Status update and future plans for the GBT-FPGA project at CERN

J. Mendez, S. Baron, M. Barros Marin

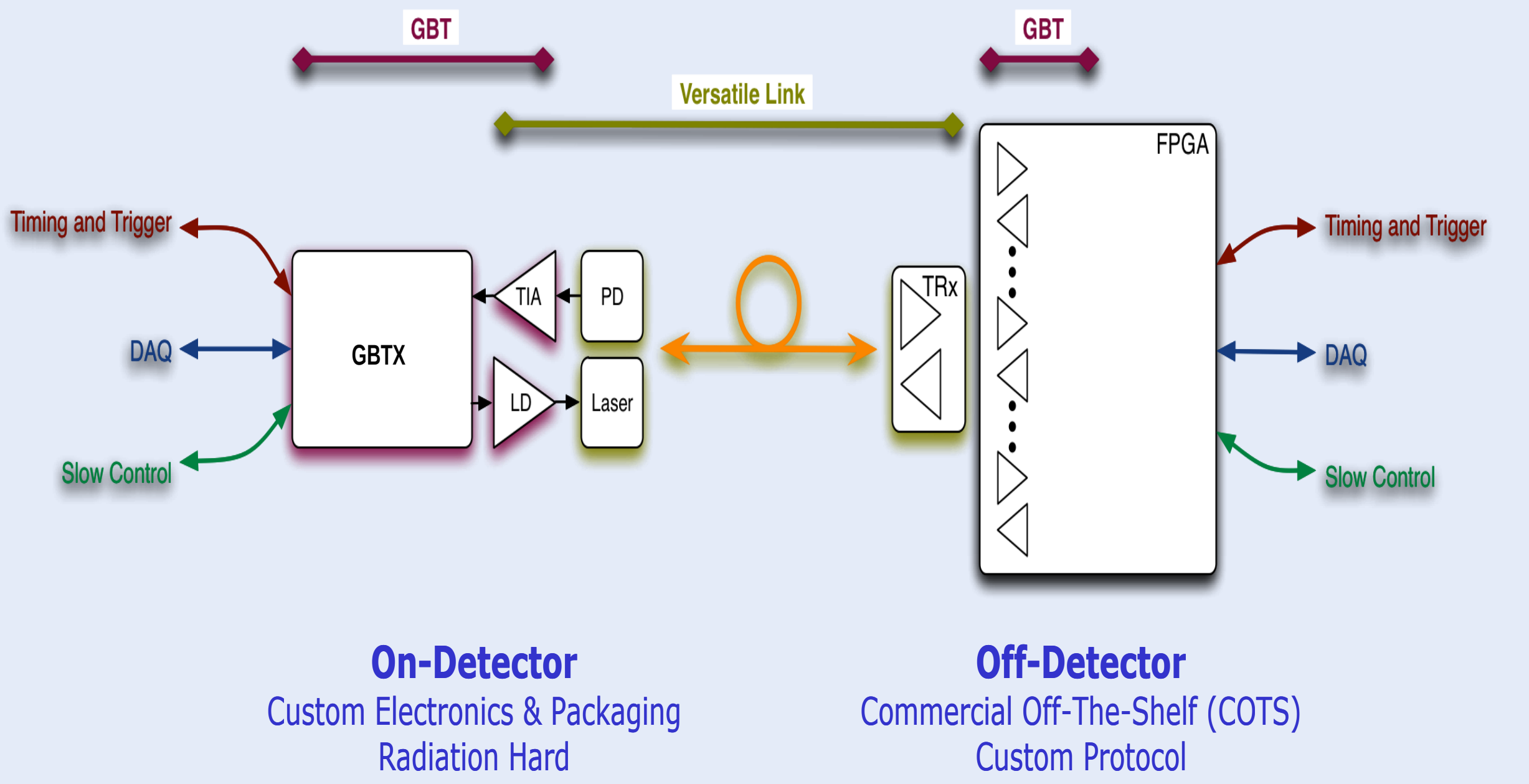
EP-ESE-BE Section, CERN, 1211 Geneva, Switzerland

Abstract

Following a study made in 2009 to implement the GBT SERDES in FPGAs, the GBT-FPGA project was launched. Its main goal is to provide a unified GBT core for multiple users and different FPGAs in order to ease communication with the GBTx. The VHDL-based IP is used for the upgrade of the LHC's experiments DAQ and Timing Systems as well as the emulation of GBTx chip for test purposes. It provides two types of implementations for the transmitter and the receiver ("Standard" and "Latency-optimized") and two encoding schemes supported by the GBTx serializer/deserializer ASIC ("GBT-Frame" (Reed-Salomon) and "Wide bus"). The package also includes some example designs for the most common FPGA development kits.

