A new approach to front-end electronics interfacing in the ATLAS experiment

J. Anderson[a], A. Borga[d]*, H. Boterenbrood[d], H. Chen[b], K. Chen[b], G. Drake[a], M. Dönszelmann[g], D. Francis[c], B. Gorini[c], F. Lanni[b], G. Lehmann Miotto[c], L. Levinson[f], J. Narevicius[f], A. Roich[f], S. Ryu[a], F. Schreuder[d], J. Schumacher[c,e], W. Vandelli[c], J. Vermeulen[d], W. Wu[b], J. Zhang[a]

[a] *Argonne National Laboratory,*
9700 South Cass Avenue B109, Lemont, IL 60439, USA
[b] *Brookhaven National Laboratory,*
Brookhaven National Laboratory, P.O. Box 5000, Upton, NY 11973-5000, USA
[c] *CERN, CH-1211 Geneva 23, Switzerland*
[d] *Nikhef National Institute for Subatomic Physics / University of Amsterdam,*
Science Park 105, 1098 XG Amsterdam, Netherlands
[e] *Department of Computer Science, University of Paderborn,*
Pohlweg 47, 33098 Paderborn, Germany
[f] *Department of Particle Physics, The Weizmann Institute of Science,*
Rehovot 76100, Israel
[g] *Radboud University Nijmegen*
Comeniuslaan 4, 6525 HP Nijmegen, Netherlands

E-mail: andrea.borga@nikhef.nl

ABSTRACT: For new detector and trigger systems to be installed in the ATLAS experiment after LHC Run 2, a new approach will be followed for Front-End electronics interfacing. The FELIX (Front-End LInk eXchange) system will function as gateway connecting: on one side to detector and trigger electronics links, as well as providing timing and trigger information; and on the other side a commodity switched network built using standard technology (either Ethernet or Infiniband). The new approach is described in this paper, and results achieved so far are presented.

KEYWORDS: ATLAS experiment; Data acquisition concepts; Data acquisition circuits.

*Corresponding author.

Contents

1. **Introduction** 1

2. **Data handling** 2

3. **FLX card hardware** 4

4. **FLX card firmware** 5

5. **Test achievements** 8

1. Introduction

The ATLAS [1] experiment at the LHC will be upgraded with new detectors and trigger electronics during the next long shutdown, now foreseen for the period 2019 - 2020: in particular the New Small Wheel muon detectors [2] and new first-level trigger electronics making use of data of the Liquid-Argon (LAr) calorimeter [3][4] will be installed. A new subsystem of the ATLAS on-line system, called FELIX (FrontEnd LInk eXchange), will be used as gateway between dedicated links connecting to detectors and trigger electronics, links providing timing and trigger information, and a commodity network (Ethernet or Infiniband). The connectivity of the FELIX system is illustrated in Figure 1. Software running on server PCs connected to a commodity network - using "Commercial Off The Shelf" (COTS) switches - will interact via FELIX with on-detector and trigger electronics for configuration, control, monitoring and calibration. Via FELIX event data to be read out will be passed to server PCs (indicated with "event readout" in Figure 1), where software will build event fragments, which will be in turn passed on to the Read Out System (ROS), a subsystem of the ATLAS DAQ system that receives and buffers data from all sub-detectors and trigger systems[1].

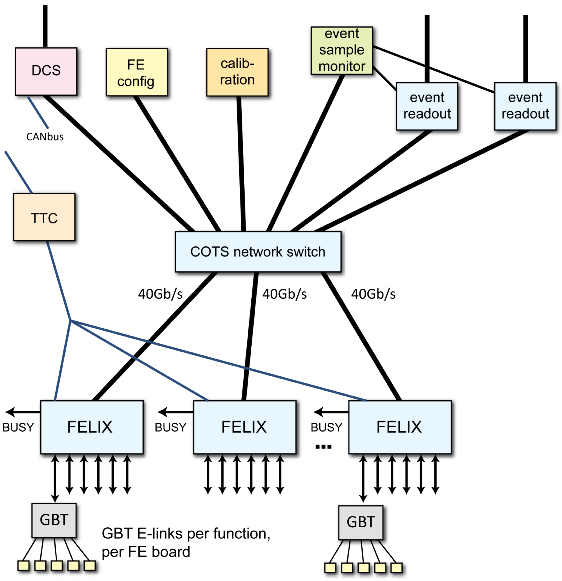

Figure 1. Overview of the connectivity of the FELIX system (DCS refers to the Detector Control System, "FE Config" to Front-End configuration).

