LpGBT Tester: an FPGA based test system for the lpGBT ASIC

J. Mendez, S. Baron, S. Biereigel, R. Francisco, J. Fonseca, D. Hernandez, S. Kulis, D. Porret, P. Moreira

CERN, 1211 Geneva, Switzerland

Abstract

The lpGBT transceiver is a radiation tolerant ASIC designed to be used in High Energy Physics detector systems. It aggregates data from up to 28 eLinks to one high-speed link running at 5.12 or 10.24 Gbps. In the downlink direction, it can be used for timing and trigger distribution by demultiplexing the incoming downlink bitstream running at 2.56 Gbps onto up to 16 eLinks. Moreover, the lpGBT provides additional interfaces to handle the detector slow control. This paper presents the lpGBT tester, based on a Xilinx development kit and a custom FMC card developed to perform the pre-production test.

Project overview

The Low Power GigaBit transceiver, named lpGBT[1], is a new radiation tolerant ASIC developed to handle the communication between the detectors and the back-end electronics of the CERN experiments. Planned to be used in the HL-LHC upgrades, this chip was designed in order to support a downlink connection at 2.56Gbps and serializes data coming from the on-detector electronics (uplink) at 5.12 or 10.24 Gbps. In addition of the main stream used to transmit the timing, trigger, control and data messages, the lpGBT supports several interfaces to monitor and control the experiments: I2C channels, GPIOs, 10-bit ADC and 12-bit DAC.

The lpGBT tester is based on a Virtex 7 evaluation board (VC707) connected to a custom FMC card featuring a socket made to host the ASIC. All of the interfaces of the device under test are connected either to the FPGA — running a custom firmware — or to analogue measurement and stimulus units. This poster is divided into three parts describing the Hardware of the mezzanine card, the Firmware and the Software that executes the tests sequences and stores the results. In order to track the development and to share the workload over the team, the design integration and merging of the tester was handled using gitlab according to the following flow:

Feature request → **New issue** → **Merge request** → **Approval** → **Merge**

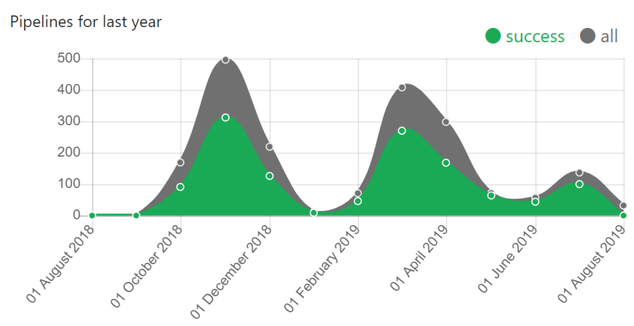
Following a discussion or just to progress and meet the specification, different features were requested during the project duration.

Following the first discussion, an issue is created on Gitlab where the developers can discuss on how the solution can be implemented and decide who can work on it.

A merge request automatically creates a branch (separated environment used to develop in parallel of the main stream) on which the feature can be implemented.

The code is reviewed by a second person of the team and the continuous integration tool checks the functionality of the system (pipeline execution, cf. figure) .

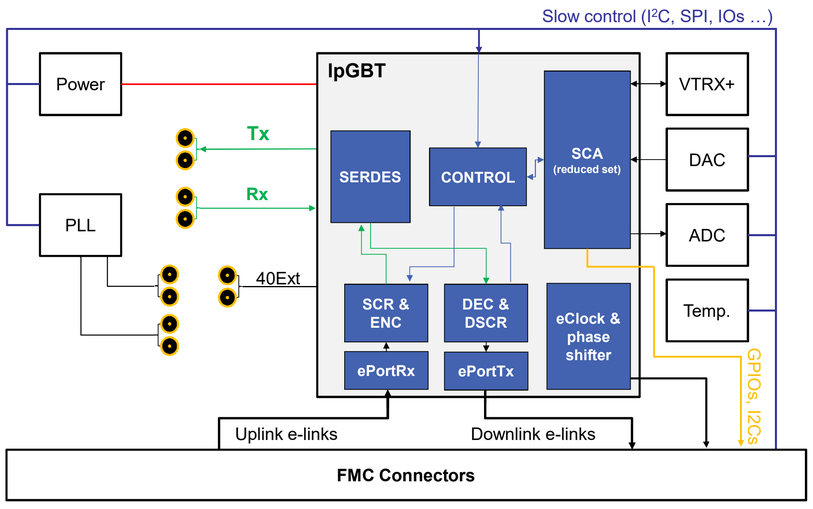
The new feature is pushed to the main branch of the project.



