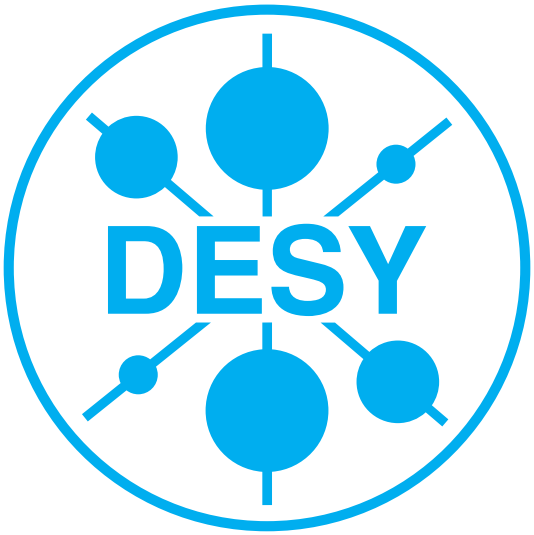


An Updated GBT Implementation for ngFEC in the CMS HCAL Upgrade

Jannicke Pearkes, University of British Columbia, Canada

October 19, 2014

The μTCA group at DESY is involved with testing and implementing the new generation front-end controller (ngFEC) for the upgrade of the Hadron Calorimeter at CMS. The ngFEC will provide control and diagnostic information over the front-end modules in the forward section of the Hadron Calorimeter. The new generation front-end controller will utilize updated hardware and firmware to provide increased reliability and faster data transmission in a more compact, reliable and power efficient design. This report will present the results of a seven week summer student project to implement an updated GBT serial communications protocol in the FPGA-based firmware of ngFEC.

Contents

1 Introduction

The Hadron Calorimeter (HCAL) at CMS measures the energy of hadrons produced in collisions at the Large Hadron Collider (LHC). The Hadron Calorimeter consists of alternating absorber and scintillator materials. Light is emitted when energy is deposited in the scintillator and is transported by fibres to phototransducers. The phototransducers convert the light to electrical charge which is digitized by the on detector electronics [1]. The reading from the on-detector electronics is then synchronized by the front-end electronics, and transmitted to the back-end electronics which process the data and communicate with the data acquisition (xDAQ), and crate control module (CCM) servers.

2 HCAL Upgrade of Front-End Controller

During Phase 1 of the LHC shutdown period many of the detector electronics of CMS will be upgraded, including the front-end controller (FEC). Firmware and software for the next generation front-end controller (ngFEC) is being developed at DESY and will provide the front-end electronics with fast and slow control signals. The controller will initially be integrated into the forward section of HCAL (HF) and will eventually be implemented for all sections of the hadron calorimeter.

The ngFEC will improve on the previous front-end controller by upgrading to the micro telecommunications architecture (μTCA) standard. The μTCA standard was developed in the telecommunications industry and offers significant advantages over the outdated Versa Module Europa (VME) standard currently being used at HCAL including an increase in data transfer rates and redundant connections [8]. These aspects are required to cope with the increased data volumes from the upgrade in luminosity of the LHC and will improve the reliability and maintainability of the back-end electronics system [8]. Other benefits of the μTCA standard include a more compact, power efficient design, and hot swappable capabilities. In particular the hot swappable design is advantageous for use at HCAL as it allows one to remove malfunctioning cards and replace them without disrupting the control of all front-end electronics.

3 DESY Test Set-Up

The next generation front-end controller for HF will consist of over 720 control channels. It will be capable of distributing fast controls at 40.0788 MHz, and slow controls at 100kHz. The fast controls include the LHC clock, trigger, broadcast and reset commands, and the slow controls include enabling power, receiving temperature information and other diagnostic bits. The DESY test stand is used to test prototype boards used in the HCAL control system and develop the firmware for the ngFEC. The DESY test stand is one of three stands devoted to the upgrade of the HCAL control system, the other two test stands are at CERN and Fermilab.