[1] Currently readout is achieved by means of sub-detector specific custom electronics, the Read Out Drivers (RODs). These receive event data via dedicated point-to-point links, process it, typically in FPGAs, and output the results via other dedicated point-to-point links to the ROS.

Timing and trigger information will be received by FELIX from the Timing, Trigger and Control system (TTC) and will be forwarded to Front-End electronics with low and fixed latency. This is facilitated by the use of the GBT protocol [5], which will be used for the links connecting to detector electronics and for part of the links connecting to trigger electronics[2]. Trigger information needed for reliable fragment building will also be forwarded via the network. FELIX will be able to generate logical signals (BUSY signals) for suppressing first-level trigger accepts.

The novelty of the approach consists of minimization of the amount of custom electronics and of non-standard point-to-point links by implementing functionality in software running on server PCs and by exploiting commodity network technology. In the past this approach would not have been possible because of inadequate performance, but thanks to today's technology this is no longer true. On the contrary, clear advantages are less dependence on firmware and custom electronics, and therefore fewer problems with long-term maintenance, stock and spares management, and component availability. Furthermore increasing the available processing power in case of need is in principle a straightforward operation, consisting of upgrading PCs and/or networking and/or adding more PCs, for which no design effort may be required. The functionality of FELIX itself is sub-detector independent and should also not depend on what state the experiments is in, i.e. whether data taking is in progress or whether dedicated calibration procedures are being executed. FELIX therefore should always be available. The FELIX approach will be used for new detectors and trigger systems for LHC Run 3, and is foreseen to be used ATLAS-wide for LHC Run 4.

The FELIX system itself will be implemented with server PCs, each with at least one PCIe card referred to as the FLX card. An FLX card has an FPGA interfacing to links connecting to detector and trigger electronics. For current technology a number of standard 4.8 Gb/s GBT links (with a net throughput of 3.2 Gb/s when using forward error correction) of about 16 - 20 per FPGA seems to be feasible. The PCIe interface is either an 8 or 16 lane PCIe Gen3 interface. The FLX cards have a dedicated optical input for the TTC system and an open-collector (Lemo) output for generating BUSY signals. For development commercially available boards are used together with small custom add-on boards, see section 3. The PCs will also be equipped with suitable commodity network interfaces; currently dual-port 40 Gb/s Ethernet interfaces are used. In this paper the focus is on the FLX card, its firmware and on tests done so far. Software aspects have been described and discussed in details in other papers [6][7].

2. Data handling

The configuration of the FELIX PC is depicted in Figure 2. An FPGA on the FLX PCIe card interfaces to the links connecting to detectors or trigger electronics. Data are moved from FPGA to the PC main memory and from the PC main memory to the FPGA using Direct Memory Access (DMA) control. In the current implementation of the firmware eight DMA channels are available. The application running on the PC - the FELIX application - manages the data transfers and the network connections. Data received from the FPGA is routed to one or more network endpoints based on extracted meta-information.

[2] For the off-detector part of trigger systems FPGAs will be used for interfacing to the links connecting to FELIX. If for these links there is no need for multiplexing many slow links on the same link (functionality provided by the up to 42 E-links of the GBT protocol) and for fixed latency transfers, probably a protocol other than GBT will be preferred. A suitable simple protocol will be selected.

