

Implementation of I2C bus master controller for CRU Slow Control in ALICE at LHC

Shuaib Ahmad Khan^{1,*}, Filippo Costa², Erno David³, Jubin Mitra¹, Tivadar Kiss³, S. Mukherjee⁴, Rourab Paul⁵, Tushar K. Das¹, A. Chakrabarti⁵, Tapan K. Nayak¹

¹Variable Energy Cyclotron Centre, HBNI, Kolkata, INDIA

²European Organization for Nuclear Research (CERN), Geneva 23, Switzerland

³Research Institute for Particle and Nuclear Physics, Wigner Research Center, Budapest, Hungary

⁴Centre for Astroparticle Physics and Space Science, Bose Institute, Kolkata, India

⁵University of Calcutta, Kolkata, INDIA

* email: shuaibkhan@vecc.gov.in

Introduction

The ALICE (A Large Ion Collider Experiment) collaboration plans for a major upgrade of the detectors during Long Shutdown 2 (LS2), to commence from the year 2020, in order to enhance the scientific discovery potential in the Run3 of the Large Hadron Collider (LHC) [1]. After LS2 the luminosities and hence the interaction rates will be increased by six times in Run3. To cope up with the increased interaction rate and in turn the large event size, a new approach based on Common Readout Unit (CRU) is being developed. It is a functional block placed in between the detector systems, the Data Acquisition System (DAQ) and the central trigger processor. The communication between the Front-End Electronics (FEE) of the detectors and the DAQ via CRU is mainly composed of three functional subsystems, a fast trigger timing and distribution system to deliver the system clock and trigger information, a data acquisition link carrying the collected data from the detector to the control room and a slow control system carrying bidirectional traffic from the control room and the embedded electronics on the detector. The slow control system is responsible to distribute control sequences and collect status information from the embedded peripheral electronics in the detector. There is a well-defined protocol stack for the slow control communication through CRU to the detectors. Inter Integrated Circuit (I2C) bus protocol [2] is an integrated part of this hierarchy. This article presents the implementation of I2C master bus controller on Altera Field Programmable Gate Array (FPGA), which is an integral part of the slow control development for CRU project.

Protocol stack for CRU slow control

There are uplink and downlink data paths for slow control data communication. Uplink is from detectors to control room via CRU and from control room via CRU to detectors is known as downlink. Data transmission from detector to CRU via optical fibres is in Gigabit Transceiver (GBT) standard [3]. It is accomplished using GBT chipset consisting of radiation tolerant GBTx ASIC [3] and GBT-SCA (Slow Control ASIC) [4] along with the other components. Communication between GBTx ASIC and GBT-SCA is through High Level Data Link Control (HDLC) protocol. Uplink data from the detector to the GBT chipset uses different buses like I2C. GBT-SCA will wrap up the data in SCA command oriented channel protocol (SCCP) format to instruct the execution of specific operations. GBT-SCA command frame will encapsulate it in the HDLC frame format. The GBTx ASIC prepares this data in the GBT frame format and sent it to the CRU via optical fiber as shown in fig 1. CRU will extract the payload and forward it to DAQ.

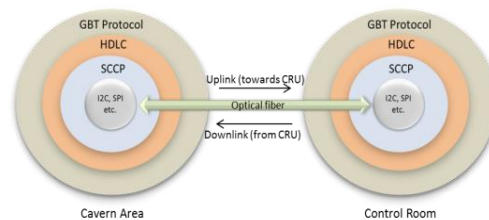


Fig. 1: Protocol stack for CRU slow control

In the downlink path, data and instructions from control room to the detector system goes via CRU where it will encapsulate the data in the format as shown in fig 1. The encapsulated data

is sent through high speed transceivers on CRU to the detectors via optical fibers. I2C bus master is utilized at different points. It will be used to control and monitor different slave devices present on the CRU board itself like transceivers and external PLL clock frequency selection and different parameters on the detectors/FEE. The GBTx ASIC can also be controlled and monitored via an I2C interface.

Implementation of I2C bus master controller in Altera FPGA board

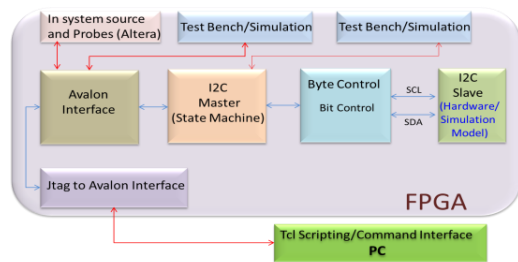


Fig. 2: I2C bus master controller implementation and simulation test setup

I2C is a two wire, single-ended, multi master, multi slave, serial computer bus. It was defined by Philips providing a simple method to exchange data between devices by using a minimum number of pins. I2C bus master firmware module was designed using VHDL syntax [5]. The design was simulated using Modelsim-Altera [5] and synthesized using Quartus Prime standard edition 15.1 and implemented on Altera Cyclone IV EP4CE22F17C6 FPGA. Memory (part no. 24LC02B) on the FPGA board was used as the I2C slave. The implementation and simulation test setup is shown in Fig. 2. Resistor Transfer Logic (RTL) simulation model of memory was used for simulating the I2C bus master. RTL model of the bit control and byte control was connected to the serial clock (SCL) and the serial data (SDA) lines of the slave as shown in fig 2. The bit controller handles the actual transmission of data and the generation of the specific control instructions [6] for START, Repeated START, and STOP signals by controlling the SCL and SDA lines. The byte controller tells the bit controller which operation has to be performed. A customized state machine was developed in VHDL for the reading and writing to different

I2C slaves. This state machine was interfaced with the bit control and byte control modules. A 32 bit Avalon memory mapped interface was developed and integrated with the state machine. With the Avalon interface it establishes the communication with the processing block. Test bench for simulation was developed at the state machine level and at the Avalon interface level. Hardware monitoring in FPGA was done through In-System-Sources and Probes (ISSP) tool of Altera, and through the Tool Command Language (TCL) scripting, using the command line terminal. I2C scanning feature was also added to TCL script. Using this feature the different slaves that are available for the communication on a FPGA board also be detected.

Results and Conclusion

Data read/write from the slave using the I2C master controller and the results of the scanning feature using the TCL script is shown in fig. 3.

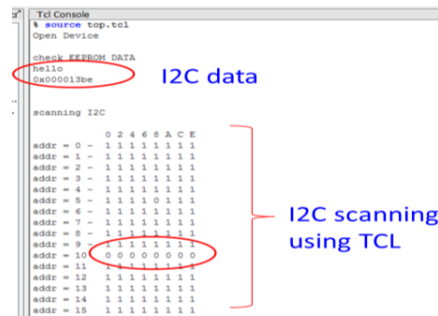


Fig. 3: I2C data read/write and Scanning

Error handling states will be added to this development to make it resilient to erroneous conditions. This development could be integrated and tested with the first version of the CRU firmware. Further results will be presented.

References

- [1] TDR for Upgrade of the ALICE Read-out & Trigger. LHCC-TDR-015, 3rd July 2014.
- [2] User manual I2C-bus specification Rev.6, 4 April 2014, NXP Semiconductors
- [3] GBT data transmission. S. Baron et al.
- [4] The GBT-SCA, a radiation tolerant ASIC in HEP exp. A. Caratelli et.al. TWEPP 2015
- [5] Quartus Handbook Volume1, altera.com
- [6] I2C-Master Core Specification Richard Herveille Rev. 0.9 July 3, 2003