Figure 1: μTCA crate at the DESY Test-Stand

The DESY test set-up consists of the following electronic modules:

- 4 Gigabit Link Interface Boards (GLIB v3)
- 1 Advanced Mezzanine Card (AMC13)
- 1 μTCA Carrier Hub (MCH)
- 1 FMC Carrier - Xilinx Series 7 (FC7)
- 1 next generation Clock Control Module (ngCCM)
- TTC/TTS Tester Module (TTT)
- 1 Charge Integration and Encode (QIE) Bridge FPGA

The GLIB_v3 cards, AMC13 and MCH are positioned in a μTCA crate which will reside in the shielded CMS underground service cavern for back-end electronics. The ngCCM will be placed in the same front-end crate as the QIE 10 detector readout modules inside the detector cavern, and therefore must be radiation hard [1]. In the DESY test-stand the QIE Bridge FPGA emulates the QIE10 modules that will measure the charge from the on detector phototransducers. The TTT module emulates the LHC clock, trigger and fast controls and is used for testing purposes. Tests are underway to determine the feasibility of using the FC7 module as an upgrade to the GLIB cards.

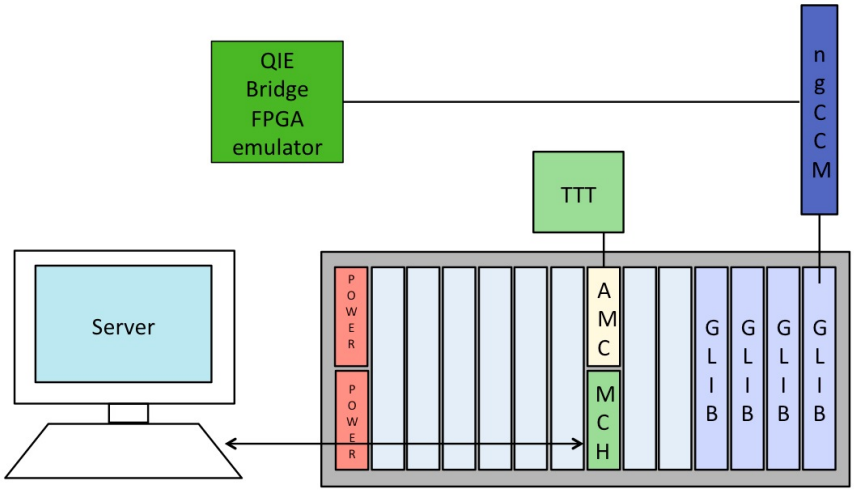


Figure 2: DESY Test Set-Up

Controls from the Control Clock and Monitoring (CCM) server arrive in the μTCA crate through an Ethernet connection to the MCH. This data is then and sent across the μTCA backplane to the GLIB cards. The emulated LHC Clock of 40.0788 MHz, trigger and fast controls pass from TTC/TTS Test Gadget through the AMC13 and is distributed across the back plane to all GLIB cards. The GLIB cards then interface between the data passing between the backplane and the ngCCMs. Data to the GLIB card is decoded using the CERN developed IPbus protocol and serialized using the CERN developed GBT protocol. Each GLIB card will control and communicate with 4 ngCCMs via a 4.8 GBps optical link. The ngCCM in turn communicates with the QIE Bridge FPGA.

3.1 ngCCM Crate Construction and Testing

During the summer school a crate was assembled for the ngCCM and a front panel was attached to allow for safe handling of the board. The Eurocard backplane connectivity was tested on site and the voltages on each of the 192 pins in connectors J8A and J8B were tested. The connectivity and voltages agreed with the requirements for the ngCCM as defined in the HF motherboard R01 prototype schematic and the crate was determined to be suitable for the ngCCM. In addition cooling fans were mounted on the top of the crate and connectors were soldered to interface with the QIE bridge FPGA.

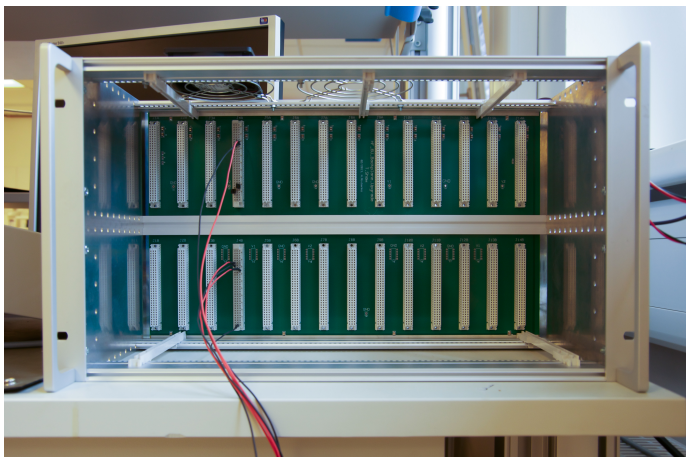


Figure 3: Assembled ngCCM crate

4 ngFEC Firmware

The firmware for the next generation front end controller is designed to be compatible with the CERN developed Gigabit Link Interface Board. The GLIB card is a double-width advanced mezzanine card designed to reside inside a μTCA crate [2]. Some of the features of the GLIB card include 4 SFP connections, 2 FMC connections, Ethernet and JTAG ports, and a high performance Virtex-6 (XC6VLX130T) FPGA onto which the ngFEC firmware is placed.

