TOFHIR2 ASIC Data Transmission VHDL Simulation

Project Report for a 2023 Summer Student

Akbar Saleh

Supervisor: Mehmet Ozgur Sahin

Table of Contents

Abstract

This project report describes the work done during an 8-week summer student internship at CERN to create a TOFHIR2 simulator. The project was part of the larger upgrade of the LHC for the high luminosity era and aimed to simulate the data transmission process of the TOFHIR2 ASIC, which is an integral part of the minimum ionizing particle timing detector. The report provides an overview of the TOFHIR2 and its location in the barrel timing layer, as well as the five stages of the TOFHIR2 data transmission process. The project focused on simulating this data transmission process using VHDL. The report shows the VHDL simulation code structure, purpose of each part of the code, and simulation results. Although the simulation code has some limitations, it provides a starting point to create a more comprehensive simulator that could be used to test the actual TOFHIR2 ASIC before making changes. Overall, this project contributes to the ongoing efforts to upgrade the LHC for the high luminosity era.

Introduction

There are many upgrades that are being done on the LHC to prepare it for the high luminosity (HL-LHC) era. One of these upgrades is the planned addition of a minimum ionizing particle (MIP) timing detector, or in short, MTD. The MTD consists of two main layers, which are the barrel timing layer (BTL) and the endcap timing layer (ETL), and these are connected to common systems such as the data acquisition (DAQ) system and the L1 trigger.

This paragraph provides an insight into the BTL since it is the layer that contains the TOFHIR2 ASICs. The basic electronics building block of the BTL is the readout unit (RU). The readout unit contains several electronic components. For instance, it has several frontend cards, and each one of them contains TOFHIR2 ASICs.

The image below shows where the TOFHIR2 is located:

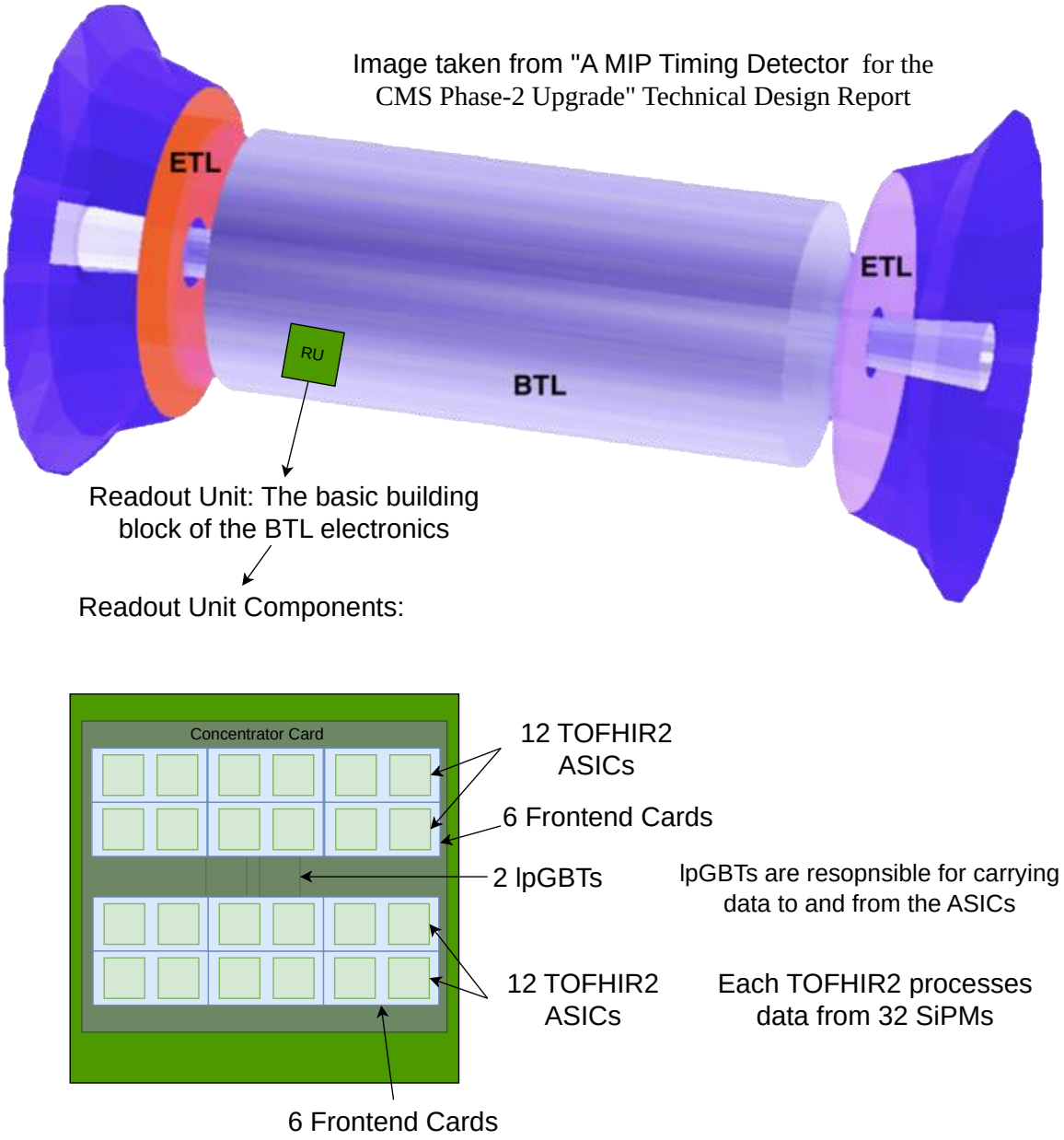

Figure 1: The TOFHIR2 is in the frontend of the detector where it can sense particles collisions and collect their data. It processes data from 32 SiPMs, which are radiation detectors.

The MTD DAQ system backend processes the data collected by the TOFHIR2 from the frontend. The data are transmitted from the frontend to the backend via Versatile Links+ (VL+), which consist of communication ASICs called lpGBTs and optical transceivers called VTRx+. In

the backend, large Xilinx FPGAs called Serenity boards unpack and process the data received from the frontend through VL+.

TOFHIR2 Data Transmission Process

This section explains the TOFHIR2 data transmission process that this project is trying to simulate. This process consists of five stages: data sensing, data filtering, encoding data, buffering data, and data transmission from the TOFHIR2 buffer to backend. The first stage of the process is the data sensing stage. This first stage is continuous, which means that the sensors are continuously reading radiation raw data coming from the environment in which they are placed. Then, in the data filtering stage, the L1 trigger is used to filter which of the continuous raw data are important. Only the important raw data are entered into the TOFHIR2 buffer. In the third stage, the encoding data stage, the raw data that was entered into the buffer in 88-bits get encoded into the corresponding encoded 110-bits. In the next stage, the raw data get stored in the buffer temporarily. Lastly, in the data transmission stage, the buffered data are sent through VL+ to the Serenity boards in the backend for processing.

TOFHIR2 Data Transmission Simulation

The purpose of this project is to simulate the data transmission process of the TOFHIR2 using VHDL. This section will show the simulation code structure, explain the purpose of each part of the code, and show the simulation results.

Code Structure

The code was divided into two parts: design and simulation. The design part contains the VHDL code that simulates the data transmission process of the TOFHIR2. Ideally, these design files should exactly match what the TOFHIR2 would do during data transmission. The design contains four files, which are tofhir2_simulation.vhd, raw_data_encoders.vhd,

encoder_8b10b.vhd, and data_buffer.vhd. The purpose of each file will be explained in the next section. The simulation part contains only tofhir2_simulation_tb.vhd.