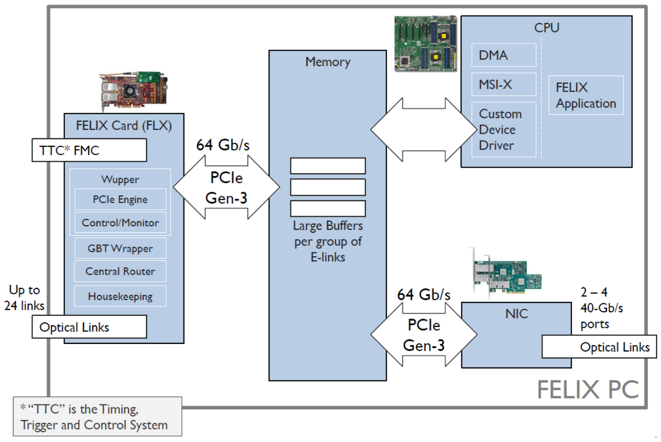

Figure 2. Configuration of a FELIX PC. One or several FLX cards stream data to host PC, application software running on the PC handles and forwards the data to the network and also forwards data received via the network to the FLX cards.

In the following it is assumed that the standard GBT protocol is used for the links to which the FLX card connects, these links are referred to as "GBT links". As noted earlier, protocols other than the standard GBT protocol may also be used. However, the standard GBT protocol is probably the most demanding with respect to FPGA and software resources needs, since for each GBT link up to 42 bonded E-links, i.e. 42 bi-directional independent data streams, may have to be handled.

For Run 3 about 2000 (11000 for Run 4) detector links will have to be connected to FELIX systems. To ensure a dense, cost- and space-efficient system, each FELIX PC will need to interface to as many links as possible. The channel density is mainly limited by the resources available in mid-priced/high-end FPGAs and, as mentioned earlier, is estimated to be feasible up to 20 detector links per FLX card. If for event data readout each of the links operates at a data rate of 3.2 Gb/s, where data flowing across E-links is 8B/10B encoded, the FELIX firmware and software will need to be able to process data, after 8B/10B decoding, at roughly 6.4 GB/s. Furthermore, with 20 GBT links the total number of data streams could be 840 (the maximum number of E-Links specified by the standard GBT protocol, i.e. 42) in one direction and 840 in the reverse direction. Fortunately high throughput is only required for data flowing via the FPGA to the host. It has been measured that the PCIe Gen3 8 lane interface currently used can sustain about 6.3 GB/s.

Data flowing via the E-links will be organised in packets of arbitrary length with packet boundaries indicated by out-of-band data patterns in case of encoded data, or by reserved bit patterns. These packets are referred to as "chunks": they can be event fragments, but also control or configuration commands or responses to these commands.

For event fragments a high throughput of chunks with arbitrary lengths is needed, while data streams from different E-links should be kept separate. This is achieved in the following way: data received via a single E-link is packed by the FPGA into blocks of fixed size (currently 1 kByte) with a 4-byte header. If within a certain configurable time-out an insufficient amount of data has been received to fill a complete block, the block is nevertheless transferred to the host, padded with a null chunk so as to complete a 1 KByte block. A chunk may be completely contained in a single block or may consist of an arbitrary number of sub-chunks spanning as many blocks as needed. Each complete chunk or sub-chunk has a trailer containing the length of the chunk or sub-chunk and flags indicating whether it is the trailer of a complete chunk, or of a sub-chunk that is the first, last or a middle sub-chunk. By collecting data in blocks the data can be transferred in bursts across the PCIe interface; as required to obtain a high throughput due to the overheads associated with the packet oriented PCIe transfer protocol. Each block has a 4-byte header which contains an E-link identifier and a sequence number, so that it is possible to associate data with an

E-link and to check that no blocks are lost and also that blocks are handled in the correct order, as needed for correct reconstruction of chunks spanning more than one block. The internal interface in the FPGA to the DMA engine can be a simple FIFO interface and there is no need to initiate a DMA transfer for each individual 1 KB block. PCIe transactions may be initiated to transfer many blocks, so that a high throughput can be achieved. However, once a multi-block DMA transfer is finished, a new DMA transfer may need to be initiated within a few μs to avoid overflow of buffers in the FPGA. This risk is minimized by using continuous DMA transfers, which are supported by the DMA controller, by means of a cyclic buffer for which a write pointer is maintained by the DMA controller and a read pointer is maintained by the application software. The DMA transfer halts when the write pointer becomes equal to the read pointer and resumes after the read pointer changes. By using a large cyclic buffer, and if processing of the data on average is fast enough, the probability for overflow will be low and the probability of interruption of data transfer into the PC will be small.

