PROCEEDINGS OF SCIENCE

The New Version of the LHCb SOL40-SCA Core to Drive Front-End GBT-SCAs for the LHCb Upgrade

João Barbosa[*]
EP-LBC, CERN
E-mail: joao.barbosa@cern.ch

Federico Alessio
EP-LBC, CERN
E-mail: federico.alessio@cern.ch

Clara Gaspar
EP-LBC, CERN
E-mail: clara.gaspar@cern.ch

The LHCb experiment is currently engaged in an upgrade effort that will implement a triggerless 40 MHz readout system. The upgraded Front-End Electronics profit from the GBT chipset functionalities and bidirectional optical fibers for readout, control and synchronization. This paper describes the new version of the firmware core that transmits slow control information from the Control System to thousands of Front-End chips and discusses the implementation that expedites and makes the operation more versatile. The detailed architecture, original interaction with the software control system and integration within the LHCb upgraded architecture are described.

Topical Workshop on Electronics for Particle Physics
11 - 14 September 2017
Santa Cruz, California

[*]Speaker.

1. Introduction

The LHCb experiment is one of four major experiments operating at the Large Hadron Collider and it consists of a high precision detector that focuses on the detection of decays of particles containing B mesons. LHCb is currently involved in an Upgrade [1] effort that will happen between 2019 and 2020. During this upgrade, most of the on-detector and back-end electronics will be replaced. In order to relay the event information from all bunch crossings to the Software Trigger, the readout electronics will have to cope with the full 40MHz bunch crossing rate without performing first-level event triggering. In view of this requirement, the experiment is using the GBT chipset [2] and its related components developed by the EP-ESE group at CERN, for data readout as well as for fast control and slow control tasks. Furthermore, to aid the slow control tasks, the GBT-SCA ASIC [3] is widely used to configure and monitor the Front-End electronics (FE) via a set of protocols such as JTAG, I2C and SPI. In LHCb, all the chips located the FE are controlled via FPGA-based electronics boards - interfaced to the software control system via a PCIe bus - with bidirectional optical links through the optical capabilities of the GBT chipset. These boards are commonly referred to as PCIe40 [4], but also as one of the three logic flavours provided for the Upgrade, namely: SODIN, SOL40 and TELL40. While the TELL40 is responsible for the readout of the FE event fragments, the SODIN plays the role of readout supervisor, distributing the LHC clock, generating synchronous and asynchronous commands to the full readout system and dispatching event data to the event building stage. The SOL40 is an Interface Board that is responsible for interfacing both TFC (Timing and Fast Control)[5] and ECS (Experiment Control System)[6] with the FE boards. It relays timing and clock information sent by SODIN into the optical link to the FE, while also using the same data frame to give the ECS capabilities to both control and monitor the FE over the same link. It is estimated that the LHCb experiment will contain around 2500 Master GBTs and GBT-SCAs and around 10000 FE chips. To be able to drive the numerous GBT-SCA chips in the FE, a specific firmware core, hosted in the PCIe40, commonly referred to as SOL40-SCA, was initially developed. It consists of a technology agnostic firmware VHDL core, developed in a generic way to support all buses and protocols of the GBT-SCA chips. This document discusses the tests done in the current architecture and summarily explains the motivations for an upgraded design of the SOL40-SCA.

2. The SOL40-SCA core

The control and monitoring of the FEs is done by driving the GBTx and GBT-SCA chips, located in the Front End boards, through the optical links in the SOL40 board. Within this board's firmware, the SOL40-SCA core is the one responsible for sending ECS commands to the GBT-SCA chip. The GBT-SCA in turn, will interface directly the devices to be controlled and monitored through a set of serial protocols (I2C, JTAG, SPI), general purpose interfaces (GPIO) and ADC/DAC channels. The location of the SOL40-SCA core in the system is shown in Figure 1.