[1]: "The lpGBT: a radiation tolerant ASIC for Data, Timing, Trigger and Control Applications in HL-LHC" from Paulo Moreira [TWEPP 2019]

Hardware

The custom dual-port FMC card connects all of the eLinks interfaces to a Virtex 7 device in order to emulate a connection with a front-end device. The interface with the high-speed ports is made using two pairs of SMA connectors that can be connected to one of FPGA's transceiver interface. The eClocks and digital interfaces (GPIOs, I2C masters) are also connected to the Virtex 7, which implements the logic needed to test the feature as presented into the Firmware section. Finally, the additional analog channels available on the ASIC, which cannot be directly monitored by the FPGA, are connected to ADCs or DAC. The block diagram below shows the connectivity of all of the lpGBT interfaces.



As shown on the block diagram above, the power and clocks are configurable through the slow control interface over the FMC connector. It provides a way to characterize the ASIC's performance over the full supply voltage range, assess its power consumption and cover the different clock schemes: recovered or external clock source. The tester is also equipped with an external temperature sensor to ease environmental tests.

Software

The tester's software is divided in several layers going from the IPBus communication driver to the test sequences. The scripts, developed in python, take benefit from the inheritance mechanism and the decorator feature to define some guidelines for the design.

Base driver ← Inherits from ← **Interface drivers**

Defines the read and write functions to access the IPBus interface.

Each interface, corresponding to one firmware module (e.g.: I2C master, SPI …), has its own driver that implements dedicated functions

Imports

Tester decorator → Imports → **Tester class**

Calls

The decorator is used to initialize the logger, the connection with the IPBus core and all of the driver before running the test itself.

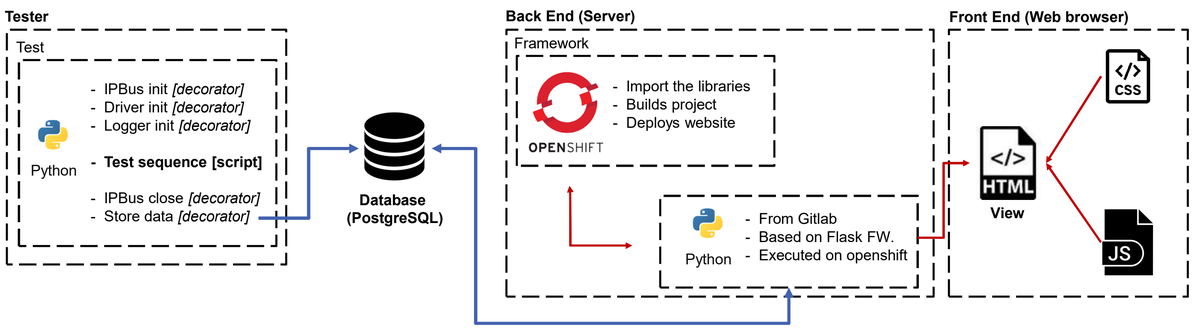
The tester class initializes all of the drivers and provides an interface to all the specific functions.

