A-channel and B-channel information, extracts broadcast commands from the B-channel data stream, and provides a 40.08 MHz clock aligned to the bunch crossing clock. This clock is used to choose the correct phase of a 40.08 MHz clock generated by a Xilinx clock management module (MMCM) in the FPGA from the 160.32 MHz recovered clock. The architecture of the TTC decoder is shown in Fig. 7.

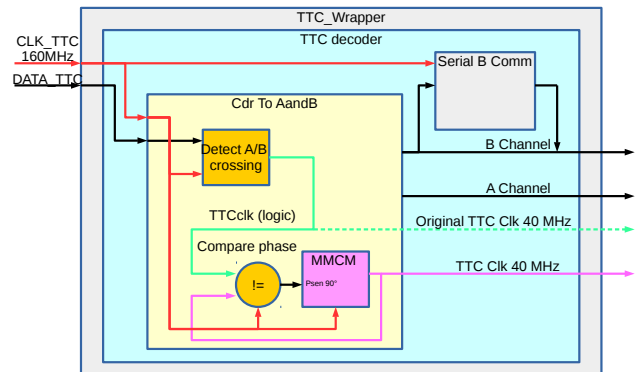


Fig. 7. Block diagram of TTC decoder

#### B. Clock Distribution

The generated 40.08 MHz TTC clock from the MMCM in Fig. 7 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. 8 shows which clock signals are generated and how they are used. For test purposes, it is also possible to use a local oscillator as the master clock.

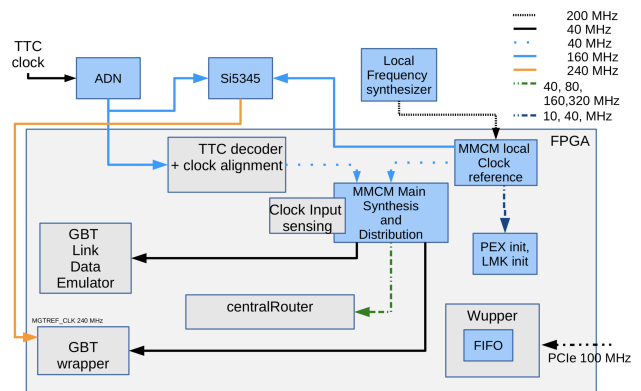


Fig. 8. Clock distribution for the FLX-712

### C. GBT Wrapper

The FELIX GBT wrapper is based on the CERN GBT-FPGA firmware with several performance improvements [12]. It encapsulates the Forward Error Correction (FEC) encoder/decoder, a scrambler/descrambler and a gearbox architecture, as shown in Fig.9. To decrease the latency, the frequency of the FEC encoder/decoder and scrambler/descrambler clock domain was increased to 240 MHz [13]. The GBT protocol supports GBT frame-encoding mode and wide-bus mode [12]. 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. In order to allow choosing between the GBT frame-encoding mode and wide-bus mode at run-time, two multiplexers are added: one for the FEC encoder and the other for the FEC decoder. A FSM (Finite State Machine) is implemented for automatic alignment of the GBT RX data stream. The registers of this GBT wrapper are mapped to the PCIe interface to allow software tools to control and monitor its status.

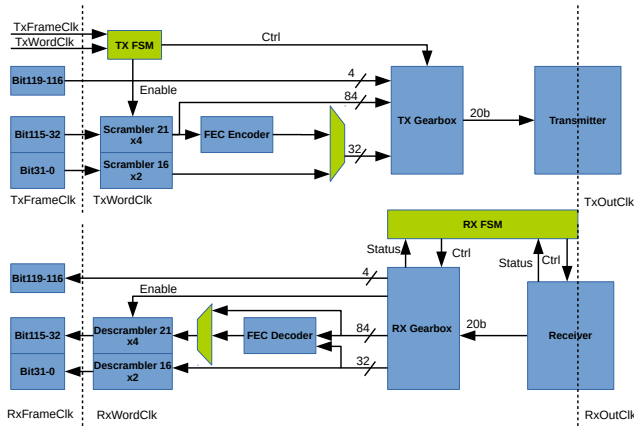


Fig. 9. The FELIX GBT wrapper [13]