Each one of the SOL40 cards can connect to 48 GBT front-ends, each one using a different bidirectional optical link. In turn, these GBT chips are able to connect to up to 32 different GBT-SCA chips. The number of actual FE devices to be controlled and monitored by each GBT-SCA, and also the number of GBT-SCAs controlled by each GBTx are highly dependable on each

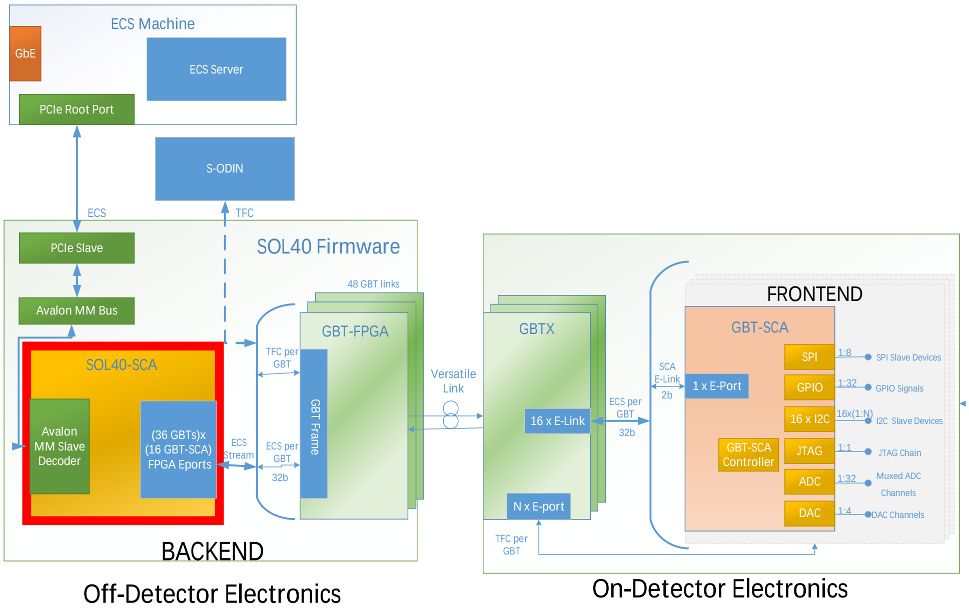

Figure 1: SOL40-SCA core in the Control system.

Front-End Board architecture. The main requirements for the SOL40-SCA core are that it must accommodate this potentially high number of devices while also simplifying the job of the software running on the PCIe40 host server, allowing it to perform complex operations in each GBT-SCA (eg. configuring an FPGA through JTAG) as fluidly as possible.

3. Current architecture

The architecture of the SOL40-SCA discussed in this section is currently distributed to users for their prototyping and test-beams. This sections briefly describes it and explains the tests that were conducted with it.

The schematic of the architecture can be found in [7]. This version of the design supports all the features the GBT-SCA has to offer, but it also implements a minimal protocol acceleration, which consists of having one ECS command (a command sent by the control software) to be translated into several GBT-SCA commands so that the software does not have to handle every single SCA reply. Furthermore it supports packet retransmission and it is instantiated once per GBT link (fiber).

The core is composed of 4 layers, each with the following functions:

- The Interface Layer implements registers for the control system to perform operations on the core through the Avalon Bus.
- The Buffer Layer holds information of the commands to be sent to the GBT-SCAs and also of their replies until the Control System reads them through the Interface Layer.
- The Protocol Layer is responsible for reading the commands stored in the Buffer Layer, written in a generic format (ECS format), transforming it into one or several SCA commands, routing them to the chosen SCA through a specific Mac Layer.

- The Mac Layer is responsible for encapsulating the SCA payload into the HDLC protocol [8] and send these frames through the GBT link.

The Control System has the choice of sending ECS commands that translate to only one SCA command or sending ECS commands that unroll into a maximum number of 8 SCA commands. This allows for the software to send only one command to the SOL40-SCA if it wants to perform an operation such as writing 128 bits to one of the serial protocol channels (eg. SPI) in a given SCA. This is shown to expedite lengthy operations, such as the configuration of an FPGA through the JTAG or SPI channel of the SCA. Table 1 shows the results of configuring a Kintex 7 FPGA through the JTAG channel in the case where the ECS command to SCA command translation ratio is 1:1 another case in which it is 1:5. These tests were conducted in two different back-end boards and host servers and they clearly show there is some speed to gain out of implementing the capability to handle arbitrarily large ECS commands, instead of limiting them to 128 bit transactions, thus avoiding dead time of GBT-SCA utilization between ECS commands.