Using the mechanism described above the data arrives in the PC as up to eight (the number of DMA channels) continuous streams of blocks. The FELIX application needs to handle the data and forward it via the network to the correct destinations. The reception of corrupt data via one or more E-links is not likely to cause problems with the demultiplexing of data, as all blocks have the same size, blocks with corrupted data can just be skipped.

In order to keep the resources needed manageable, data to be forwarded via the GBT links can only be passed as fixed size packets of 32 bytes from the host to the FPGA. Of the 256 bits of each packet the first 16 bits are reserved, 11 bits are used for specifying the GBT link and the E-link via which the data should be passed, 4 bits are used for specifying the length of the payload data and the last bit indicates whether the payload is complete. Any unused bytes will be ignored and will not be passed via the E-link indicated in the header of the packet.

3. FLX card hardware

To prove that the FELIX approach is viable and for firmware development at ANL, BNL, CERN, Nikhef and Weizmann Institute a board from HiTech Global (HTG-710) is used in a suitable server PC. Figure 3 shows the card, referred to in this context as the FLX-710. Up to 24 bi-directional optical links can be connected to the card. A custom FMC mezzanine card (called TTCfx) provides an additional optical input for the TTC (Trigger, Timing and Control) system via an ST optical connector. The TTCfx hosts TTC clock and data recovery circuitry (ADN2814) and a clock jitter cleaner

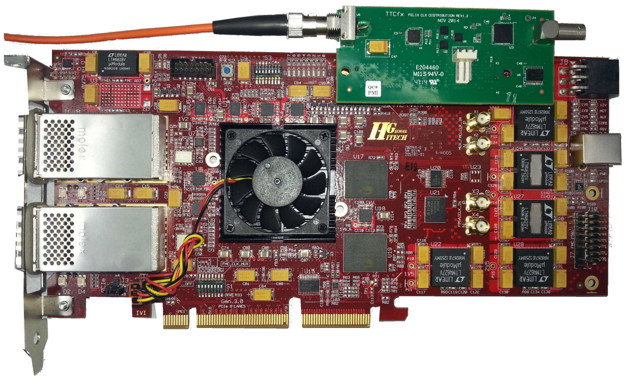

Figure 3. The HiTech Global HTG-710 FPGA development card (FLX-710) used in the demonstrator system and for firmware development. Each of the two CXP cages on the left can host a 12-way bi-directional optical transceiver with MTP/MPO optical connectors. The board mounted on the FMC connector at the top of the board is a TTCfx board.

(CDCE62005), as well as a Lemo output that can be used as BUSY signal for throttling the Level-1 trigger if required. A second version of the TTCfx is being redesigned with another jitter cleaner (Si5338) to achieve better stability and phase reproducibility, also after a system power cycle. The FLX-710 is equipped with a Xilinx Virtex-7 690T FPGA and is connected to the host system via an 8-lane PCIe Gen-3 interface. Via JTAG both the FPGA and an on-board EEPROM can be programmed. The PCs are also equipped with a dual-port Mellanox ConnectX-3 card, which can be configured either as a 40 Gb/s Ethernet or a 56 Gb/s Infiniband interface. Furthermore, the Xilinx VC-709 evaluation kit (referred to as FLX-709), which makes use of the same FPGA as the FLX-710, with up to four SFP+ input links, is also supported. The FLX-709 is targeting detector Front End electronics development, and it is preferred for this purpose because of cost and availability.

