SISSA

A generic firmware core to drive the Front-End GBT-SCAs for the LHCb upgrade

View the article online for updates and enhancements.

A generic firmware core to drive the Front-End GBT-SCAs for the LHCb upgrade

F. Alessio,[a,1] **C. Caplan,**[b] **C. Gaspar,**[a] **R. Jacobsson**[a] **and K. Wyllie**[a]

[a]*CERN,*
  Geneva, CH-1211, Switzerland
[b]*CBPF,*
  Rio de Janeiro, Brazil

E-mail: Federico.Alessio@cern.ch

ABSTRACT: The LHCb experiment has proposed an upgrade towards a full 40 MHz readout system in order to run between five and ten times its initial design luminosity. The entire Front-End electronics will be upgraded in order to cope with higher sub-detector occupancy, higher data rate and to work in a complete trigger-less fashion. In this paper, we describe a novel way to transmit slow control information to the Front-End electronics, by profiting from bidirectional optical connections and the GBT and GBT-SCA chipset capabilities. The implementation and preliminary validation tests are shown as well.

---

[1]Corresponding author.

Contents

1 The upgrade of the LHCb experiment

The LHCb experiment [1] is a high-precision experiment at the LHC devoted to the search for New Physics by precisely measuring its effects in CP violation and rare decays. By applying an indirect approach, LHCb is able to probe effects which are strongly suppressed by the Standard Model, such as those mediated by loop diagrams and involving flavor changing neutral currents.

In the proton-proton collision mode, the LHC is to a large extent a heavy flavor factory producing over 100,000 bb-pairs every second at the nominal LHCb design luminosity of $2 \times 10^{32}\, \mathrm{cm}^{-2}\mathrm{s}^{-1}$. Given that bb-pairs are predominantly produced in the forward or backward direction, the LHCb detector was designed as a forward spectrometer with the detector elements installed along the main LHC beam line, covering a pseudo-rapidity range of $2 < \eta < 5$ well complementing the other LHC detectors ranges.

LHCb proved excellent performance in terms of data taking [2] and detector performance over the period 2010–2012 accumulating $\sim$3 fb^{-1} of data and it is foreseen to accumulate other $\sim$5 fb^{-1} over the period 2015–2018. Due to the foreseen improved performance of the LHC accelerator, the prospect to augment the physics yield in the LHCb dataset seems very attractive. However, the LHCb detector is limited by design in terms of data bandwidth — 1 MHz instead of the LHC bunch crossing frequency of 40 MHz — and physics yield for hadronic channels at the hardware trigger. Therefore, a Letter Of Intent [3], a Framework TDR [4] and a Trigger and Online TDR [5] document the plans for an upgraded detector which will enable LHCb to increase its physics yield in the decays with muon by a factor of 10, the yield for hadronic channels by a factor 20 and to collect $\sim$50 fb^{-1} at a leveled constant luminosity of 1–$2 \times 10^{33}\, \mathrm{cm}^{-2}\mathrm{s}^{-1}$. This corresponds to ten times the current design luminosity and increased complexity (pileup) of a factor 5.

2 The upgrade of the LHCb readout architecture

In order to remove the main design limitations of the current LHCb detector, the strategy for the upgrade of the LHCb experiment essentially consists of ultimately removing the first-level hardware trigger (L0 trigger) entirely, hence to run the detector fully trigger-less. By removing the L0 trigger, LHC events are recorded and transmitted from the Front-End electronics (FE) to the readout network at the full LHC bunch crossing rate of 40 MHz, resulting in a $\sim$40 Tb/s DAQ network. All events will therefore be available at the processing farm where a fully flexible software trigger will perform selection on events, with an overall output of about 20 kHz of events to disk. This will allow maximizing signal efficiencies at high event rates.