Host PC CPU	Intel(R) Atom(TM) CPU E640 @ 1.00GHz	Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz (target system)
On Board FPGA	Stratix V 5SGXEA7N2F45C3N GX	Arria10 10AX115S4F45E3SG
CPU-FPGA Interface	PCI Express Gen 1	PCI Express Gen 3
JTAG speed (no encoding)	18 KB/s	267 KB/s
JTAG speed (5x encoding)	87 KB/s	405 KB/s
Speed-up provided by encoding	4.8x	1.5x

Table 1: Tests performed configuring a Kintex 7 through the SOL40-SCA and a GBT-SCA.

From Table 2 one can estimate, assuming linear growth of resources usage, that instantiating the SOL40-SCA for 48 links would take up 110% of the Logic Modules available in the Arria 10 FPGA mounted on the PCIe40. This calls for a redesign of the core so that it either consumes less resources per instance or is modified to support more than one GBT link per instance. The new solution is explored in the next section.

Entity	ALMs out of 427200	Dedicated Logic registers	Memory block bits
Top	10242.5 (2.3 %)	14255 (.8 %)	122856 (.2 %)
Interface Layer	284.7	554	0
Buffer Layer	991.5	2001	105448
Protocol Layer	3484.7	3108	0
Mac Layer	5633.2	9325	18496

Table 2: Resources occupied by a SOL40-SCA core that supports 17 GBT-SCAs in one GBT link.

4. Upgraded architecture

The new design has as its main aim to increase the number of links that an instance of the SOL40-SCA supports while still adding functionality to expedite lengthy ECS commands and minimizing the deadtime between them. The new design will replace the Mac Layer with the GBT-SC core provided by the EP-ESE team at CERN to do the serialization and HDLC protocol handling [9]. Other extensive changes will be concentrated in the Buffer Layer and the Protocol Layer and are explained in the following sections. There are two driving requirements of this design:

- Increase the speed of the serial bus channels in the GBT-SCA by implementing arbitrarily large ECS commands by the use of large memories.
- Through the use of the same memories, allow for a better fluidity of commands by not requiring the control software to wait for every reply before sending a new command. These are instead queued in the memories waiting to be handled by the protocol layer.

The schematic of the upgraded architecture is shown in Figure 2 and each layer is explained in the following sections.