Figure 4: GLIB Card, *GLIB Project Page [2]*

The firmware for the ngFEC consists of two main modules, the GLIB system firmware developed primarily at CERN, and the user logic being developed at DESY and the University of Virginia. The user logic of ngFEC consists of the ngFEC module, the ngCCM module, the GBT module and the DTC module. The ngFEC module buffers and decodes information sent via the IPBus protocol from the server. After encoding it converts the necessary commands to I^2C signals. The DTC handles the signals coming from the TTT module. The ngCCM module takes the data from the ngFEC modules and formats it to be compatible with the ngCCM. Finally the GBT module encodes/decodes and serializes/deserializes the data going to and from the ngCCM module using the GBT serial communications protocol.

5 GBT-FPGA Project

Gigabit Data and Timing (GBT) is a CERN developed serial communications protocol for high energy physics experiments. GBT is fault tolerant and includes several encoding features to ensure a fast and reliable performance in heavily radiated environments. In the next generation front-end controller GBT is used for serial communication via the gigabit optical link between the GLIB card and the ngCCM.

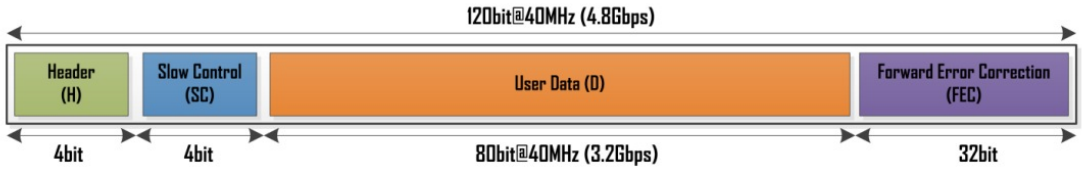


Figure 5: GBT encoding scheme, *GBT-FPGA User Guide [6]*

For the ngFEC the data is encoded using the GBT-Frame encoding scheme described in Figure 5. This encoding scheme sends a 4 bit header that is used to align the data and communicate whether the controller is in idle mode or data mode. The forward error correction bits are additional bits passed by the Reed Solomon encoding algorithm in order to recover from transmission errors due to ionizing radiation.

The GBT-FPGA project has developed a framework for implementing GBT links. Up to four GBT links can be assigned to a GBT bank. The GBT link consists of three main components as shown in Figure 6. The GBT Tx module scrambles, encodes and interleaves data coming from the FPGA. The Multi-gigabit Transceiver serializes and deserializes the data going to and coming from the FPGA using Xilinx's Gigabit Transceivers. GBT Rx de-interleaves, decodes and descrambles the incoming data.

The data is scrambled to reduce the occurrence of long sequences of 1s or 0s in the bit stream [3] in order to achieve a good DC balance. DC balance is important in serial com-