### D. Central Router

The Central Router routes and formats data streams between the GBT wrapper and the PCIe DMA engine. It handles the two data path directions independently. On the GBT side, it implements a data manager for each link supporting both of the GBT frame-encoding data (80-bit) and wide-bus data (112-bit), according to the GBT configuration. On the PCIe engine side, there is a FIFO with a 256-bit wide port. A block diagram for the to-host path is shown in Fig. 10. In the central router, the data stream is organized in E-groups and E-links. For GBT frame-encoding mode, there are five E-groups in each direction. For wide-bus mode there are seven E-groups in the to-host direction and three in the from-host direction. Each E-group transfers 16 bits of data at 40 MHz. There are four possible E-link data widths in each E-group: 2, 4, 8 and 16 bits, corresponding to data rates of 80, 160, 320 and 640 Mb/s. The 640 Mb/s uses two adjacent 320 Mb/s lanes since it is not supported directly by the GBTx ASIC [6]. Each possible E-link configuration in an E-group is managed by an “E-proc” that processes the data and interfaces to a dedicated 2-Kbyte

E-link FIFO. The FELIX GBT configuration defines which E-links exist and therefore which E-proc’s are active.

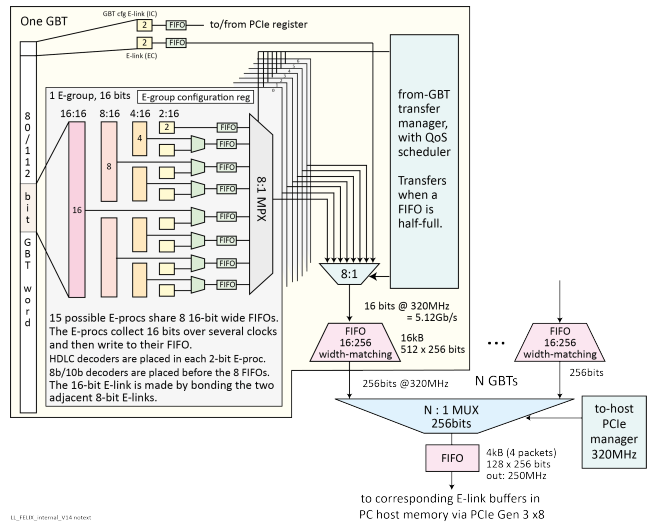


Fig. 10. Block diagram of the to-host path of the FELIX Central Router

### E. BUSY and Flow Control

FELIX supports both a BUSY and a flow control architecture, as shown in Fig. 11. Assertion of a BUSY signal is a request to the Central Trigger Processor (CTP) to stop generating Level-1 Accept triggers which eventually stops the data flow. Components in the data flow both upstream and downstream from FELIX can send busy-on and busy-off requests. Because BUSY assertion forces ATLAS dead time, its use should be limited to stopless recovery, start of run or emergency situations when buffers are almost full. Each FLX-712 card is capable of asserting BUSY via a LEMO connector on its panel. Flow control, on the other hand, recognizes that the congestion is likely only temporary and, assuming the data source has sufficient buffers, transmission can be paused without harm or data loss. FELIX can issue XON and XOFF flow control signals to its input links when its buffers become full. FULL mode uplinks, typically driven by FPGAs with buffers, may also support flow control. GBT mode uplinks, typically driven by front-end ASICs with small derandomizer buffers, so far do not handle flow control.

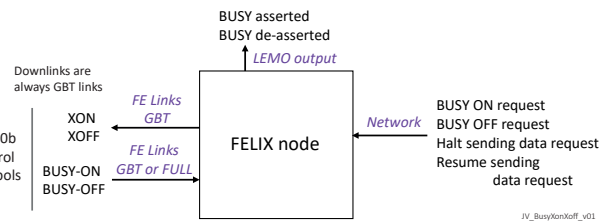


Fig. 11. FELIX BUSY logic

### F. PCIe Wupper

PCIe firmware, called Wupper, was designed to provide a simple Direct Memory Access (DMA) interface for the

Xilinx PCIe Gen3 hard block [14]. 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. For each 8-lane PCIe Gen3 endpoint, the throughput achieved is the theoretical maximum throughput of 64 Gb/s. For FLX-712, the 16-lane PCIe interface provides a theoretical maximum throughput of 128 Gb/s. A maximum effective throughput of somewhat more than 100 Gb/s has been observed. Eight 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 (one per descriptor), a status register for every descriptor is provided in the register map. The block diagram of the Wupper design is shown in Fig. 12. Its functional blocks can be categorized into two groups: DMA control and DMA write/read. The DMA control parses and monitors received descriptors. It also makes the descriptor status available to software via the PCIe interface. Depending on the address range of the descriptor, the pointer to the current address is handled by DMA control and incremented every time a TLP (Transaction Layer Packet) completes. DMA control can handle a circular buffer DMA if this is requested by the descriptor. DMA control also contains a register map, with addresses of the descriptors, status registers and external registers for the user space register map. 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.

