PhD Progress Report

Admission to the 2nd year

Simran Yadav

simran.yadav@studenti.unipd.it

Technologies for Fundamental Research in Physics and Astrophysics

XL cycle

Department of Physics and Astronomy "Galileo Galilei"

University of Padova

About me:

Name: Simran Yadav

Curriculum: Electronics

Annual Year: 2024-2025

Supervisor: Angelo Cotta Ramusino

Co-supervisor: Nicolò Vladi Biesuz

Research Topic: Development of photosensors and readout for the upgrade of the LHCb RICH detector for High Luminosity LHC

Education: Masters in Physics, University of Delhi

Masters Project: Probing dark matter through the Stochastic gravitational wave background

Research Project and Objectives

- My research is part of the R&D program aimed at enhancing the performance of the Ring Imaging Cherenkov (RICH) detectors of the LHCb experiment at CERN. This upgrade is essential to ensure that the RICH detectors maintain their particle identification efficiency under the significantly higher particle rates expected following the High-Luminosity LHC (HL-LHC) upgrade, scheduled for Long Shutdown 4 (LS4), currently planned for 2034.

- As an intermediate milestone, the front-end electronics of the RICH detectors will undergo a partial upgrade during Long Shutdown 3 (LS3), scheduled to begin in July 2026. This upgrade will implement a new readout chain for the presently installed Multianode Photomultiplier Tubes (MaPMTs). At the core of the new readout system is the "FASTRICH" ASIC—a custom chip developed at CERN to meet the demanding timing precision and high data-rate requirements of the upgraded environment. The first FASTRICH prototypes were delivered to CERN on May 7, 2025, and in August a set was mounted onto Front-End Boards (FEBs) developed at INFN-Ferrara.

- My initial research focuses on characterizing the FASTRICH ASIC and contributing to the development of a Quality Control (QC) system for the FASTRICH-based front-end boards (FASTRICH-FEBs).

- Subsequent stages of my research will involve applying the QC system—or alternative available readout systems based on the FASTRICH FEBs—to evaluate the performance of candidate single-photon counting devices proposed for the LHCb Upgrade.

QC system of FastRICH-FEB

The FASTRICH-FEB QC system will be based upon the “TEI1000-02-A3I11-A” module recently developed by Trenz Electronic GmbH and featuring an Altera 1SX040HH3F35I3VG Stratix 10 SoC FPGA with an integrated ARM processor.

The FASTRICH-FEB QC system will be built in a modular fashion, with each “TEI1000-02-A3I11-A” module managing, through ancillary control and interconnection cards, up to two FASTRICH-FEBs—comprising up to eight ASICs, each with four serial links operating at data rates up to 1.28 Gbps.

The tasks performed by each FASTRICH-FEB QC unit (“TEI1000-02-A3I11-A” module + ancillary card) will be:

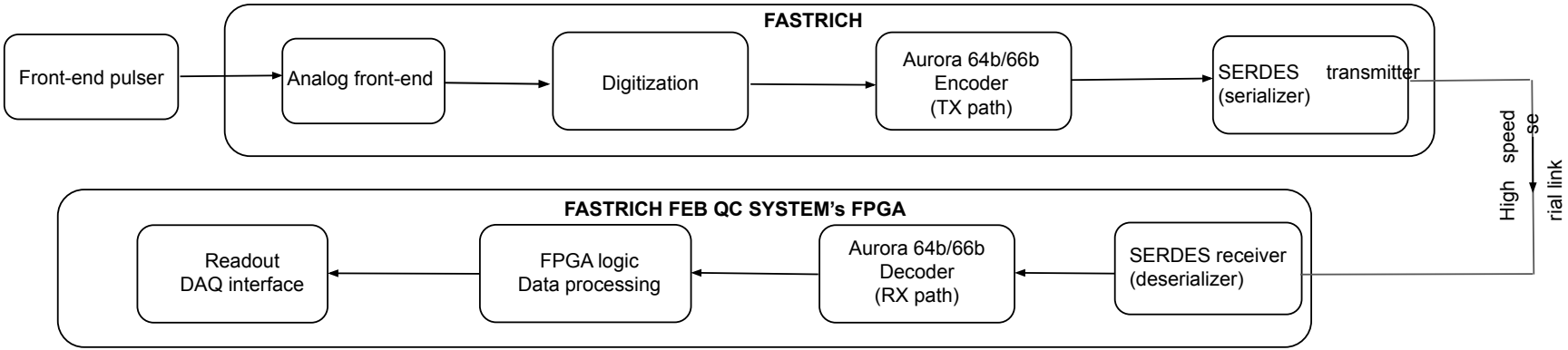
- Supplying bias voltages and monitoring current consumption;
- Generating “Fast Control” timing signals and stimuli to the FASTRICH-FEBs under test;
- Configuring the FastRICH ASICs through their I2C slow control ports;
- Deserialize, decode and monitor the signal integrity of the ASICs’ serial output links encoded according to the Aurora 64b/66b protocol;
- Inject charge at the FEB’s inputs and analyze the FASTRICH ASICs’ output to verify the full functionality of each FEB under test.