Introduction to the GBT-FPGA

The GBT-FPGA is used to communicate with the on-detector system. The diagram shown in this section introduces a typical system featuring the "Rad-Hard Optical Link for Experiments". The GBT-FPGA is implemented into the Back-end FPGA-based board (off-detector side) to transmit timing, trigger and Control (TTC) as well as the slow control to the Front-Ends. It can also receive and forward the data from the detector to the central data acquisition system (DAQ) at 4.8 Gbps. Trigger related electronic systems in High Energy Physics (HEP) experiments, such as Timing Trigger and Control (TTC), require a fixed, low and deterministic latency in the transmission of the clock and data to ensure correct event building. On the other hand, other electronic systems that are not time critical, such as Data Acquisition (DAQ), do not need to comply with this requirement. The GBT-FPGA project provides two types of implementation for the transmitter and the receiver: the "Standard" version, targeted for non-time critical applications and the "Latency-Optimized" version, ensuring a fixed, low and deterministic latency of the clock and data (at the cost of a more complex implementation). The GBT-FPGA supports two encoding schemes supported by the GBTx: GBT based on Reed-Salomon and Wide-bus. The Reed-Salomon encoding allows correcting bursts of bit errors caused by a Single Event Upset (SEU). For the the wide-bus, the Forward Error Correction (FEC) is replaced by user data that implies the removing of the correction system. This encoding scheme cannot be used for the TTC.

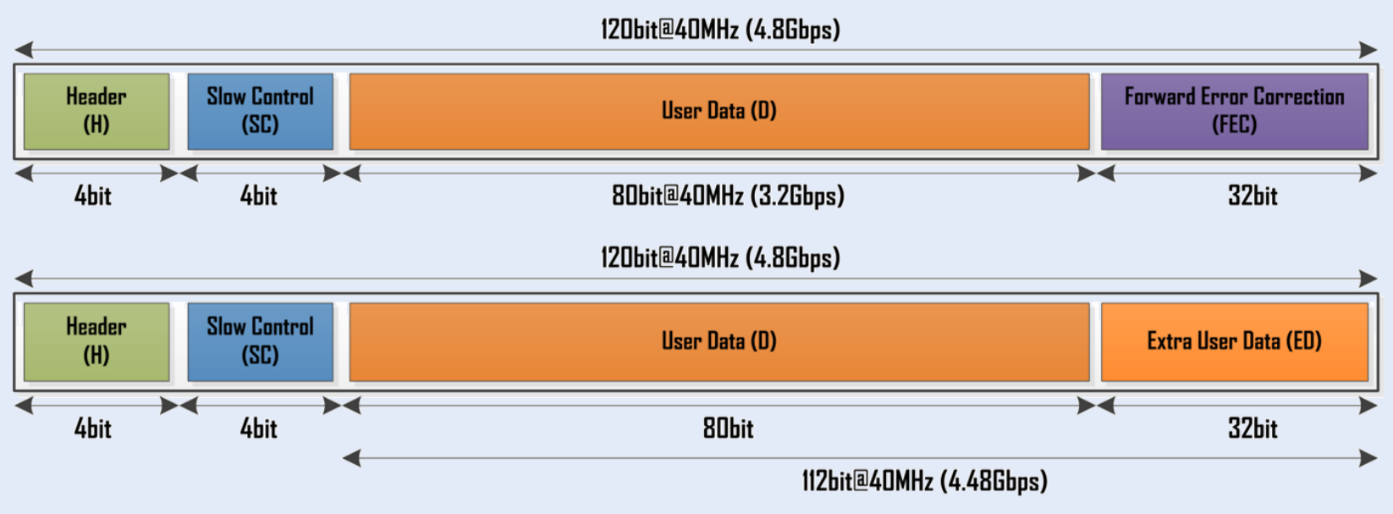


On-Detector
Custom Electronics & Packaging
Radiation Hard

Off-Detector
Commercial Off-The-Shelf (COTS)
Custom Protocol

	Standard	Latency-Optimized
Latency	Non Fixed, Higher & Non Deterministic	Fixed, Low & Deterministic
Logic Resources Utilization	Low	Low
Clocking Resources Utilization	Low	High
Clock Domain Crossing	Don't Care	Critical
Implementation	Simpler	Complex