Purpose of Different Parts of the Code

There are four design files in the design part: tofhir2_simulation, raw_data_encoders, encoder_8b10b, and data_buffer. The top level module is tofhir2_simulation. This module instantiates the raw_data_encoders and data_buffer modules, and it also does the required wiring between them. The second module is raw_data_encoders, and the purpose of this module is to read the raw events data of 88-bits, and encodes them into 110-bits. The raw_data_encoders does the required encoding by instantiating 11 of encoder_8b10b module. The encoder_8b10b module is the module that contains the logic to do 8b/10b encoding. The last module, data_buffer, is the module that takes the encoded data outputted from raw_data_encoders, breaks it into 8-bits chunks, buffer it, and transmits it to the backend.

The simulation file tofhir2_simulation_tb reads the raw events data file to provide input to the design, and it also generates clock signals.

Simulation Results

This paragraph will list all inputs and outputs that will be shown in the images below and explain their purpose. In all the images, there are six inputs, which are clk_40MHz, clk_160MHz, reset_n, L1_trigger, enable, and raw_data. The 110-bits encoded_output and the data_buffer are for intermediary processing. Lastly, the encoded_output is for sending the data to the backend. Now, each one of these ports will be explained. The clk_40MHz and clk_160MHz are clock signals. The reset_n is used to control sending idle K28.5 words. If reset_n is 0, the encoded output of 110-bits will be all idle words rather than actual encoded data. If reset_n is 1, the encoded output of 110-bits will be the encoded bits corresponding to the 88-bits input raw

data. The L1_trigger input is used to control when to read raw data from the raw events data file. When the L1_trigger becomes 1, the input raw data will be read from the events data text file. When the L1_trigger is zero, the input raw data will stay the same since no more data will be read from the events data text file. The enable port is used to control the individual 8b/10b encoders. If the enable is zero, encoding will not be allowed. The raw_data is the raw data that is being read from the text file. The encoded_output is the encoded 110-bits that correspond to either idle words or the input raw data of 88-bits. The data_buffer is used to store the encoded 110-bits in 8-bits chunks. The encoded_output of 8-bits corresponds to the data that is being outputted from the buffer.

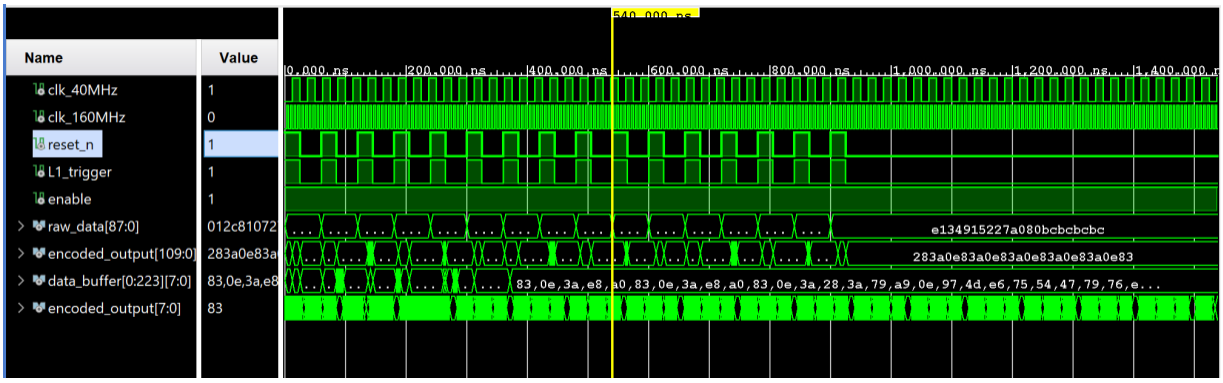

Figure 2: Only when the L1_trigger becomes 1, the raw data changes since it gets read from the raw data file. Also, when reset_n was zero for a long time, the encoded_output was always the same since it was the encoded idle K28.5 words.