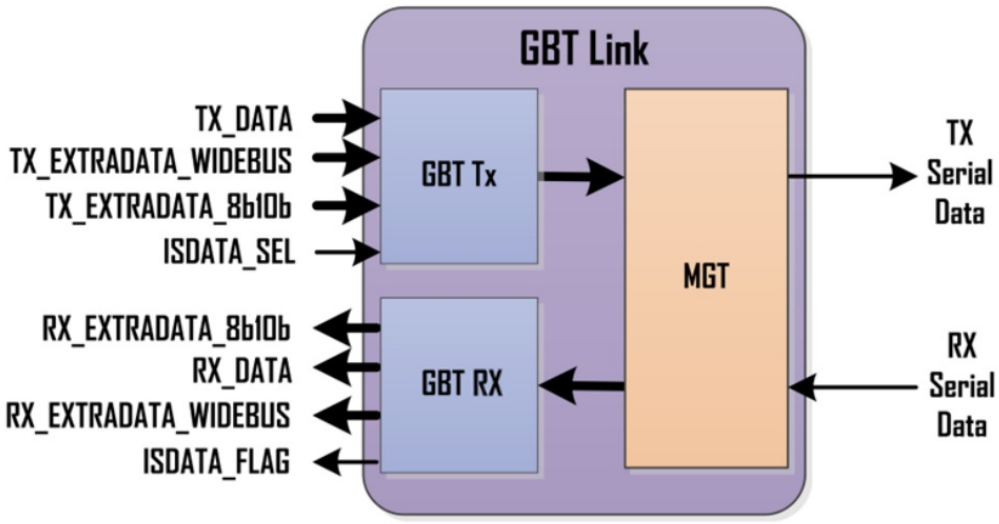


Figure 6: Schematic of GBT Link, *GBT-FPGA User Guide [6]*

munications to counteract bit errors occurring due long sequences of 1s or 0s charging the capacitors used in the high pass filters of the transmission circuits. After scrambling the data is then encoded using a Reed-Solomon encoding (RS) algorithm. The encoding scheme allows one to recover information lost due to bit errors from ionizing radiation or damaged links. The Reed Solomon (RS) encoding algorithm is interesting because it allows for error correction while sending a comparatively small number of additional bits. The encoding structure used in GBT is double interleaved RS(15,11) meaning that a sequence of up to 16 consecutive bits can be corrected while only adding 32 forward error correction bits to the original message length of 88 bits [5].

There are two versions of GBT, the "standard" version and the "latency optimized" version. The standard version has non-deterministic latency and whereas the latency optimized version has a low, deterministic latency. The standard version is well-suited for non-time critical applications such as (DAQ) and the latency optimized version is well-suited for time dependent applications such as timing, trigger and control [6]. Although the latency optimized version is best suited for ngFEC it is a much more complex implementation than the standard version, therefore an attempt at implementing both the latency optimized and standard versions was made.

6 GBT Implementation for ngFEC

During this project the GBT implementation in the ngFEC firmware was updated. This was done in order to make the GBT implementation more modular and less resource intensive. The aim of the project was to increase the number of SFP connections on the

GLIB card from two to four.

To update the design the previous GBT implementation was removed and an extensively modified version of the gbt_3_0_2 reference design was integrated with the ngFEC firmware. The modifications to the reference design included widening the busses of the GBT reference design to accommodate up to four SFP connections, adding a generate loop inside the GBT reference design in order to generate enough components for 4 SFP connections, and modifying the pattern generator module to receive input from the ngCCM. The new data types required to widen the busses of the gbt_3_0_2 reference design were placed into user_gbt_package.vhdl. Finally the ngFEC user logic file was cleaned for improved readability.

These steps resulted in simpler and more modular design as all GBT files are contained within one package. The modular design means that it will be easier to update in the future and easier to port to the FC7 card which is being investigated as an upgrade to the GLIB card.

A second challenge was that the clocking resources and paths are limited on any FPGA. In the ngFEC design there are many time dependent processes and many timing constraints, finding a solution to these placement issues is non-trivial and requires careful placement of the logic and clocking resources. On the Virtex 6 (XC6VLX130T) there are 10 clocking regions each with a limited number of buffers (BUFGs), mixed mode clocking managers (MCMMs), and clocking routes. [10]. Each clocking region is limited to 12 clocks sourced by a horizontal clock buffer (BUFH) per region.

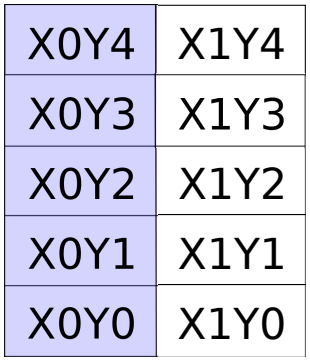


Figure 7: Clocking Regions of the Virtex 6 XC6VLX130T - highlighted regions contain 2 MCMMs

Attempts to place the resources effectively included releasing user timing and placement constraints, placing the MMCMs in different areas of the FPGA, and changing the areas where the user and system logic was placed on the FPGA. However, none of these attempts were particularly successful. In cases where the constraints on the system were relaxed, the design was either unroutable by the automatic routing system, or the tim-