The final system may make use of a board (BNL-711) being developed for the DAQ platform for the LTDB (Liquid Argon Trigger Digitizer Board) production test stand. The card will feature a Xilinx Kintex Ultrascale XCKU115 FPGA, with up to 48 duplex optical links based on MiniPODs, a PCI Gen3 x16 interface based on a PCIe switch (PEX8732) and all the features of the TTCfx board. Figure 4 shows a simplified block diagram of the card. The first prototypes are being produced and are expected to be available by the end of 2015.

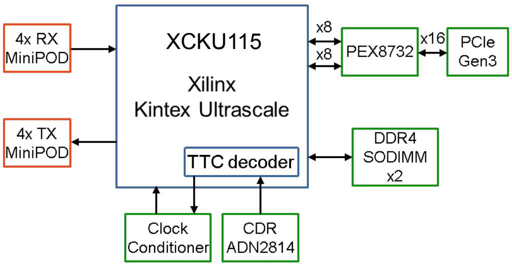

Figure 4. Block diagram of the BNL-711.

4. FLX card firmware

The main functional blocks of the FLX firmware are shown in Figure 5. For interfacing to GBT links their tasks are : (i) GBT link handling, including stable clocking using the TTC clock, (ii) routing with fixed and low latency of timing and trigger information distributed via the TTC system to multiple E-links aggregated in GBT links, (iii) internal data buffering and routing, (iv) data transfer to and from the host, (v) forwarding of TTC information to the host.

The GBT protocol can be used with or without Forward Error Correction (FEC), i.e. in "standard" or "wide" modes. GBT protocol specific functionality may not be needed for links connecting off-detector electronics with FELIX, but higher throughput may be desirable. It is foreseen that these links will run at 9.6 Gb/s using a simple protocol yet to be defined.

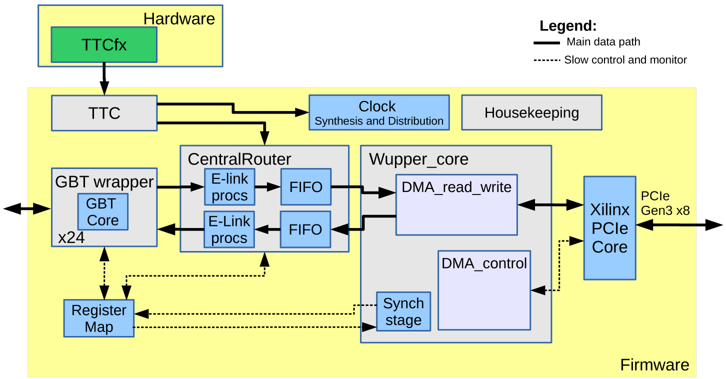

Figure 5. FLX firmware simplified block diagram.

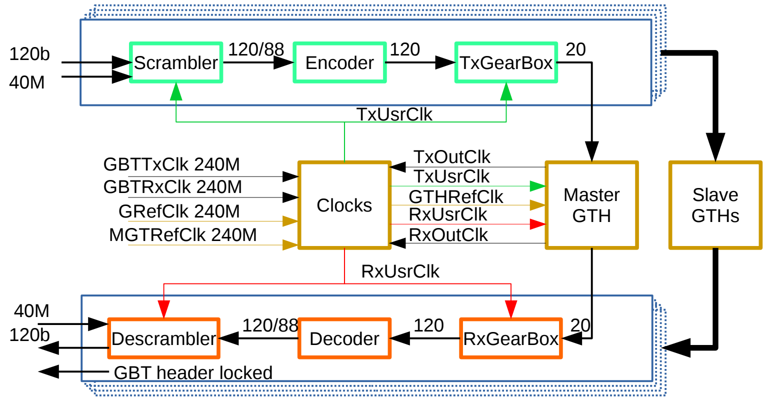

Figure 6. Block diagram of the FELIX GBT Wrapper.