Standard vs. latency optimized



GBT Frame vs. WideBus

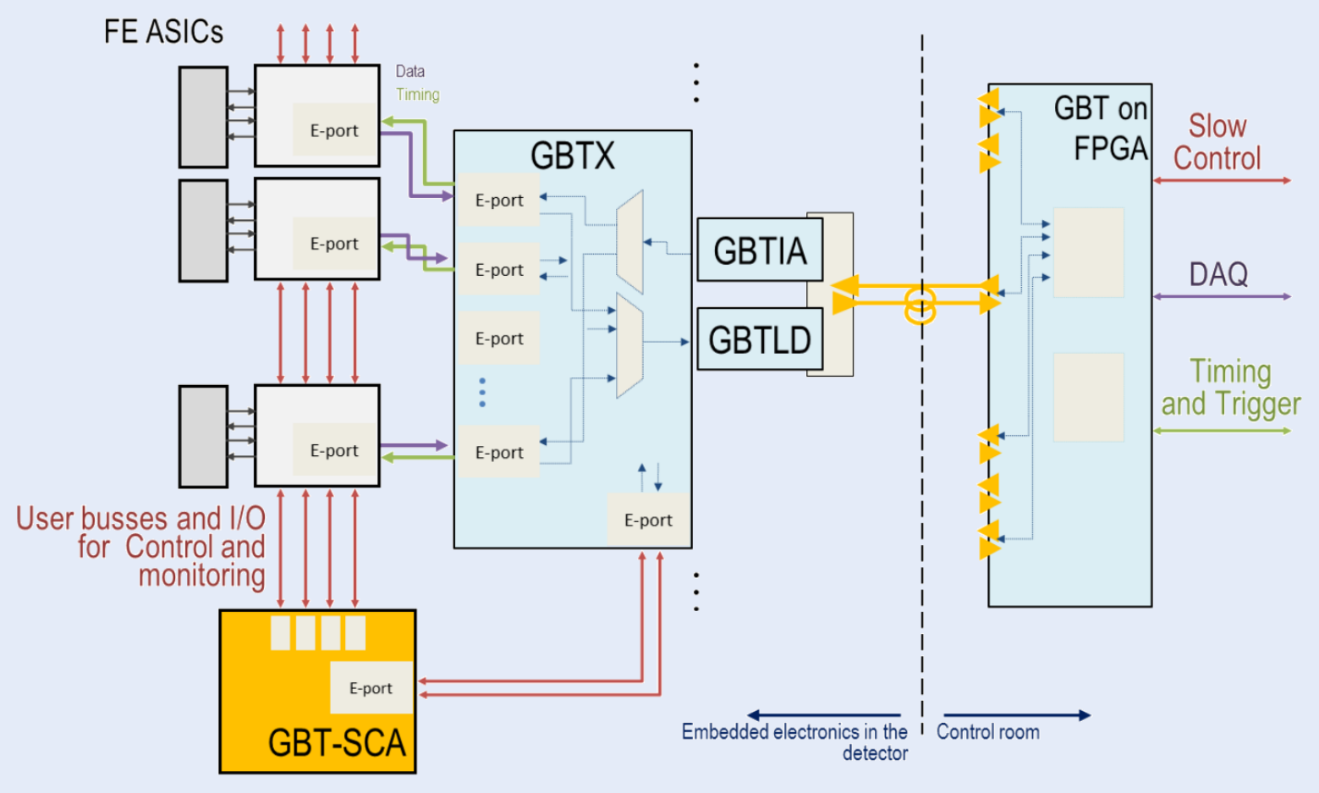
Introduction to the GBTx / SCA slow control

The communication between the Front-Ends and the control room includes a slow-control system carrying bi-directional traffic using a unique link shared with the user data. The slow-control is used to configure or read status of the GBTx (Internal Control, IC) or to send different commands to an external component named SCA (External Control, EC). This chip provides different interfaces to manage the front-end electronic.

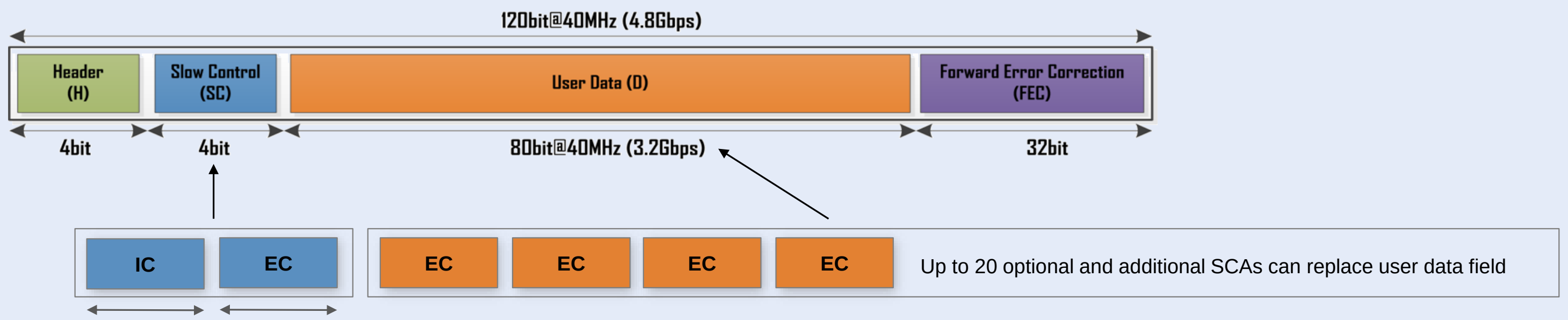
List of SCA interfaces:

- 16 I2Cs
- 1 SPI master (8 slave select lines)
- 1 JTAG master
- 32 GPIOs
- 32 ADC inputs (including 1 temperature sensor)
- 4 Analog outputs

The SCA is in charge of the communication with the peripheral chips connected to its ports. Its control is performed using 2 encoded bits per frame at 40 MHz resulting on a data bandwidth of 80 Mbps. The same mechanism, as shown below, is used for the GBTx configuration.