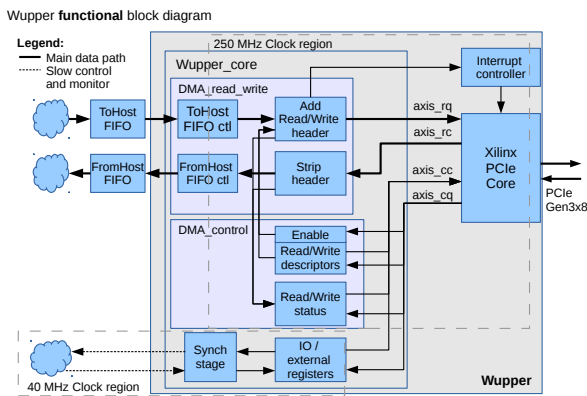


Fig. 12. Structure of the Wupper PCIe engine

#### IV. FELIX SOFTWARE

The FELIX software suite has different layers: for example, low-level software tools, test software and production software, as shown in the Fig. 13. Access to the FELIX hardware

level is controlled 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. Cmem\_rcc is a driver that is already used for more than ten years on ROS (ReadOut System) PCs and VMEbus SBCs (Single Board Computer). 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.

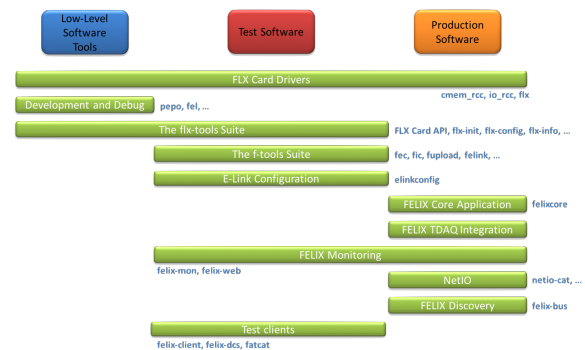


Fig. 13. FELIX software tools

FELIX supports dynamic configuration of E-links on GBT links, such as E-link width, encoding of an E-link's data and whether an E-link is disabled or enabled. Such a configuration should of course match the configuration of the front-end GBT links. The Elink Configurator is a graphical tool developed to offer a user-friendly interface for creating and modifying a configuration, as shown in the Fig. 14. It displays a graphical representation of the division of E-links in 16 bits of the GBT frame (a so-called E-group) for both to-host and from-host directions. The user can enable or disable any 16-bit E-group and define its E-links as needed. The Elink Configurator tool is also capable of saving the configuration to a local file and loading the configuration from a previous saved file. It supports the two modes in which the GBT links can be used, i.e. GBT frame-encoding and wide-bus modes, and also supports FULL mode links.

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. 15. 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.

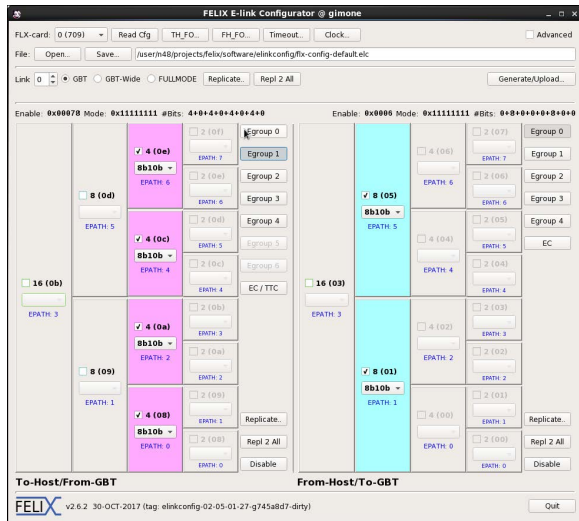


Fig. 14. Screen shot of the FELIX Elink Configurator tool

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.

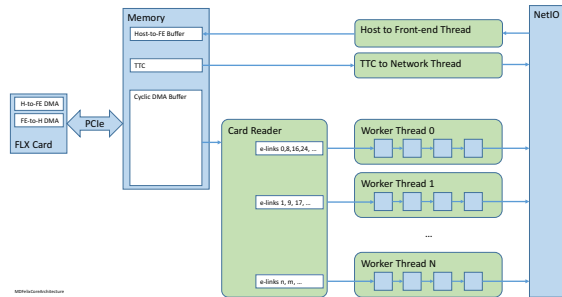


Fig. 15. The felixcore application architecture

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 APIs. 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 and similar network technologies [15]. 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. 16, 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.