For links making use of the GBT protocol the FELIX GBT Wrapper module, depicted in Figure 6, transfers the data from/to the detector Front-Ends to the rest of the FELIX logic. It is derived from the CERN GBT-FPGA design (version 3.0.2) and uses Xilinx GTH quads (four channels with a dedicated PLL) as its unit. The design can be dynamically scaled at firmware build time. Every channel incorporates a GBT encoder and decoder. The main modifications with respect to the original design are: (i) the core is now transceiver independent, (ii) a run-time choice of the GBT mode (with or without FEC) has been added, (iii) a simpler RxGearBox with Rx alignment replaces the RxGearBox and frame-aligner, (iv) the alignment operation can be now automatically controlled via both firmware and software, (v) the scrambler and descrambler have been moved from the 40 MHz clock domain to the 240 MHz domain. Especially for the path toward the Front-End fixed, low and reproducible latency is key. The core had been therefore carefully tuned for low latency performance, currently achieving figures of 52.9 ns for transmission, and 59 ns and 50.7 ns for reception of data respectively with or without FEC.

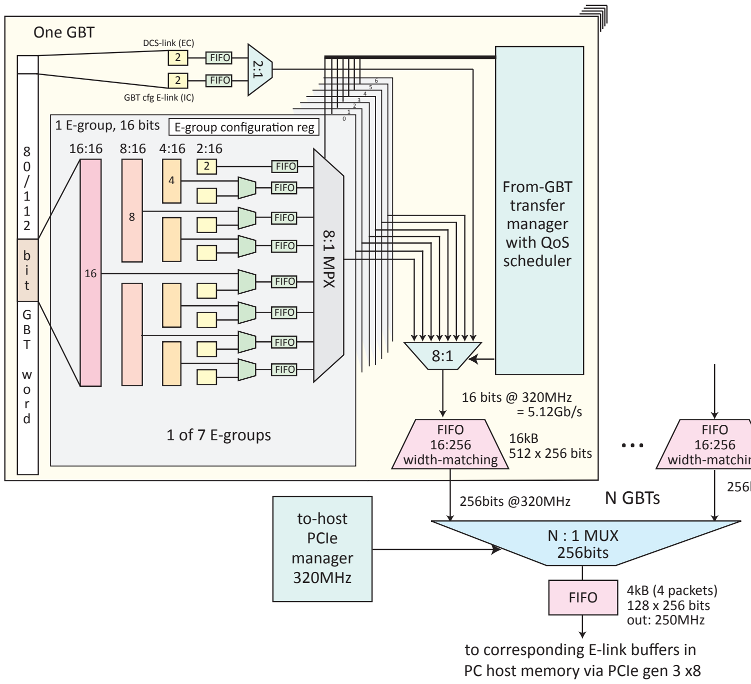
One GBT
DCS-link (EC)
2
FIFO
2:1
2
FIFO
GBT cfg E-link (IC)
80/112 bit GBT word
1 E-group, 16 bits
E-group configuration reg
16:16
8:16
4:16
2:16
2
4
8
16
FIFO
8:1 MPX
0
1
2
3
4
5
6
From-GBT transfer manager with QoS scheduler
8:1
16 bits @ 320MHz = 5.12Gb/s
1 of 7 E-groups
FIFO 16:256 width-matching
16kB 512 x 256 bits
...
FIFO 16:256 width-matching
256bits
256bits @320MHz
N GBTs
to-host PCIe manager 320MHz
N : 1 MUX 256bits
FIFO
4kB (4 packets) 128 x 256 bits out: 250MHz
to corresponding E-link buffers in PC host memory via PCIe gen 3 x8

Figure 7. Schematic overview of the flow and queuing of data received via E-links in the Central Router.

The Central Router handles routing and queuing of the serial data transferred via the up to 42 E-links of each GBT link (Figure 7). Per GBT link E-links are grouped in "E-groups", each associated with 16 bits in the GBT frame. An "E-group" therefore has either 8 2-bit, 4 4-bit or 8 2-bit E-links, and also supports one (non-standard) 16-bits E-link. There are five E-groups if FEC is used, as in that case there are 80 bits available in the GBT frame for E-links. In each GBT frame there are four bits reserved for two dedicated 2-

bit E-links, one for GBTx chip configuration (Internal Control, IC), the other (External Control, EC) to be used by the Detector Control System (DCS). These two E-links are indicated at the top left of Figure 7. If required, the Central Router performs 8B/10B or HDLC (High-Level Data Link Control)[3] decoding and detects chunk boundaries. Chunk trailers are added to the data and blocks of data are formed. Each E-link has a dedicated FIFO for queuing its blocks for transfers to the host. Arbitrating block transfers to the PCIe Engine is another task of the Central Router. For data to be output via the E-links (not indicated in the figure) 8B/10B coding or HDLC coding is performed.