The implementation of the tasks performed by the FASTRICH-FEB QC system will require:

- Firmware development:
  - design of HDL module implementing the lower layers of the ASIC configuration and readout protocols.
- Software (SW) development
  - design and implementation of the interface layer between the ARM-side application and the FPGA fabric, managing configuration registers, data exchanges, and control/status monitoring;
  - development of the software running on the SoC’s ARM processor core.
- Control software (server)
  - design of the overall QC test structure, defining the sequence and logic of automated test routines (e.g., functionality checks, timing scans, error injection) to validate the FASTRICH-FEB performance
  - development of a user interface to launch tests, display live diagnostics, and log results for downstream analysis.

Overview of the high speed data path (Aurora based) for the QC system:

1. Analog front-end:
   - Photodetector's signal conditioning, amplification, shaping;
2. Digitization:
   - Digitized events include time of arrival, time over threshold, position;
   - control logic implementation;
   - event framing and buffering, error correction and alignment, data packing in Aurora format.
3. Aurora protocol layer:
   - Encodes framed data;
   - Provides clock compensation, channel bonding, flow control and error detection.
4. DAQ receiver:
   - FPGA or custom logic receives and decodes Aurora stream.
   - Reassembles data into events, sends to PC or PCIe.



1st year progress

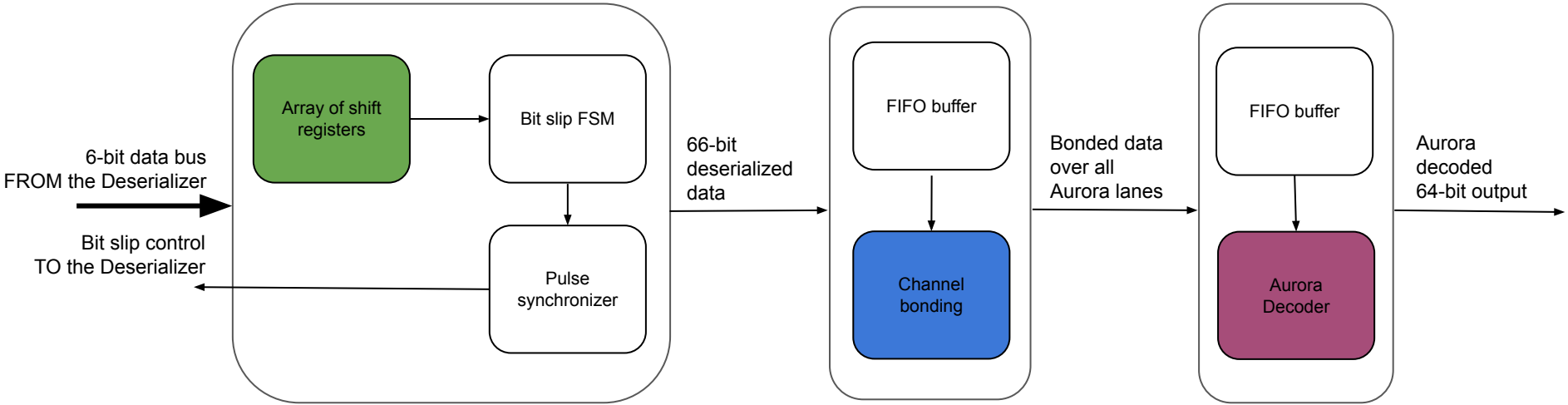
During my first year of research, I focused on understanding the features of the FASTRICH ASIC and developed the HDL modules targeted to the chosen ALTERA 1SX040HH3F35I3VG Stratix 10 SoC FPGA and implementing the following key functions of the FASTRICH-FEB QC system:

- ❏ locking and deserialization of serial data transmitted via FASTRICH output links at data rates of 320, 640, or 1280 Mbps;
- ❏ stripping of the AURORA 64b/66b framing protocol, used by the FASTRICH output serial ports, to ensure robustness against synchronization losses caused by serial bit errors;
- ❏ decoding of FASTRICH data packets, which encode digitized information for particle “hits”

To support development and validation, I made extensive use of a QuestaSim simulation framework provided by the LHCb back-end experts which allowed me to deepen my understanding of FASTRICH operation by following the data flow originated by running the simulation scripts. I am also using QuestaSim to test the HDL modules I implemented.

Synthetic descriptions of the HDL modules which I developed is following.