GBT-SCA connection



Up to 20 optional and additional SCAs can replace user data field

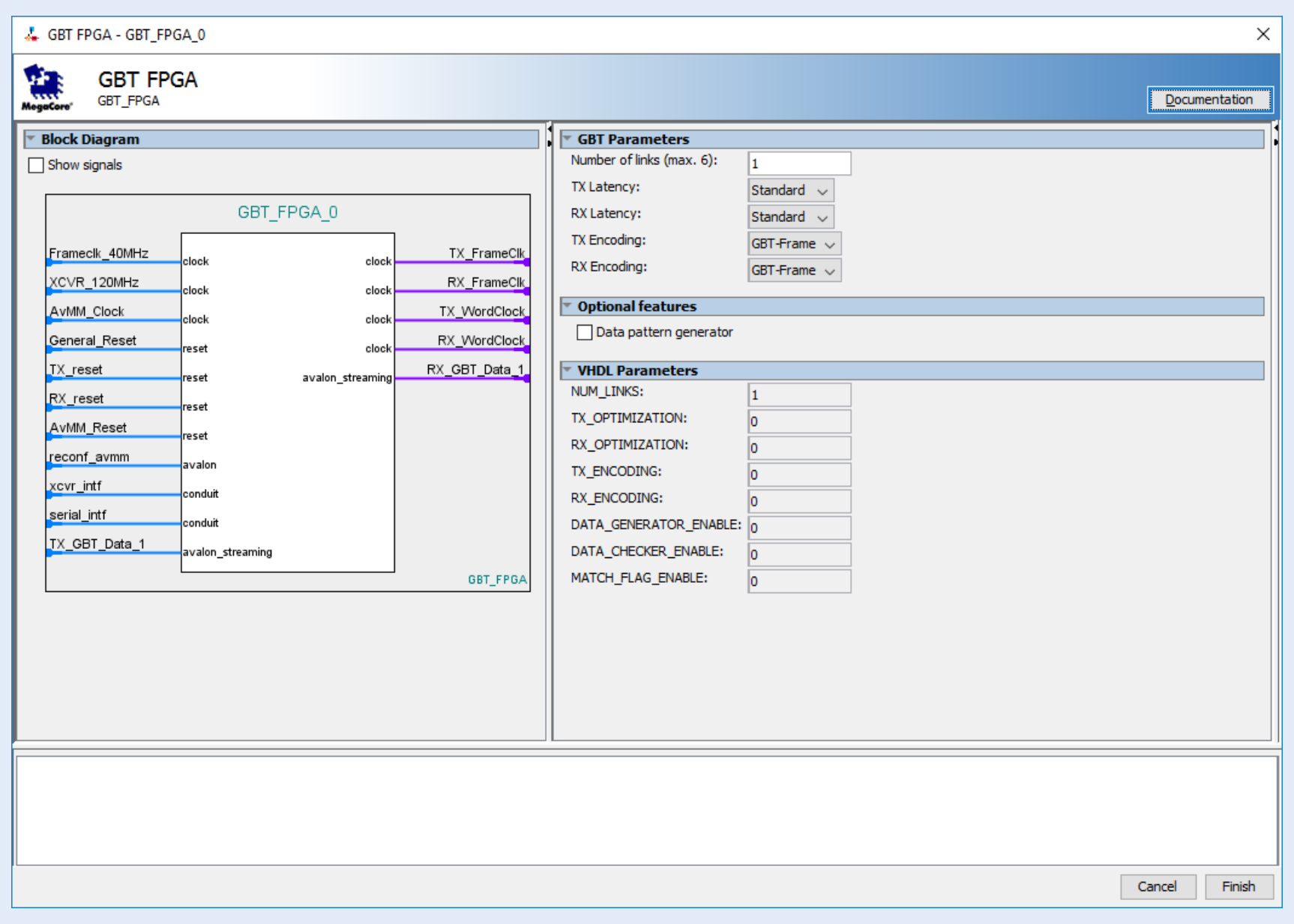
Warning: 21 SCAs implies using all e-ports for slow control communication.