Feature test

Each individual test is scripted into one dedicated file. The use of the tester decorator makes the development easier as only the specific procedure has to be defined here. Finally, a global script can be called to execute each individual tests one after the other

The use of the tester decorator eases the development of the individual test as all the initialization processes are executed by design. Therefore, all the measured data returned at the end of the sequence are automatically stored into a generic database system based on the one originally designed for the electronics pool at CERN. This storage system was selected as it provides an easy solution to save the results and an interface to easily analyze the data and compute some statistics.

The figure below details the data management and shows the different processes involved. First, the tester decorator gets the measurement results from the user's script and stores them into a PostgreSQL instance. As the database design follows a generic enough architecture to work with the lpGBT tester, a flask based web application could be design to handle all kind of results and generate graphs or reports for analysis.



Firmware

The LpGBT tester firmware targets the Xilinx Virtex 7 FPGA hosted on the VC707 development kit. This setup provides an Ethernet interface used for the control of the tester via the IPBus protocol. This standard provides an easy way to read and write registers using python. Therefore, all of the modules developed to test the different interfaces of the lpGBT can be configured and monitored using a memory mapped bus. The diagram below shows the different cores implemented to check the functionality of the ASIC. The design can be easily separated into three parts: slow control features, data path and clocks.

Slow control

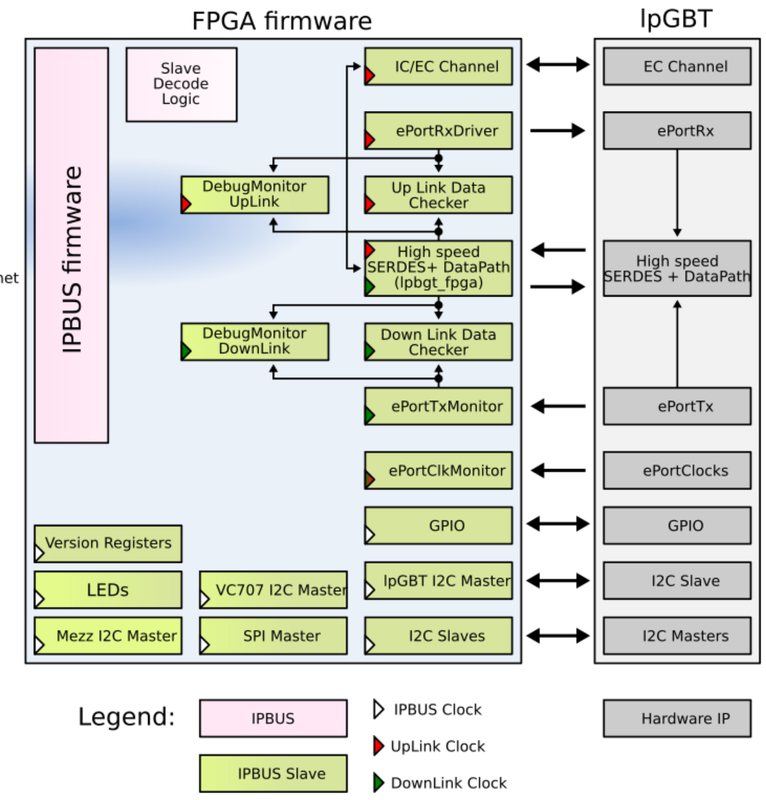
An I2C master is implemented to configure the lpGBT ASIC through its I2C slave interface. Once configured, the ASIC's slow control interfaces (I2C masters, GPIOs, ADCs, DACs) can also be tested by the firmware: The lpGBT I2C masters have their I2C slave counterparts implemented into the FPGA, GPIOs can be monitored or set via a dedicated core and the ADC and DAC channels are connected to analogue circuits controlled over dedicated I2C and SPI links (VC707 I2C Master, Mezz I2C Master and SPI Master).

Clocks

The ePortClkMonitor module[2] oversamples the 29 e-clocks outputs of the lpGBT. The sampling clock is generated using a PLL with the phase shifter enabled, allowing to get a picosecond resolution. Nevertheless, the FPGA input is limited to a frequency of 640MHz.