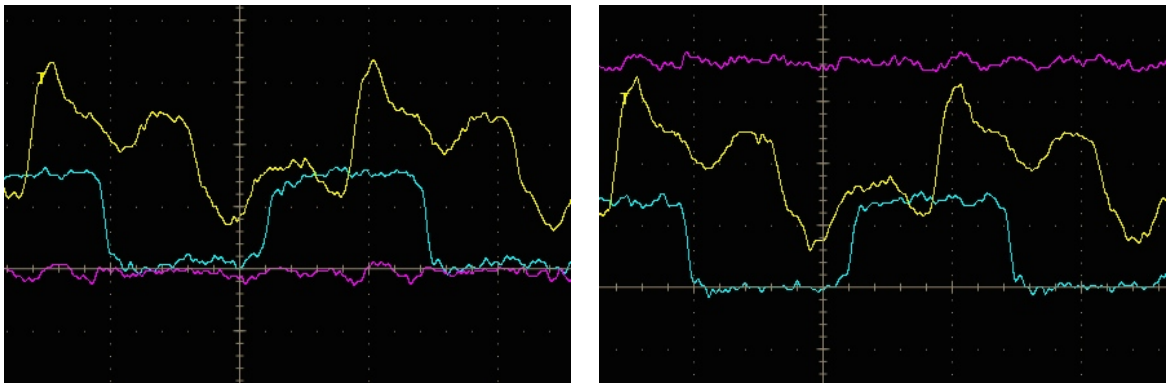
ing constraints were relaxed to the point where the design did not function as intended. Other cases where the constraints were slightly tighter the design either contained unroutable component pairs or the timing constraints that were unmet. During the time period of this project only two fully routed and properly constrained links utilizing the standard version of gbt_3_0_2 were acquired.

7 Testing

Testing of the updated design was performed by uploading the design to the GLIB card and sending test commands from the cmscastor1 server. Test commands allowed the registers on the QIE Bridge FPGA to be read from or written to, and provided a method for sending control signals to the ngCCM. The signals on the ngCCM and clock signals from the GLIB card could both be monitored using an oscilloscope allowing one to check the power enable pin on the ngCCM and the alignment, shape and frequency of the clocks.

8 Results

Both the latency optimized and standard versions of GBT were implemented. The latency optimized version was implemented using gbt_0_1_0_beta and gbt_3_0_1. The standard version was implemented for gbt_3_0_2. It was possible to read out temperatures from the ngCCM and read and write to registers on the QIE Bridge FPGA using one GBT-link for all versions.



(a) Power Enable Off (b) Power Enable On

Figure 8: Effect of the power enable command sent via GBT to the ngCCM

The latency optimized version showed promising results with slow controls, however metastability was observed in the power enable signal when implementing more than one GBT-link. This may have been due to the fact that the design was under-constrained at the time of implementation. Further systematic tests utilizing the constraints provided in the reference design may improve the performance of the latency optimized version

of GBT.

The standard version of the gbt_3_0_2 design was implemented with one and two SFP links. Slow controls were effective, however a misalignment of the Tx frame clock and the received clock on the ngCCM was apparent. These two clocks, pictured as the blue signals and yellow signals in Figure 9, operate at the LHC clock frequency of 40.0788 MHz and should be in phase with each other. However, each time the FPGA was reprogrammed the phase alignment shifted. This is likely due to the non-deterministic latency of the standard implementation of GBT.