The direct consequences of this approach are that some of the LHCb sub-detectors will need to be completely redesigned to cope with an average luminosity of $2 \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and the whole LHCb detector will be equipped with completely new trigger-less FE electronics. In addition, the entire readout architecture must be redesigned in order to cope with the upgraded multi-Tb/s bandwidth and a full 40 MHz dataflow [6]. Figure 1 illustrates the upgraded LHCb readout architecture. It should be noted that although the final system will ultimately be fully trigger-less, a first-level hardware trigger based on the current L0 trigger will be maintained in software. This is commonly referred to as Software LLT and its main purpose is to allow a staging installation of the DAQ network, gradually increasing the readout rate from the current 1 MHz to the full and ultimate 40 MHz. This however will not change the rate of event recorded at the FE, which will run fully trigger-less regardless of the DAQ output rate.

In order to keep synchronicity across the readout system, to control the FE electronics and to distribute clock and synchronous information to the whole readout system, a centralized Timing and Fast Control system (TFC, highlighted in figure 1) has been envisaged, as an upgrade of the current TFC system [7]. The upgraded TFC system will then be interfaced to all elements in the readout architecture by heavily profiting from the bidirectional capability of optical links and FPGA transceivers and a high level of interconnectivity. In particular, the TFC system will heavily profit from the capabilities of the GigaBit Transceiver chipset (GBT) [8] currently being developed at CERN for its communication to the FE electronics. In addition, the TFC system will also be responsible to transmit slow control (ECS) information to the FE, by means of FPGA-based electronics cards interfaced to the global LHCb ECS.

3 Fast and slow control to FE via the TFC system

Figure 2 illustrates in detail the logical architecture of the upgraded TFC system. A pool of Readout Supervisors (commonly referred to as S-ODIN) centrally manage the readout of events, by generating synchronous and asynchronous commands, by distributing the LHC clock and by managing the dispatching of events. Each S-ODIN is associated with a sub-detector partition which effectively is a cluster of Readout Boards (TELL40) and Interface Boards (SOL40). While the TELL40s are dedicated to read out fragments of events from the FE and send them to the DAQ for software processing, the SOL40 boards are dedicated to distribute fast and slow control to the FE, by relaying timing information and clock onto the optical link to the FE, and by appending ECS information onto the same data frame. By profiting from the characteristics of the GBT chipset [8],
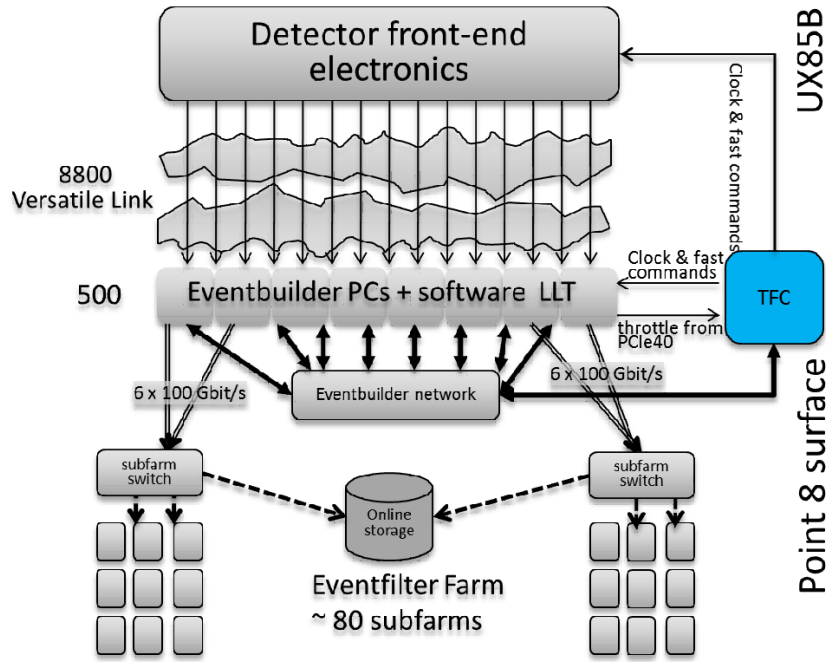
Figure 1. The upgraded LHCb readout architecture.

fast commands, clock and slow control are therefore transmitted on the same bidirectional optical link. This is a major novelty with respect to the current LHCb experiment where fast control and slow control are sent over different networks.