GBT-FPGA status

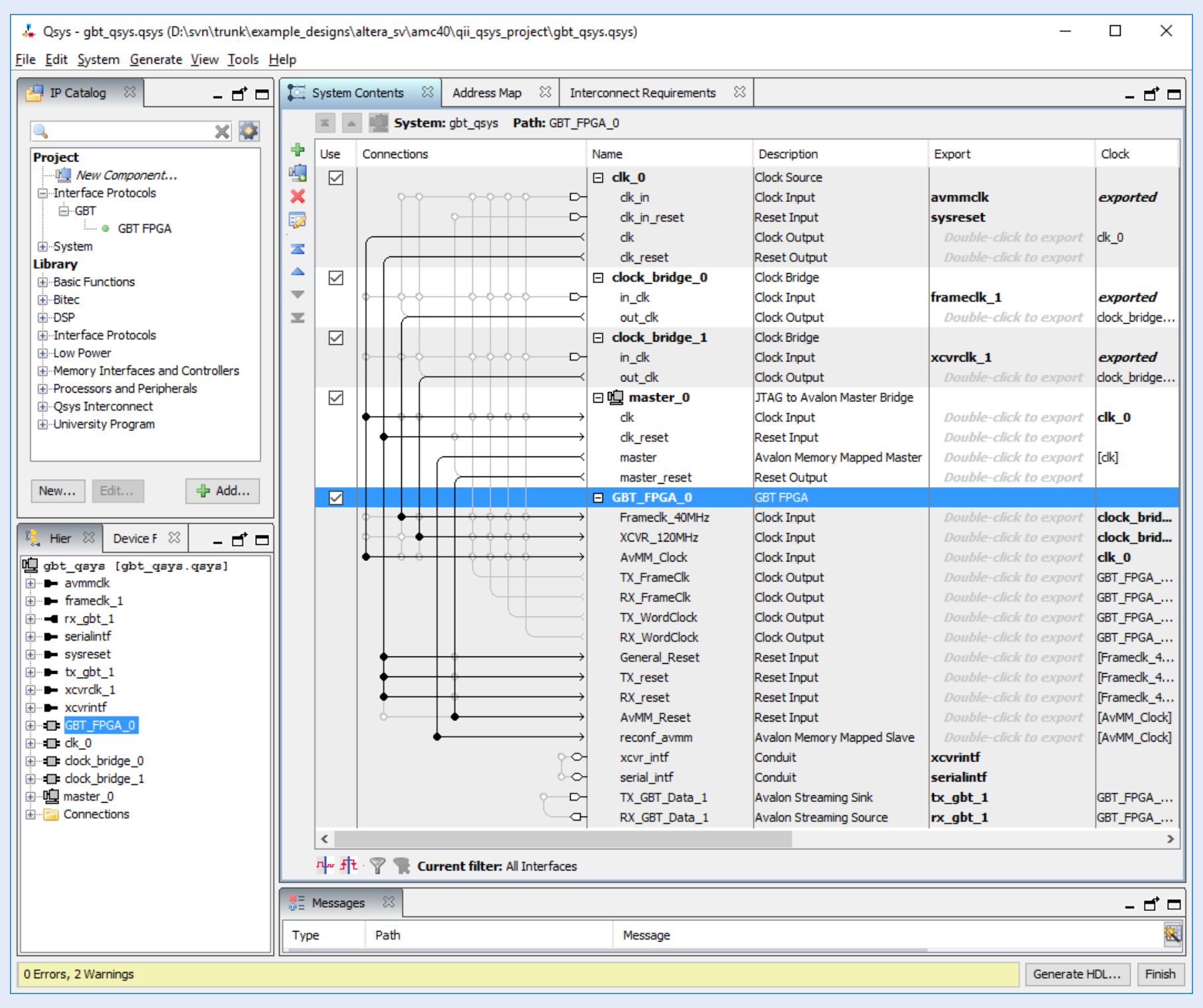
New releases of the GBT-FPGA are foreseen for 2016. They will include the support of the latest development tools, new FPGAs and the slow control feature presented in the adjacent sections.

A new version will be released in March including the correction of different bug, the improvement of the RX frameclock phase aligner module used with the RX latency optimized implementation as well as the simplification of the clock scheme. Concerning the development tool, new example designs were made for Vivado 2015 (Xilinx) and Quartus 15.0 (Altera).

In addition, an effort has been made to support QSYS for Altera devices in the coming release. This simplifies the GBT-FPGA IP integration for Altera development as shown on the figures below. The GBT-FPGA bank comes as a library-registered IP configurable via a wizard (number of links, TX/RX latency version an TX/RX encoding). The same implementation will be made for the Xilinx FPGA.