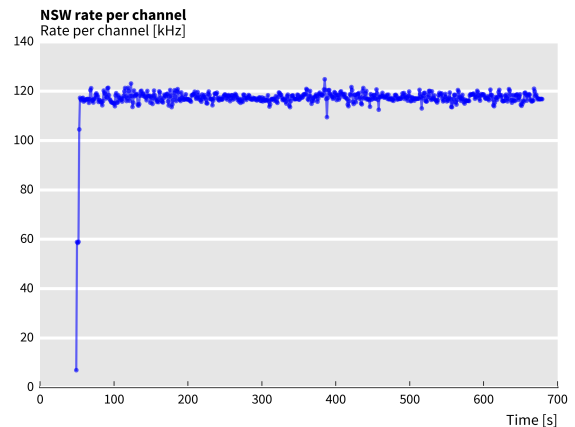


Fig. 16. Felixcore performance for GBT mode

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. 17. 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.

## 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)

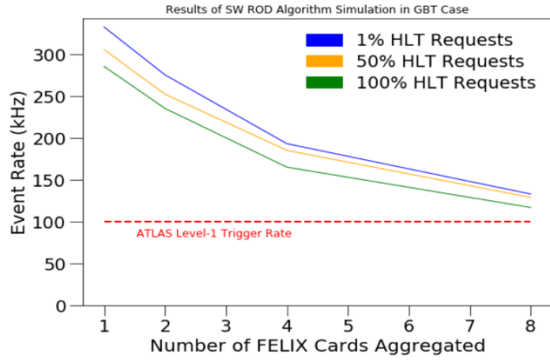


Fig. 17. Software ROD performance

system [3] [16]. 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, I<sup>2</sup>C, SPI and JTAG interfaces [17]. Other highlights also include the ability to read out ADC monitoring data and configure the GBTx on the L1DDC board [18]. 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. 18. GBT-SCA chips are used to control the power rails, I<sup>2</sup>C buses and also perform on-board temperature measurement [17]. 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 [19] [20].

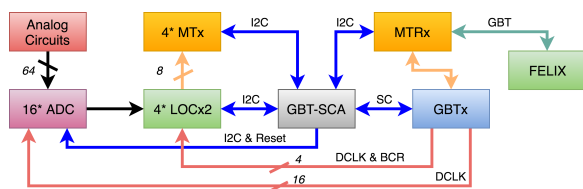


Fig. 18. GBTx and GBT-SCA connections in the LTDB prototype

#### 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 [21]. 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. 19. 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  $10^{-15}$ .

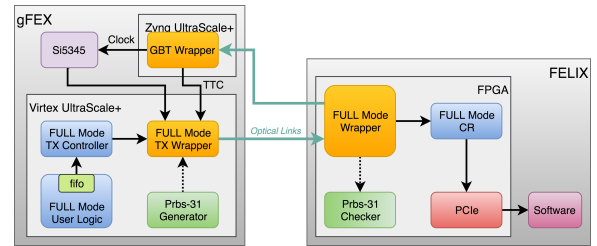


Fig. 19. Block diagram of gFEX and FELIX integration test

## 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](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](http://www.xilinx.com/support/documentation/ip_documentation/ibert_ultrascale_gth/v1_3/pg173-ibert-ultrascale-gth.pdf)
- [10] A. Borga *et al.*, “The C-RORC PCIe card and its application in the ALICE and ATLAS experiments”, 2015 JINST **10** C02022
- [11] 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
- [12] M. Barros Marin *et al.*, “The GBT-FPGA core: features and challenges”, 2015 JINST **10** C03021
- [13] K. Chen *et al.*, “Optimization on fixed low latency implementation of the GBT core in FPGA”, 2017 JINST **12** P07011
- [14] A. Borga *et al.*, PCIe Gen3x8 DMA for virtex7, <http://stackoverflow.com/>
- [15] Libfabric, OpenFabrics, <https://ofiwg.github.io/libfabric>
- [16] H. Chen *et al.*, “The prototype design of gFEX - a component of the 11calo 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
- [17] A. Caratelli *et al.*, “The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments”, 2015 JINST **10** C03034
- [18] 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
- [19] 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
- [20] 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
- [21] 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