At the FE, the synchronous fast control information are decoded and fanned out by a GBT Master per FE board, also responsible to recover and distribute the clock in a deterministic way. The slow control information is relayed to the GBT-SCA chipsets via the GBT Master. The GBT-SCA chipset is capable of efficiently distribute ECS configuration data to the FE chips by means of a complete set of buses and interfaces, in a generic way [9]. Monitoring data is sent back on the uplink of the same optical link by following the return path, from the GBT-SCA to the Master GBT to the corresponding SOL40.

The hardware backbone of the entire readout architecture is a PCIe Gen3 electronics card hosted in a commercial PC. The same hardware is used for the TELL40, the SOL40 and the S-ODIN boards, only the different firmware changes the flavor of the board. The board will be equipped with up to 48 bidirectional optical link, an Altera Arria X FPGA and a 16x PCIe Gen3 bus interfaced to a multi-core PC.

Figure 3 shows schematically the implementation of the merging of fast and slow control information on the same optical link to the FE electronics [10] in the firmware at the SOL40 board.

A *TFC Relay and Alignment* block extracts at maximum 24 bits out of the full TFC word which was transmitted by S-ODIN encoding the fast commands, timing information and various resets. These 24 bits are then relayed onto the GBT link to be transmitted to the FE. The word is generated at 40 MHz and transmitted with constant latency. The TFC word from S-ODIN is used to reconstruct the clock locally in the FPGA to then be used to drive the logic in the firmware.
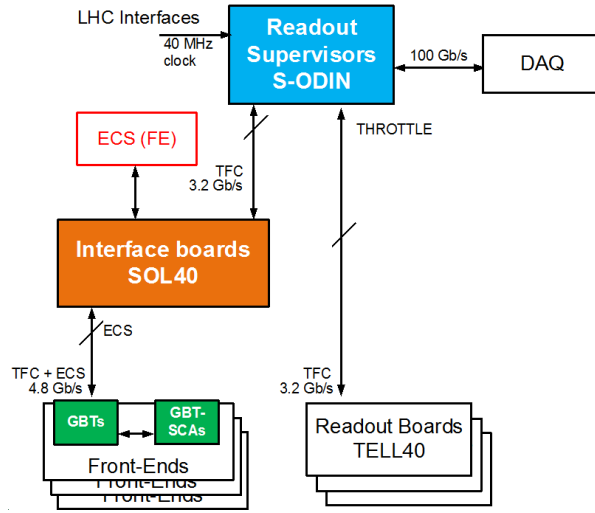
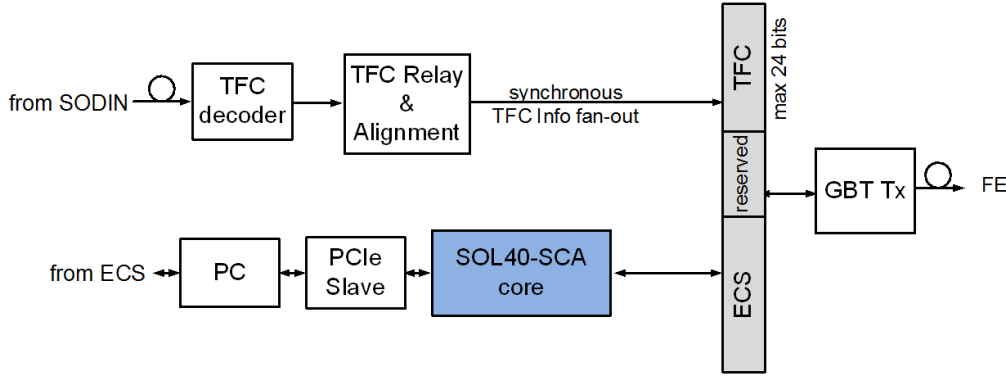**Figure 2**. Logical architecture of the upgraded TFC system.

Figure 3. Schematic view of the algorithm to merge TFC and ECS information on the GBT link towards the FE electronics in the SOL40 firmware.