Data are streamed from the FPGA to host PC memory for packet processing and routing by the PC, as described in [6] and [7]. In order to sustain the high data rates a high-bandwidth interface is required towards the PC. A custom PCIe Engine has been developed for this purpose, which has been published as an LGPL OpenCore project [8]. The PCIe Engine is called Wupper[4], expressing the idea of something (in this case data) "jumping" from one side to the other of the PCIe interface. Figure 8 shows its functional block diagram.

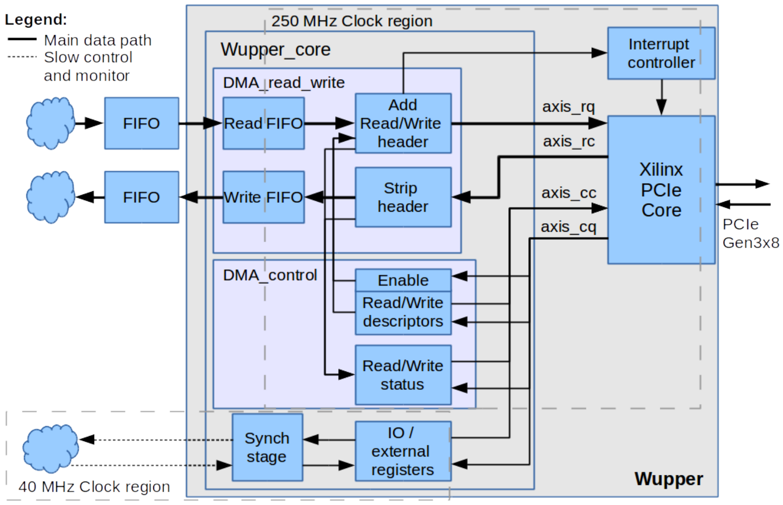

Figure 8. Functional block diagram of Wupper, the PCIe Engine.

Wupper provides a simple DMA interface for the Xilinx Virtex-7 PCIe Gen3 hard block, specifically designed for its 256 bit wide ARM AMBA AXI4-Stream interface. Wupper provides an interface to a standard FIFO with the same width as the Xilinx AXI4-Stream interface (256 bits). It is running at 250 MHz, therefore its maximum throughput is 64 Gb/s. The user application side of the FPGA simply reads or writes to the FIFO; Wupper handles the transfers into or from host PC memory, according to the addresses and transfer direction specified in the DMA descriptors. The core manages a set of DMA descriptors, with an *address*, a $read/\overline{write}$ flag, the $transfersize$ (number of 32 bit words) and an *enable* line, which are mapped as normal PCIe memory or IO registers. A status register for every descriptor is provided in the register map for detecting pending/processed requests. As already described in section 2, automatic cyclic transfer operations with wrap-around addressing are supported. User configurable selective MSI-X interrupts are also available. The register map is synchronized inside Wupper to an independent lower clock frequency to ease design timing closure.

[3] HDLC encoding/decoding is required for communication to Slow Control Adapter (SCA) ASICs via E-links.

[4] A wupper is a person performing the act of bongelwuppen, the version from the Dutch province of Groningen of the Frisian sport Fierljeppen (canal pole vaulting).