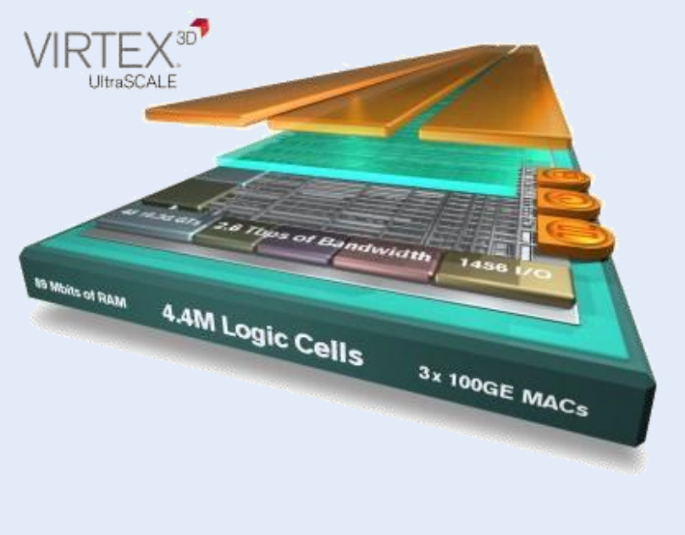


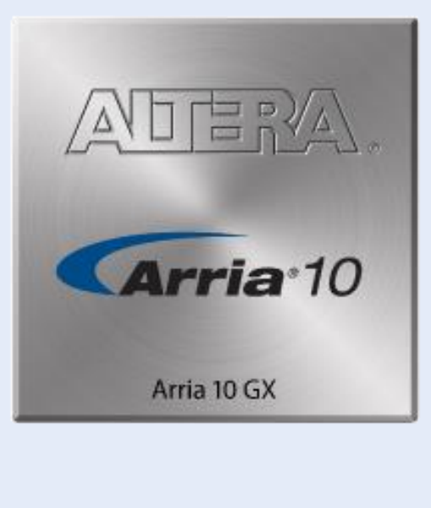
QSYS IP configuration



QSYS system implementing the GBT-FPGA IP

Two new FPGAs are going to be supported soon: the Ultrascale for Xilinx and the Arria 10 for Altera.