Regarding the slow control part, LHCb has developed a firmware core, commonly referred to as *SOL40-SCA*, in order to generically drive each GBT-SCA chip at the FE, covering all of its functionality and protocols. Its location within the SOL40 firmware is highlighted in figure 3. This is achieved by developing the firmware in a completely configurable way, i.e. the chosen SCA protocol can be selected in real-time via commands issued by the LHCb ECS system [11] together with the configuration data. The destination of such data can be selected via a configurable mask. The core is designed to cover a full GBT link with up to 16 GBT-SCAs connected to it. It can then be replicated as many times as needed to cover all GBT links connected to a SOL40 boards. In total, the same firmware will allow driving generically the entire LHCb upgraded FE electronics over a total of ∼2500 duplex optical links and ∼90 SOL40 boards. It is technology independent, developed in HDL language, it does not make us of any technology specific element and it is completely agnostic of the content of the data field. It is basically a generic SCA driver via optical links for the BE electronics at the LHC.

4 The SOL40-SCA core

The core provides a way to control with high parallelism and flexibility many FE chips via the GBT-SCA interfaces through GBT links. Its main functionalities can be listed as follows:

- Provide a generic hardware interface (FPGA) between the ECS system and the FE electronics

- Build and encode/decode GBT-SCA compliant packets

- Serialize and de-serialize command packets in the command word sent to the FE electronics according to GBT-SCA specifications

- Support for all GBT-SCA protocols (SPI, I^{2}C, JTAG, GPIO and ADC+DAC)

- Support for all GBT-SCA commands and channels

- Support for many GBT-SCAs per GBT link and many GBT links per FPGA

- Possibility of re-transmission of packets and transmission monitoring

- Modularity, i.e. components can be removed if not needed

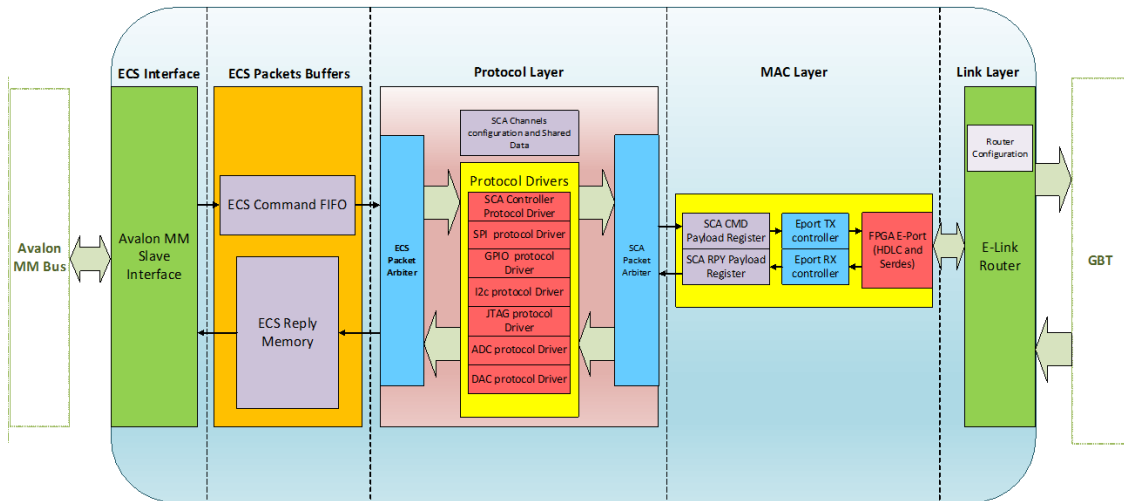- Robustness, reliability, programmability, flexibility.

Figure 4. Top view of the SOL40-SCA core architecture.

The core is essentially composed of a series of layers as it is illustrates in figure 4. Their main roles are to:

- store the ECS configuration packets and decode them as commands and viceversa in the *ECS Interface* and *ECS Packets Buffers Layers*

- build the corresponding GBT-SCA packets with the selected protocol in the *Protocol Layer*

- encode it in the specified communication protocol (HDLC [12]) in the *MAC Layer*

- serialize and route the packets to the selected GBT-SCA connected to a GBT at the FE in the *Link Layer*.