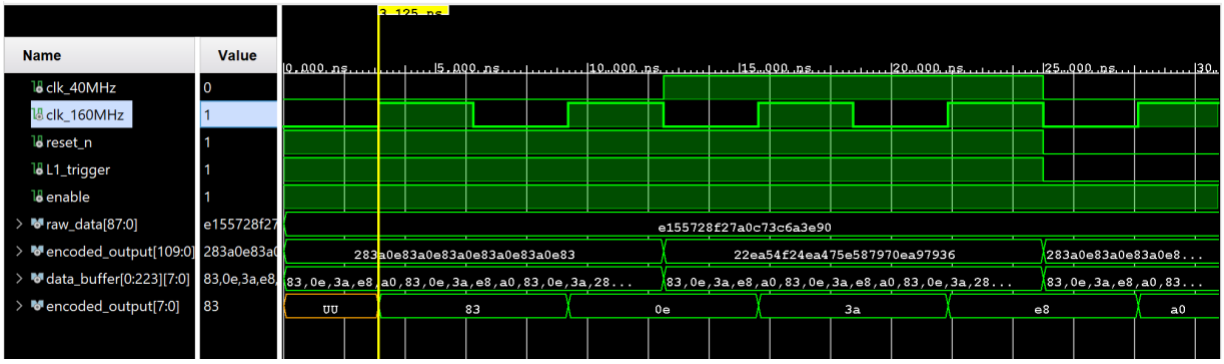

Figure 3: This image shows that the encoded data of 110-bits (283a0…0e83) were added to the buffer, and then the data were being outputted from the buffer in 8-bits chunks at 160MHz starting with the least significant bits x"83" and moving on to the more significant bits.

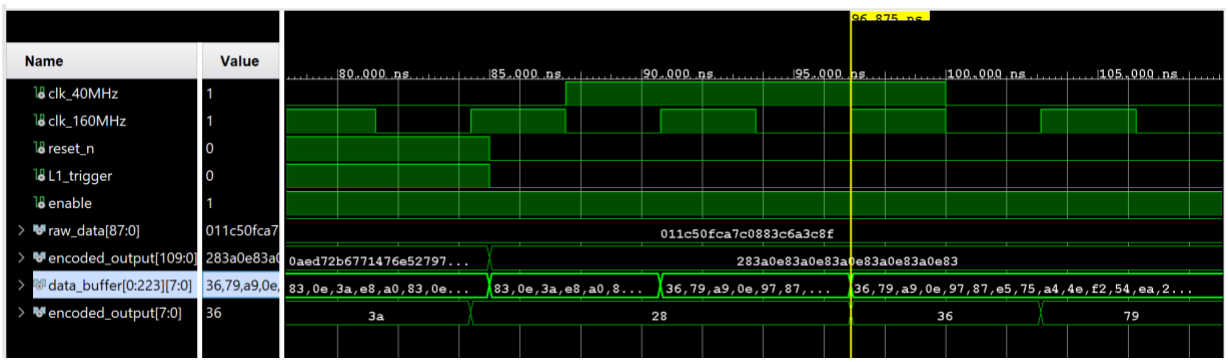

Figure 4: Once all the encoded data of 110-bits (283a0…0e83) were outputted, that data were removed from the buffer and the buffer started to output the next 110-bits of encoded data, which were (22ea54…a97936).

Conclusion

To conclude, the aim of this report was to show the TOFHIR2 data transmission process and cover the VHDL simulation that was done to simulate the TOFHIR2 data transmission process. Although the current simulation code has some bugs and limitations, it can provide a good starting point to make a more comprehensive TOFHIR2 data transmission simulator. This simulator can be used to understand how the data transmission parts of the TOFHIR2 ASIC function. Also, if this simulator was improved, it could be used to test how the actual TOFHIR2 data transmission parts would work before making the changes on the actual TOFHIR2 ASIC. The TOFHIR2 ASIC is an important part of the BTL electronics and the MTD, and so this project contributes to the ongoing efforts to upgrade the LHC for the high luminosity era.

References

[1] CMS Collaboration, "A MIP Timing Detector for the CMS Phase-2 Upgrade," CERN-LHCC-2019-003 ; CMS-TDR-020, ISBN: 978-92-9083-524-0, 2019. [Online]. Available: https://cds.cern.ch/record/2667167?ln=en