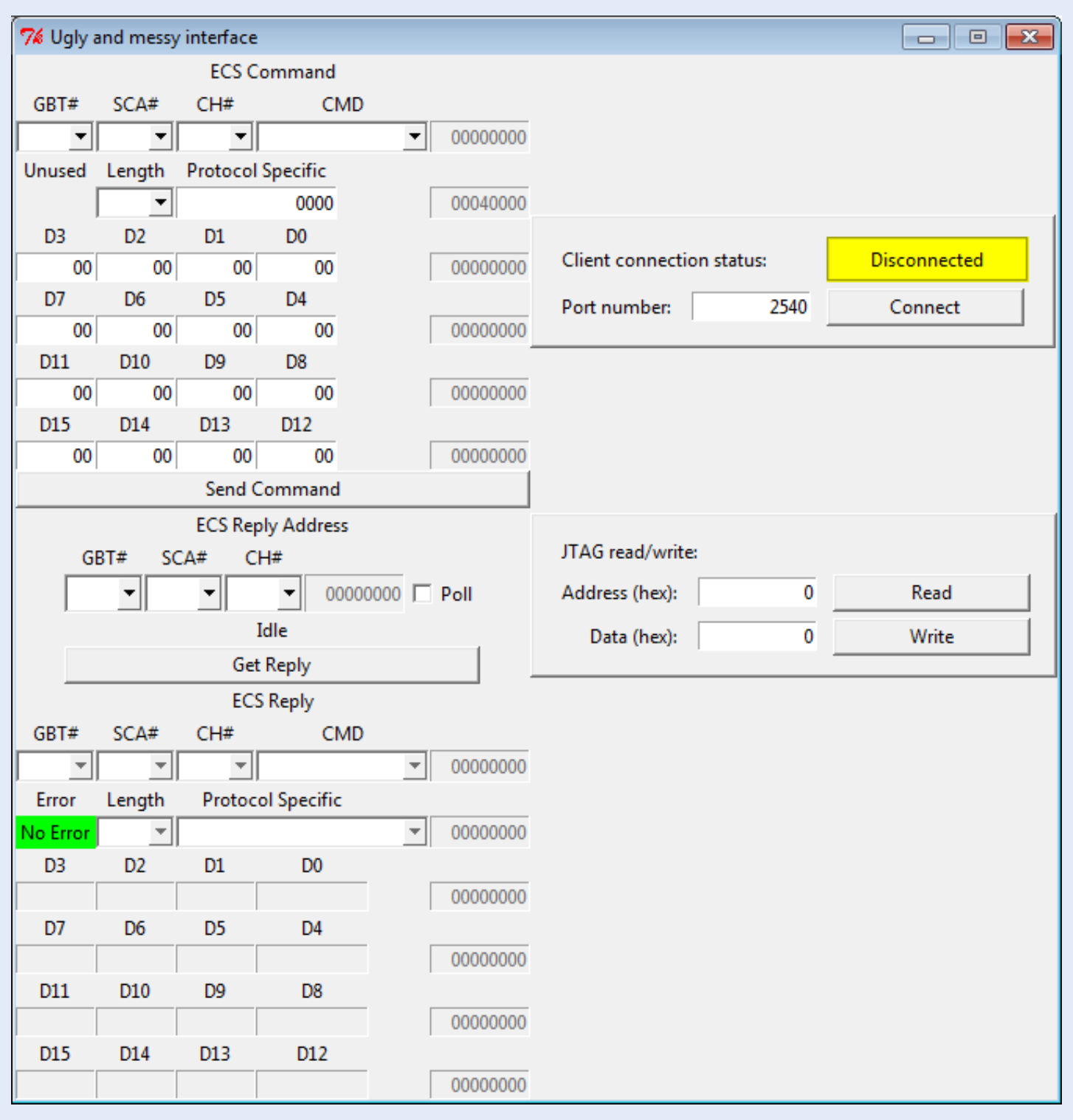




Implementation architecture of the GBTx / SCA slow control

The current version of the GBT-FPGA does not contain the slow control feature. That means that the bits dedicated to the internal and external control are fixed to a specific value. In a future release, an additional interface will be implemented to provide this slow control feature. The development will be based on an already existing module developed by LHCb. In this module, the VHDL-based development looks like an external IP, connected to the GBT-FPGA. This IP is configurable by a standardized bus (Avalon-MM) from Altera devices. The software control is performed using the graphical interface shown on the right, via JTAG. One part is dedicated to send the request to the front end and the other to get the reply. The different fields are described below:

- GBT# : Id of the GBT bank
- SCA# : Id of the SCA in the GBT bank selected (up to 21)
- CH# : Link selection (One GBT bank can implement many links)
- CMD : Command (I2C read, I2C write …)
- Length : Number of data bytes
- Protocol specific : Depends on the CMD (e.g: I2C address for I2C commands)
- D[12:0] : Data


Ugly and messy interface

ECS Command
GBT# SCA# CH# CMD 00000000
Unused Length Protocol Specific 0000 00040000
D3 D2 D1 D0: 00 00 00 00 00000000
D7 D6 D5 D4: 00 00 00 00 00000000
D11 D10 D9 D8: 00 00 00 00 00000000
D15 D14 D13 D12: 00 00 00 00 00000000
Send Command

Client connection status: Disconnected
Port number: 2540 Connect

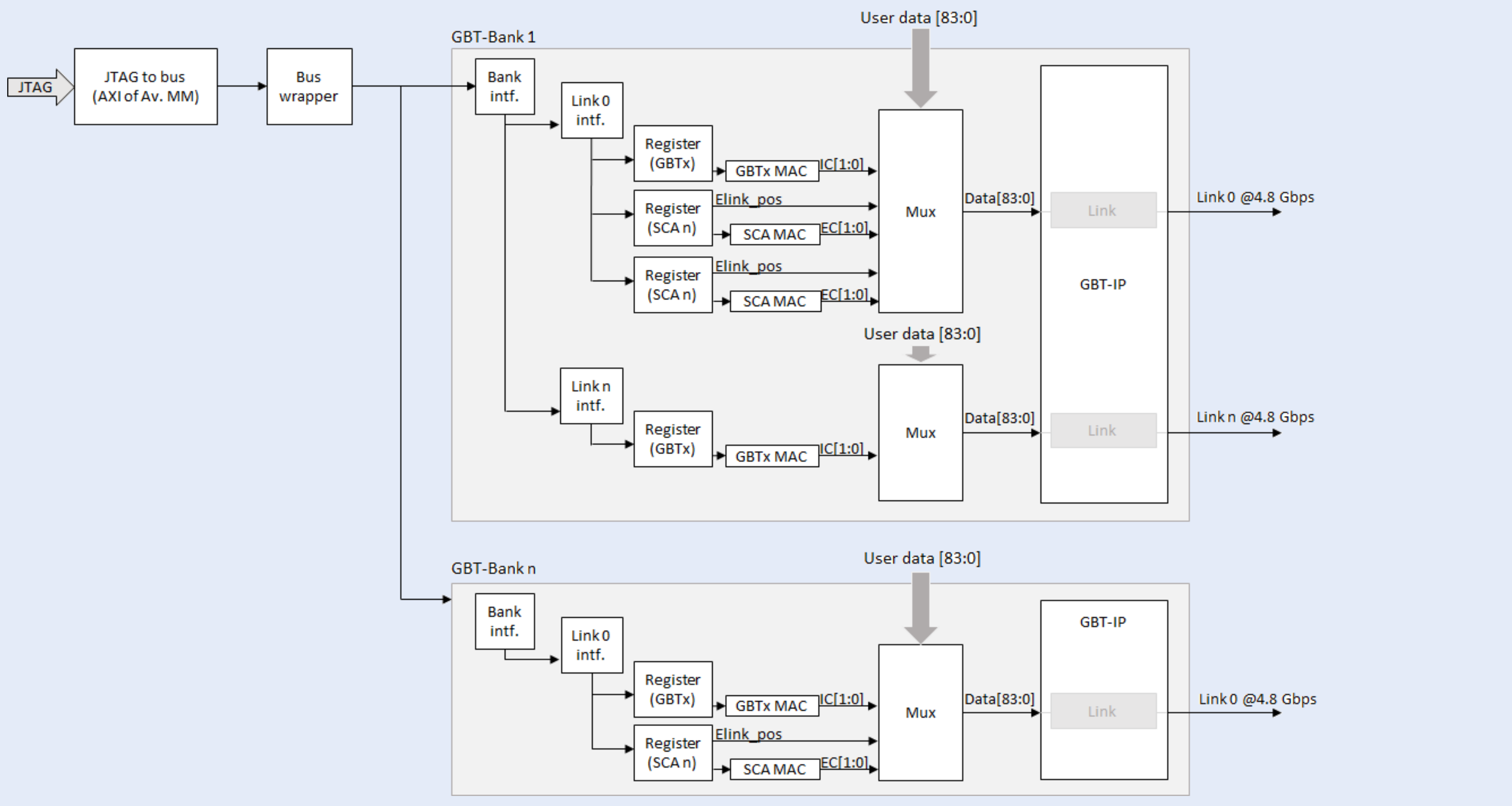
ECS Reply Address
GBT# SCA# CH# 00000000 □ Poll
Idle
Get Reply