In practice, the ECS generates a command which is transmitted to the FPGA via the PCIe bus. This commands contains an *extended addressing scheme* to tell the core where and how to route the configuration packet and a *command code scheme* which tells the core what actions to perform (i.e. read/write or wait for response/do not wait). In addition, it may contain the configuration data to be sent to the FE in case of a write operation. In the FPGA, the command is stored in a buffer in order to be picked up by the Protocol Layer when not busy. The ECS command is then decoded and the SCA specific protocol packets are built accordingly. The information about which protocol to be built is in the ECS command and it is completely generic, that is the core is able to build run-time any SCA packet simply based on the content of the command. Finally, the packet is encapsulated in the HDLC protocol to then be routed to the corresponding bit field in the GBT word to be sent through the optical link to the corresponding Master GBT at the FE. The corresponding bit field is selected based on the connections at the FE. In order to be as generic as possible, this is also a configurable parameter so that the core can be used with any FE configuration. The core also features the possibility of packet retransmission in case a particular transaction failed.

4.1 ECS Interface Layer

In order to access the PC through a PCIe bus, the SOL40 board internally uses an Altera Avalon MM bus, which is mapped onto one of the PCIe BARs (bar 0). Hence, an Avalon MM Slave Interface is used in the ECS Interface Layer to perform read and write operations to and from the control PC.
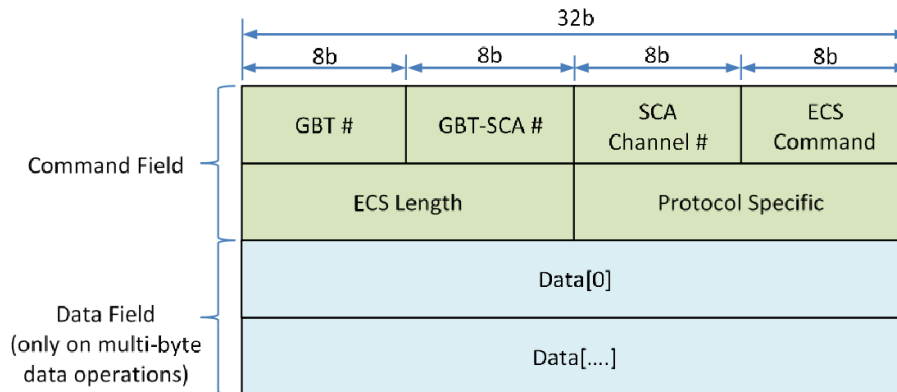

Figure 5. Current ECS command format as transmitted through the Avalon MM Bus.

Figure 5 illustrates the structure of the generic ECS command which is built by the control system via dedicated graphical interfaces and scripts. This command is transmitted to the firmware core and contains all the relevant information so that the core can generically and flexibly build GBT-SCA compliant packets. In the first field, an extended addressing scheme is implemented: the addresses of the GBT link, of the GBT-SCA and its channels are included. In addition, there is an ECS Command field dedicated to specific commands from the ECS (for ex, Read or Write). In the second field, the length of the ECS command in number of bytes for frame boundary definitions and a protocol specific field are inserted. This is followed by Data packets if a write operation is requested. All fields are 32-bit aligned so that the ECS system can transmit a full command as a 32-bit words table.

The same ECS command is generated by the firmware core in response to a polling by the ECS.

4.2 ECS packets buffers layer

The ECS commands are then stored in a FIFO. This is because the clock frequency used by the Avalon MM Slave Interface is 40 MHz with a fixed data size of 32 bits. However, the output bandwidth per GBT-SCA is 80 Mb/s, i.e. two bits every 40 MHz. Considering that an ECS command can span over various 32 bit words, the ECS command must be buffered and stored in order to allow building the corresponding SCA packets and transmit them via the corresponding pair of bits through the GBT link.

An ECS Command FIFO is dedicated to store ECS command packets. A single FIFO structure per GBT link was chosen as the ECS data stream comes as a single thread of many commands. However, they are dispatched asynchronously to their associated channels and GBT-SCAs. It is therefore a simple way to create back-pressure control and avoid congestion while building packets and transmitting them. This also means that the ECS can send a table of commands in one continuous write operations and the firmware will take care of reading and decoding the commands on a per-channel basis.

An ECS Reply Memory is dedicated to store the replies to a specific ECS command. It is designed as a RAM structure rather than a FIFO so that the software can access the memory following a mapping of the extended addressing scheme. The ECS can therefore poll a spcific reply based on the previously generated command.