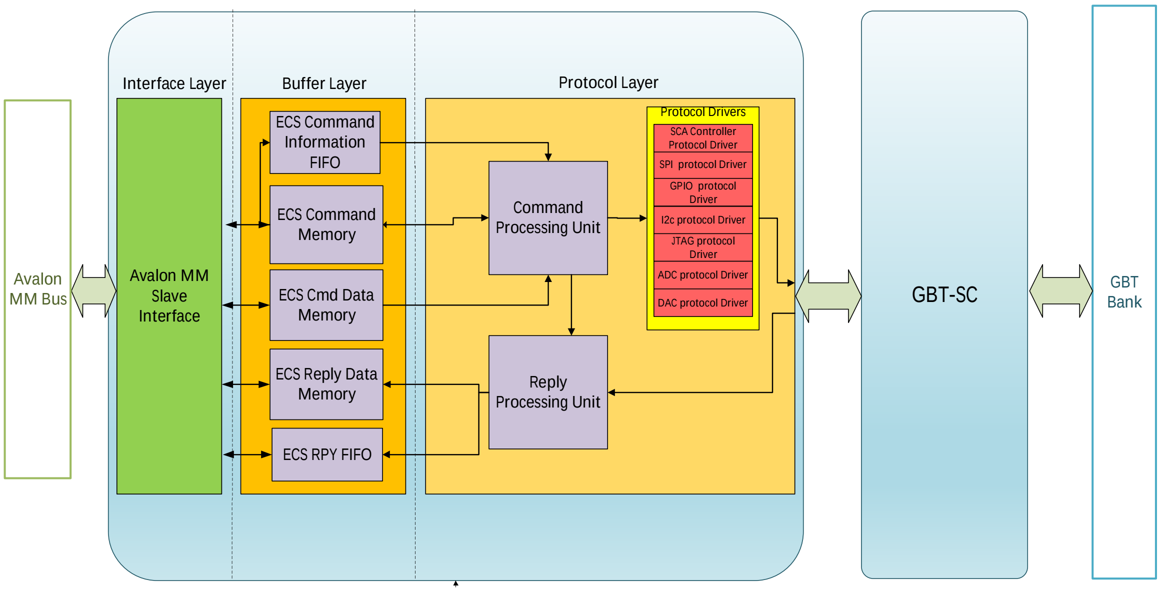

Figure 2: Schematics of the upgraded version of SOL40-SCA.

4.1 Buffer Layer

The buffer layer was redesigned to be able to support much larger commands and a more continuous burst of them by the means of using separate memories for command settings and command data. In this manner the data to be sent through an SCA channel is not limited to 128 bits per ECS command and can be as large as the data memory used. When an ECS command is written in the command setting memory, it will contain a pointer to the data memory as well as how many bits should be written or read. Reply data is however written in a separate memory by the Protocol Layer. A unique identifier per ECS command is queued both in the command FIFO and the reply FIFO. This identifier contains a pointer to the data memory so that the control software knows where to read replies from and the protocol layer knows where to get the data to be sent to the GBT-SCA.

4.2 Protocol Layer

The protocol layer is divided into three main components: a command processing unit, a reply processing unit and a set of protocol drivers. The command processing unit consists of a set of finite state machines that will manage the access to the memories and FIFO and also the actual sending of commands to the GBT-SC. It will in turn make use of the protocol drivers, which consist of state machines that will dictate which set of SCA commands are used to carry out a certain ECS command. Lastly the Reply Processing Unit will simply handle the replies coming from the GBT-SC layer, writing the incoming data into the Reply Data Memory and the summary of an ECS command execution into the Reply FIFO, together with a unique identification of the command and a pointer to the data memory. This layer is able drive the GBT-SCA to output 500KB/s of serial bus data.

5. Conclusion

To drive the GBT-SCAs used in the upgraded electronics, LHCb uses a custom-made firmware core denominated SOL40-SCA. The current architecture in use is able to drive all the protocols the GBT-SCA provides but is currently occupying too many resources in the target FPGA Arria 10. Also its performance is highly dependable on the software performance, which makes it slower and more unpredictable. The new architecture intends to take a lot of protocol building away from the software and implement it in firmware, while also allowing for more complicated sets of commands to be handled by the firmware with a single software instruction. This allows for a speed of 500 KB/s in the serial protocols SPI, JTAG and I2C for an ECS command. Moreover, the extended memory space and the decoupling of the command and reply chain will allow the software to handle commands and replies separately, so new commands are sent before the previous replies arrive.

References

[1] Liu, G., and Neufeld, N. (2014), DAQ architecture for the LHCb upgrade. *Journal of Physics: Conference Series. Vol. 513. No. 1. IOP Publishing*.

[2] Moreira, P. (2013), The radiation hard GBTX link interface chip. *PH-ESE Electronics Seminars, November. Vol. 26*.

[3] Caratelli, A., et al. (2015), The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments. *Journal of Instrumentation 10.03: C03034*.

[4] Bellato, M., et al. (2014), A PCIe Gen3 based readout for the LHCb upgrade. *Journal of Physics: Conference Series. Vol. 513. No. 1. IOP Publishing*.

[5] Alessio, F., et al. (2016), Timing and Readout Control in the LHCb Upgraded Readout System. *No. LHCb-PROC-2016-015*.

[6] Alessio, F., et al. (2010), The LHCb Run Control System, *Real Time Conference (RT), 2010 17th IEEE-NPSS, Lisbon, 2010, pp. 1-6*.

[7] Alessio, F., et al. (2015), A generic firmware core to drive the Front-End GBT-SCAs for the LHCb upgrade. *Journal of Instrumentation 10.02: C02013*.

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

[9] Mendez, J., and Baron, S. (2017), New slow control module for GBT based systems and status update of the GBT-FPGA project, *TWEPP17, this conference*.