JTAG read/write:
Address (hex): 0 Read
Data (hex): 0 Write

ECS Reply
GBT# SCA# CH# CMD 00000000
Error: No Error Length Protocol Specific 00000000
D3 D2 D1 D0 00000000
D7 D6 D5 D4 00000000
D11 D10 D9 D8 00000000
D15 D14 D13 D12 00000000

LHCb GUI to execute slow control commands

The VHDL-based IP will be modified to be generic and compatible with the different devices from Altera and Xilinx supported by the GBT-FPGA team. This implies the modification of the interface with the external system as well as the modification of implemented modules. The bloc diagram below shows the future architecture of the GBT-SC:



GBT-SC architecture

The GBT-SC IP will implement the following modules:

- The **Bank interface** provides a bus interface with the Fabric. This bus is a memory bus based interface that contains address and data buses as well as few signals for the control (write, read ..). The goal of this module is to redirect the data received to the selected link.
- The **Link n interface** works like a switch. Its goal is to redirect the data received from the Bank interface to the select SCA when a multiple instantiation per link is made by the user. It is possible to manage up to 21 SCAs per link.
- The **Register (SCA n)** contains the following table to be compliant with the already existing IP:

Address	Name	Length (word)	Comment
0x00	Control register	1	Control and status register
0x04	Reply addr	1	Full address of the channel to check for replies
0x08	Cmd header	2	Address, command code, length and protocol specific information
0x10	Data	4	Data to be transmitted
0x20	Reply header	2	Address, command code, length and protocol specific information
0x28	Reply data	4	Data from the SCA

- The **Register (GBTx)** has not be defined yet.
- The **SCA mac** generates the slow control 2 bits to be included into the GBT frame.
- The **Mux** places the SCA bits in the GBT frame.

Two additional modules, implemented in the example designs, allows communicating with the GBT-SC using the JTAG interface. These are device specific:

- The **JTAG to bus** is provided by the device manufacturer: JTAG to AXI for Xilinx and JTAG to Avalon-MM for Altera.
- The **Bus wrapper** is a VHDL-based module that manage the device specific bus to create the common memory mapped bus defined by the GBT-SC IP.

Conclusions

The GBT-FPGA core is currently supported for many devices from Xilinx and Altera. This VHDL-based IP can be configured to work in different modes with different encoding algorithms: "standard" vs. "latency-optimized" and "GBT frame" vs. "WideBus". The GBT-FPGA team is focusing its efforts on the implementation of the slow control feature with the internal control (GBTx) as well as the external control (SCA). This functionality will be released in June 2016. In parallel, the IP will be ported to the latest FPGA from the two companies previously listed (Ultrascale for Xilinx and Arria 10 for Altera) and IP blocs will be created to be used with QSYS (Quartus II) and Vivado. Finally, a new release including correction of minor bugs, simplification of the clock scheme and porting to the latest development tool will be made in March 2016.

ACES, CERN, 7 March – 10 March 2016