4.3 Protocol Layer

The GBT-SCA chipset supports a large variety of buses that can be interfaced to FE chips. In the Protocol Layer, each ECS command is transformed into an SCA command where the right protocol for a particular SCA channel is built. This allows the user to flexibly select whichever bus they need to drive by simply indicating it in the command code scheme as shown in figure 5. In this way the same firmware can be used for all possible combinations at the FE without being dependent on the sub-detectors' choices at the FE.

In addition, an additional important feature is that the Protocol Layer keeps information regarding the SCA command generated for packet retransmission, manages the reading of ECS commands from the ECS command FIFO based on a busy state or a non-functional state (i.e. when the wrong SCA was selected for example) and when the packet is ready, it transmits it to the MAC Layer. This is managed by two arbiter modules, one dedicated to arbitrate the reading/writing of ECS commands and one dedicated to arbitrate the transmission/reception of SCA commands. Figure 6 shows an example of an operation on the protocol layer, where the ECS PC sends I^{2}C write command to a certain I^{2}C device on the FE.

4.4 MAC Layer

The MAC Layer is mostly responsible to encapsulate the SCA payload packet into the HDLC protocol [12] and to serialize it in pair of bits, to be then slotted in on each GBT word on the optical link. In addition, it de-serializes the data stream and extract the payload when a reply is received. It also features link reset, connection and test operations and error detection capabilities.

The heart of the MAC Layer is a block called FPGA E-Port. This is based on the original E-Port IP Core [9] but with some key differences. The block is made in a device independent way,
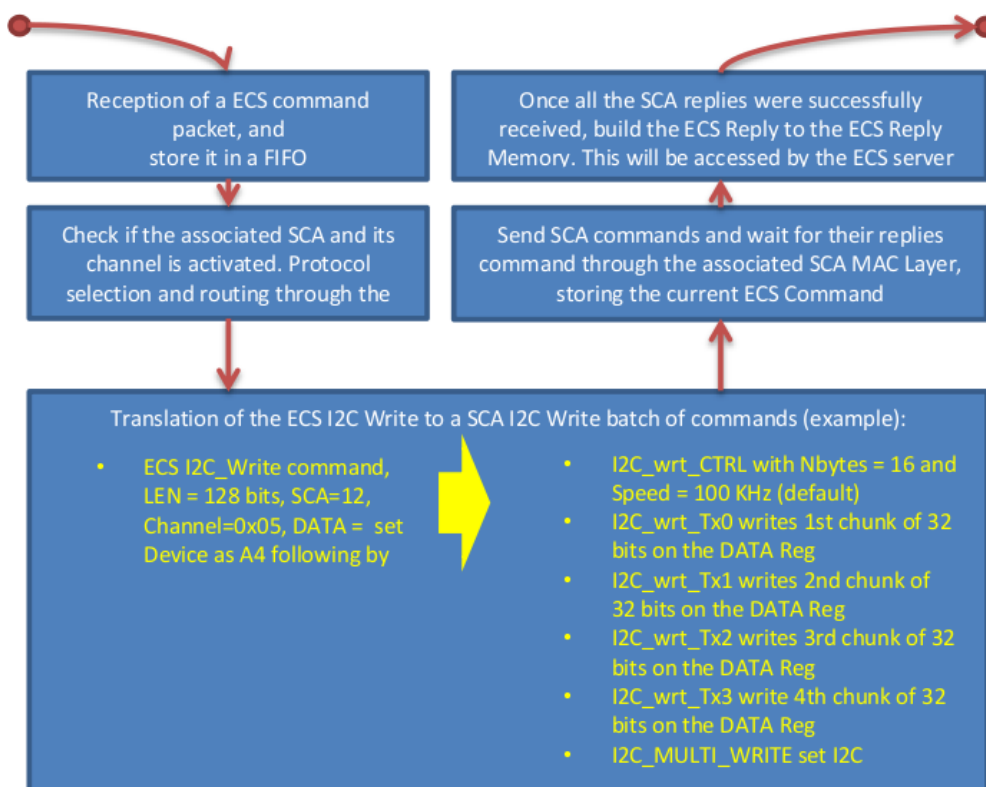
Figure 6. Example of I^{2}C write operation as managed in the Protocol Layer.

it does not a backup connection and it is designed without Triple Modular Redundancy as it is to be used in a safe radiation environment.