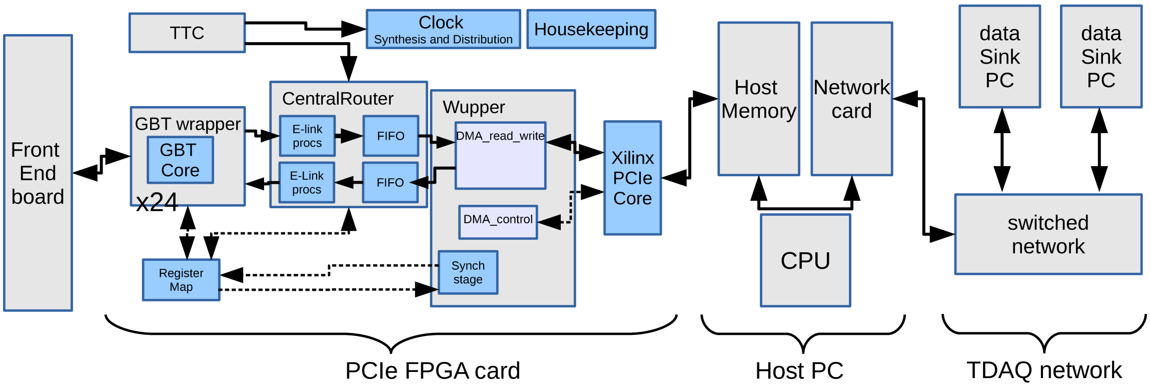

Figure 9. Architecture of the full test system.

5. Test achievements

Three categories of tests can be distinguished:

1. Testing with a suitable receiver the quality of the bunch crossing clock, as transferred via dedicated E-links, and determining the variation in latency of Level-1 accept signals. For this type of test either a Xilinx KC705 Development Kit or the CERN GLIB [9] board is used as receiver; furthermore an FMC card with a real GBTx chip has also been used.

2. Testing data integrity and throughput performance for data transfers from a suitable data source via the FLX card and its PCIe interface to the host PC. Loopback connections are possible with an FLX card in conjunction with data generators implemented in the FPGA. The GLIB board can be used as data source, as well as the KC705. In an extended version of this test the full data path is exercised by forwarding the data to one or several network destinations.

3. Testing with a suitable Front-End receiver the integrity of data forwarded to it via the network, through the host PC and FLX card. The receiver can be either a GLIB board, a KC705, or another FLX board, loopback connections with an FLX board are also possible.

Currently tests of Type 1 and Type 2 have been accomplished, using data generators implemented in the FPGA, and internal as well as external loopback of the data. The expected behaviour had been observed. For Type 2 testing of the full data path is in sight, while data handling in the PC and transfers across the network using software generated test data have already been tested separately. Tests of Type 3, with less demanding performance requirements with respect to throughput, are imminent.

References

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

[2] The ATLAS Collaboration, *New Small Wheel Technical Design Report,* 2013, CERN-LHCC-2013-006, ATLAS-TDR-020, https://cds.cern.ch/record/1552862.

[3] The ATLAS Collaboration, *ATLAS Liquid Argon Calorimeter Phase-I Upgrade Technical Design Report,* 2013, CERN-LHCC-2013-017, ATLAS-TDR-022, http://cds.cern.ch/record/1602230.

[4] The ATLAS Collaboration, *Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System,* 2013, CERN-LHCC-2013-018, ATLAS-TDR-023, http://cds.cern.ch/record/1602235.

[5] *CERN GBT Project: the GBTx link interface ASIC*, https://espace.cern.ch/GBT-Project/GBTX/Manuals/gbtxManual.pdf.

[6] J. Anderson et al., *FELIX: a High-Throughput Network Approach for Interfacing to Front End Electronics for ATLAS Upgrades*, in *Proceedings CHEP2015 conference,* Okinawa, Japan, April 2015, ATL-DAQ-PROC-2015-014, https://cds.cern.ch/record/2016626.

[7] J. Schumacher et al., *Improving packet processing performance in the ATLAS FELIX project: analysis and optimization of a memory-bounded algorithm*, in *DEBS '15 Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, pp. 174–180, 2015.

[8] *PCIe Gen3x8 DMA for Xilinx Virtex-7*, http://opencores.org/project,virtex7_pcie_dma.

[9] *CERN GLIB Project*, https://espace.cern.ch/project-GBLIB/public/default.aspx.