Datapath

The datapath checker itself is made of several modules. First of all, the ePortRxDriver is a PRBS7 generator sourcing data at different rates through the lpGBT ePortRx. The sequences are deserialized into the lpGBT and sent back to the FPGA through the high speed link at 10.24 or 5.12 Gbps. The sequence is then decoded by the lpGBT-FPGA core[3] and tested by the uplink data checker which counts the number of errors. The same process exists for the downlink path: PRBS7 sequences are sent over the high speed links and the ePortTxMonitor rebuilds the frames from the e-links to check their integrity.

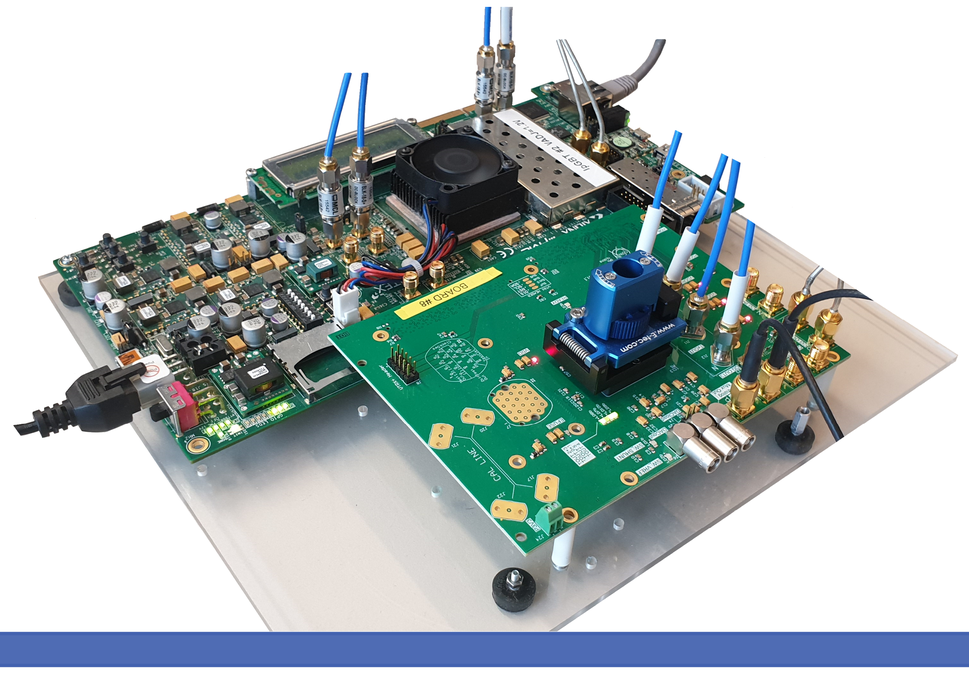


In order to parallelize the firmware development, each block was developed into a dedicated branch and was reviewed before being merged according to the flow described in the project overview section. The most complex blocks were checked in simulation first before being implemented and committed to the continuous integration environment.

[2]: "Methods for Clock Signal Characterization using FPGA Peripherals" from Stefan Biereigel [TWEPP 2019]
[3]: "New LpGBT-FPGA IP: Simulation model and first implementation" from Julian Mendez [TWEPP 2018]

Setup

The photo below shows the final lpGBT tester setup, which is made of the VC707 evaluation board from Xilinx and the custom dual FMC card that hosts the socket used to test the ASICs.



Conclusions

The lpGBT tester has been successfully used to fully characterize the first lpGBT ASIC prototypes and perform accurate SEU and X-ray test campaigns. It is now being used to test the first production lot. Because it is designed to be as configurable as possible, the tests can be ran to extract a lot of details for chip characterization or just quickly evaluate the functionality for mass production. Therefore, only a small part of the setup will have to be updated before using this system for the final production testing.

Topical Workshop on Electronics for Particle Physics, Santiago de Compostela, 2-6 September 2019