An additional feature is the possibility of retransmitting a packet if the transmission of a previous command failed. This is done in the MAC Layer as the full protocol included the communication protocol is already done at that stage. A programmable expiration time is used to wait for the response from the corresponding SCA and programmable bit transmitted within the ECS command is used to tell the core whether to re-transmit a packet or instead simply signal a flag to the ECS without re-transmitting the packet. This can be done on a per-command basis at run-time.

Another additional feature is the possibility of waiting for a response from the corresponding GBT-SCA. Another specific bit in the ECS command is used to tell the core whether to wait for the GBT-SCA to acknowledge the response or simply send the packet a programmable time after, without caring about the acknowledge. It is however necessary to wait a minimum time before transmitting the following packet because the GBT-SCA must receive the previous packet in its entirety. This can be done on a per-command basis at run-time as well.

4.5 Link Layer

Finally, the last layer between the core and the GBT link is the Link Layer. It is a simple layer whose only purpose is to provide a generic and programmable logical routing so that the SCA packet can reach the right GBT-SCA over the corresponding GBT link. This is done by an E-Link router whose configuration is a matrix loaded in a configurable register, changeable at runtime.

5   Conclusion

Within its upgrade, the LHCb experiment has developed a generic firmware core to drive any GBT-SCA within the upgrade of the experiment. This is achieved by implement an HDL-based code, capable of driving any protocol of any GBT-SCA over any GBT link, programmable at run-time. The core is so generic that can be used in any FE environment featuring the presence of the GBT chipset.

The firmware core will be ready by the beginning of 2015 in time for allowing its usage by sub-detectors in test-benches, test-beams and in order to commission the FE electronics for the upgrade of the LHCb. A heavy testing campaign together with the very first GBT-SCA chips will be performed in order to test robustness, reliability and compatibility issues.

Acknowledgments

References

[1] LHCb collaboration, *The LHCb Detector at the LHC*, 2008 *JINST* **3** S08005.

[2] R. Jacobsson, *Performance of the LHCb Detector during the LHCb Proton Runs 2010-2012*, *IEEE Nucl. Sci. Symp. Med. Imag. Conf.* **2012** (2012) 1479.

[3] LHCb collaboration, *Letter of Intent for the LHCb Upgrade*, CERN-LHCC-2011-001 (2011).

[4] LHCb collaboration, *Framework TDR for the LHCb Upgrade*, CERN-LHCC-2012-007 (2012).

[5] LHCb collaboration, *LHCb Trigger and Online Upgrade Technical Design Report*, CERN-LHCC-2014-016 (2014).

[6] F. Alessio et al., *Trigger-less readout architecture for the upgrade of the LHCb experiment at CERN*, 2013 *JINST* **8** C12019.

[7] F. Alessio and R. Jacobsson, *Timing and Fast Control for the Upgraded Readout Architecture of the LHCb experiment at CERN*, *IEEE Trans. Nucl. Sci.* **60** (2013) 3438.

[8] P. Moreira et al., *The GBT Ser-Des ASIC prototype*, 2010 *JINST* **5** C11022.

[9] A. Caratelli et al., *The GBT-SCA, a Radiation Tolerant ASIC for Detector Control and Monitoring Applications in HEP Experiments*, in proceedings of *Topical Workshop on Electronics for Particle Physics 2014*, Aix en Provence, France, 22–26 September 2014.

[10] F. Alessio and R. Jacobsson, *A New Readout Control system for the LHCb upgrade at CERN*, 2012 *JINST* **7** C11010.

[11] C. Gaspar et al., *The LHCb Experiment Control System: on the path to full automation*, at *13th International Conference on Accelerator and Large Experimental Physics Control Systems*, Grenoble France (2011), pg. 20.

[12] International Standards Organization, *Telecommunications and information exchange between systems — HDLC procedures*, ISO/IEC 13239:2002.