Synthetic descriptions of the HDL modules developed by me: Aurora protocol decoder



Synthetic descriptions of the HDL modules developed by me: Aurora protocol decoder

Ensures word-level synchronization across all lanes by detecting a common bond frame and buffering aligned data.

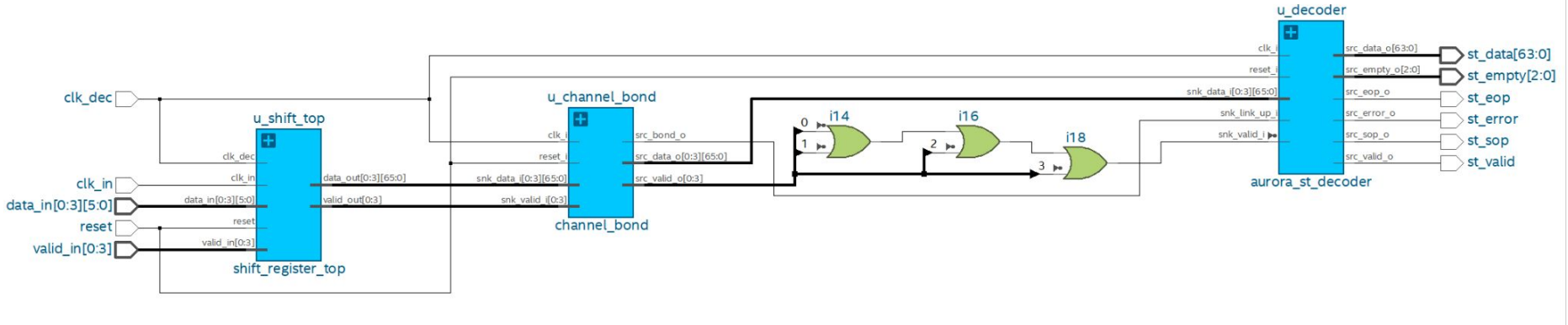
Aligns incoming data using bit-slipping and header detection across 11×6-bit registers to produce 66-bit aligned frames.

Extracts valid Aurora data packets (with SOP, EOP, EMPTY flags) from the bonded stream and outputs to an Avalon-ST interface.

Synthetic descriptions of the HDL modules developed by me: Aurora protocol decoder

Aurora receiver HDL module

- Implements the top-level receiver pipeline for Aurora protocol.
- Handles multi-lane 66-bit serial input aligned over 4 parallel lanes.
- For the integration we will be using SoC Module with Stratix® 10 SX 1SX040 plus ARM Processor, speed grade 3 (1SX040HH3F35I3VG).

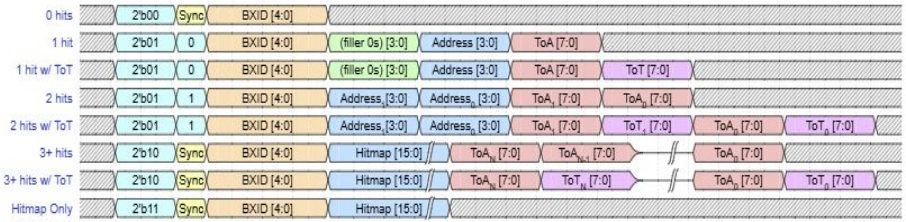


Synthetic descriptions of the HDL modules developed by me: Aurora protocol decoder

- Currently, I am working on the frame building to extract the information about the hitmap which includes BXID, Time of arrival (ToA), Time over threshold (ToT).
- A full orbit of LHC has 3564 bunch crossing and there are almost 11.2k LHC orbits per second.
- Each orbit can be expressed using 12-bits, but we can group similar events so we don't need to send the full BXID every time. So after first event for following events we send 5-bits (LSBs).
- Distance between subsequent events is 32 bunch crossing, after this we again send full 12-bit BXID.
- Bunch crossing happen at a rate of 40MHz.

Hitmap packet scheme:

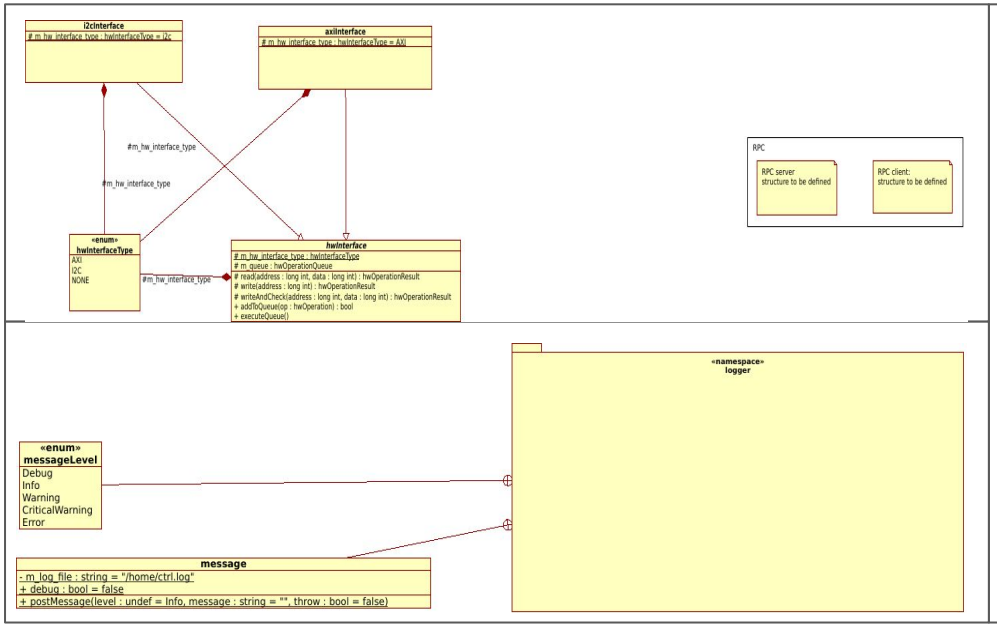
1. Total BXID (12 bits), representing one full orbit;
2. max. flag reporting the end of frame;
3. ToA and ToT;
4. gate start field, start of BX, ToA of hits;
5. Granularity of time bins for ToA changes.

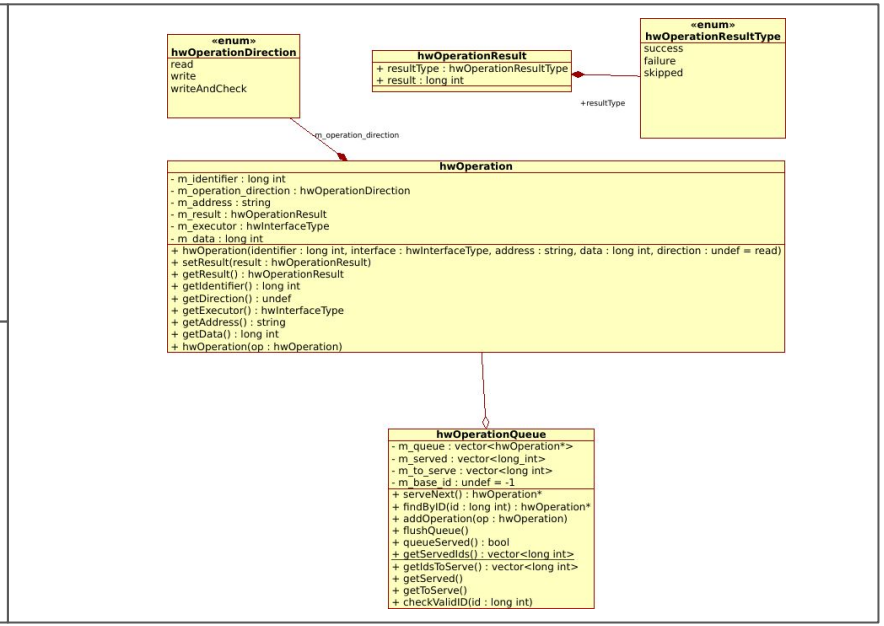


Packet Type	Size
0 Hits	8
1-2 Hits	16 + (2*1 + 1) * ((ToT?) ? 16 : 8)
3+ Hits	24 + $countones(hitmap) * (4 + ((ToT?) ? 16 : 8))
Hitmap Only	24

Software Operation and Control Architecture

- This firmware exploits the AXI interface to interact with processing units.
- Software (running on CPU) handles scheduling of test operations.
- From the firmware perspective, a test is a sequence of instructions in registers/memory, exposed to the CPU for execution.
- Test execution produces data over high-speed links, stored in a memory area accessible by the CPU.





PhD Courses

Courses completed:

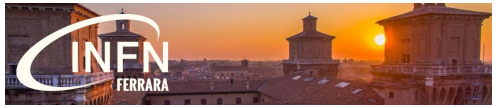
1. Design of Readout Integrated circuits for particle detectors- 2,5 credits
2. Programmable System on Chip (SoC) for data acquisition and processing- 4 credits
3. Advanced scientific programming in Matlab- 6 credits
4. Electronic systems in high energy physics - 4 credits
5. Fundamental of FPGA-based digital design- 2,5 credits

Courses on-going:

1. Advanced FPGA design and design management techniques- 2,5 credits

2nd year Research activities

- In my second year, research activities will focus on finalizing and commissioning a prototype FASTRICH-FEB QC unit.
- In the subsequent stages, I will use the QC system also to evaluate the performance of candidate single-photon counting devices under consideration for the LHCb Upgrade.



Thank You