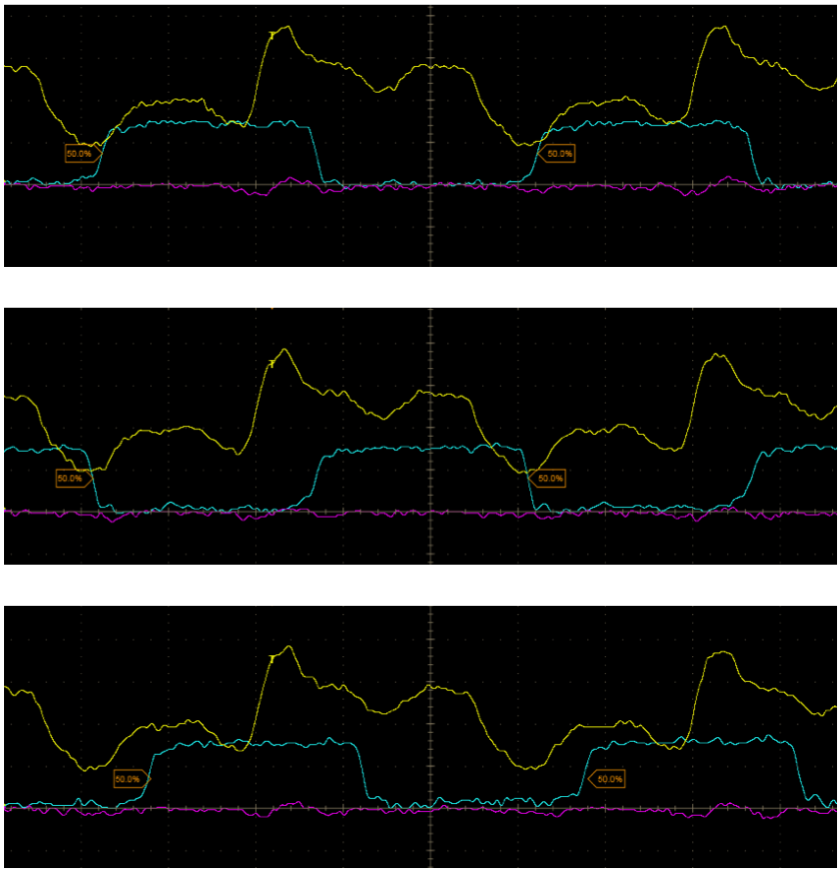


Figure 9: Changing TX frame clock alignment after reprogramming of the FPGA

The resource usage between GBT implementations varied between GBT releases and the choice of optimization being used. The gbt_3_0 releases utilized an additional TX phase locked loop. The latency optimized version requires a separate RX phase aligner for every additional GBT-link. To further improve the resource usage in the future, certain components, such as the RS decoder could be shared [7].

Table 1: Resource Utilization for 1 GBT Link in Reference Design 0.1.0

Resource	Standard	Latency Optimized	Available
Slice registers	12,492 (7%)	16,241(10%)	160,000
MMCMs	3 (30%)	4 (40%)	10
BRAMs	45(17%)	78(29%)	264

Table 2: Resource Utilization for 2 GBT Links integrated with ngFEC

Resource	Original ngFEC	Standard	Available
Slice registers	24,409(15%)	24,583 (15%)	160,000
MMCMs	7 (70%)	6 (60%)	10
BUFGs	26(81%)	24 (75%)	32
BRAMs	108(40%)	112 (42%)	264

9 Conclusion

The new GBT implementation is an improvement on the previous design. Slow controls are functional and fast controls are being sent to the front end. However, additional investigation is required to ensure the alignment of the tx_frameclk with the clock being received on the ngCCM. An implementation of the latency optimized version of GBT 3.0.2 may provide a solution to this problem with stable phase alignment. Further work to be performed on the ngFEC firmware includes careful placement of the FPGA logic in order to achieve four GBT-links on the Virtex 6 FPGA, such that the design meets all timing and stability requirements.

10 Acknowledgements

I would like thank my supervisors Özgür Şahin and Ulf Behrens, and the CMS μTCA group for providing me with this project and the resources and support to work on it. I would especially like to thank Özgür for his patience, persistence and advice while working on this project together. Finally, I would like to thank the organizers of the DESY Summer School for running such an fantastic program and providing such an amazing learning opportunity.

References

[1] CMS Collaboration, "CMS Technical Design Report for the Phase 1 Upgrade of the Hadron Calorimeter" Technical Report, CERN-LHCC-2012-015, (2012).

[2] CERN. The GLIB Public Project Page:
https://espace.cern.ch/project-GBLIB/public/default.aspx

[3] F. Marin et al., "Implementing the GBT data transmission protocol in FPGAs". Presentation. Topical Workshop on Electronics for Particle Physics, (2009)

[4] P. Vichoudis et al., "GLIBv3 Manual 1.20". (2013)

[5] S. Baron et al., "Implementing the GBT data transmission protocol in FPGAs". TWEPP-09 Topical Workshop on Electronics for Particle Physics, (2009)

[6] S. Baron, M. Barros Marin. "GBT-FPGA User Guide", (2014)

[7] S. Baron, "GBT-FPGA Firmware Starter Kit for Altera and Xilinx Devices", (2012)

[8] M. Şahin et al., "CMS HCAL Slow Control Upgrade". Poster, (2013)

[9] Xilinx Incorporated, "Constraints Guide". UG325, (2012)

[10] Xilinx Incorporated, "Virtex-6 FPGA Clocking Resources User Guide". UG362. (2014)