Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communication Engineering
(37^{th}cycle)

Advanced digital readout and signal processing techniques for radiation sensors

By

Stefan Cristi Zugravel

******

Supervisor(s):
Angelo Rivetti, Supervisor
Manuel Dionisio Da Rocha Rolo, Co-Supervisor
Michele Goano, Co-Supervisor

Doctoral Examination Committee:
Prof. Kugathasan Thanushan, Referee, Geneva University
Prof. Gabrielli Alessandro, Referee, Università degli Studi di Bologna
Prof. Giordano Raffaele, Università di Napoli Federico II
Prof. Placidi Pisana, Università degli Studi di Perugia
Prof. Paternò Andrea, CERN

Politecnico di Torino
2025

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Stefan Cristi Zugravel
2025

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

“Cosa fate nella vita?
- Beh noi lavoriamo nella meccanica di precisione, tecnologie avanzate al servizio di progettazioni particolari e specifiche. Non lo so... Hardware... Cioè creiamo dei supporti che serviranno per progettare grosse situazioni, non so strumenti di precisione per una svolta futura magari della meccanica, non so se mi spiego...
- Sì, insomma, abbiamo un negozio di ferramenta. Cioè, non è che il negozio di ferramenta è il nostro, noi ci lavoriamo come commessi, come galoppini insomma.”
-Aldo, Giovanni e Giacomo

Acknowledgements

I would like to acknowledge Professor Angelo Rivetti for giving me the opportunity to undertake this path. My gratitude goes also to Professor Manuel Rolo and Professor Michele Goano which were always available whenever I needed advice.

I would like to acknowledge the entire ZDC team, in particular Nora De Marco, Pietro Cortese, Carlo Puggioni, Chiara Oppedisano and Luca Quaglia. Working for them (and with them) was a pleasure.

My acknowledgments also go to the Turin MIZAR group, in particular Mario Bertaina and Andrea Di Salvo. They consistently demonstrated an extreme respect for me and for my work.

I would like to thank all the colleagues who shared the office with me, in particular Stefano Durando, Greta Andrini, Chiara Ferrero, Umberto Follo, Francesca Lenta, Sofia Blua, Matteo Barbagiovanni and Valerio Pagliarino. Your presence made this journey much more enjoyable.

Lastly, but not by importance, I would like to acknowledge the continuous presence and support of my mother, Cristina Staviri, my father, Nicolae Zugravel and my significant other, Giorgia Rista. My deepest love for you helped me in the moments of discouragement.

Abstract

This thesis will present the work performed for two different and separated projects.

In the context of the ALICE experiment at CERN, the ZDC detector upgraded its readout system in order to be able to acquire data in all collisions in self-triggered mode without dead time at the expected luminosity offered by LHC in Run 3. The ZDC operating conditions in Pb–Pb collisions are extremely challenging due to the presence of ElectroMagnetic Dissociation processes that increase the need for high readout rate and the large signal dynamics (from a single neutron to ≈ 60 neutrons). The new acquisition chain is based on a commercial 12 bit FMC digitizer mounted on a commercial VME carrier board. The signals produced by the ZDC channels are digitized, the samples are processed through an FPGA that, thanks to a custom trigger algorithm, flags for readout the relevant portion of the waveform and extracts information such as timing, baseline average and event rate. When running in self-triggered mode the ZDC system is able to sustain a readout rate of ≈ 2.5 Mevents/s not only meeting the bandwidth request, but taking into account a safety factor of 2, for the channels of the most exposed calorimeters. My contribution consisted of the development of the FPGA readout firmware used for data taking, of the commissioning of the hardware at CERN and of the constant supervision of the data taking stability and performance of the ZDC detector during Pb–Pb collisions. The architecture of the detector and, more specifically, of the new readout system will the presented in chapter 2. Thanks to the new Continuous Readout Mode and the LHC increased luminosity, the ZDC detector was able to acquire more events than during Run 1 and Run 2 combined. A report of the data taking periods, the performance of the new readout system and a showcase of the acquired data for each period and data taking condition will be presented in chapter 3. It will be shown that the energy spectra resolution improved w.r.t. the previous readout system. Run 1 and Run 2 had a resolutions of the 1 neutron signal of about 20%, meanwhile, during

Run 3, the detector constantly achieved 16%. The ZDC time resolution allows the ALICE data analysis team to more easily reject parasitic collisions.

In the context of the NUSES collaboration and the PBR space mission, the INFN microelectronics team undertook the realization of the MIZAR ASIC. This 64-channel custom IC built in 65-nm CMOS technology designed for readout of SiPMs signals is expected to be used as a technology demonstrator for the use of such detector in space for the study of Extensive Air Showers produced by Ultra-High Energy Cosmic Rays and Ultra-High Energy tau Neutrinos. The integrated ADC is configurable to operate with a resolution between 8 and 12 bit at a 200 MHz sampling rate. The power budget of the system is 5 mW/channel. Given the extreme data bandwidth constraints due to the earth-space link, an elaborate multi-layer trigger system was implemented into the ASIC in order to reject as much noise as possible without affecting the data. This resulted in a configurable double threshold for each channel and a hitmap generator. The resulting hitmaps are sent to an FPGA for real time evaluation of the patterns. The configuration of the ASIC parameters, the hitmap trigger decision and the data packet formation are all managed by a readout FPGA. My contribution for the collaboration consisted of the complete development of the FPGA readout firmware, i.e. of the design of the hitmap evaluation algorithm, the configuration logic and the data handling machinery. This will be presented in detail in chapter 4. The performance of the designed readout system were asserted with simulations and in laboratory by means of an ASIC emulator developed with a second FPGA board. All the performed tests and the results obtained will be discussed in chapter 5.

Contents

Nomenclature

Acronyms / Abbreviations

ADC Analog to Digital Converter

ALICE A Large Ion Collider Experiment

ALR ALICE Trigger Rate

ASIC Application Specific Integrated Circuit

ATB ALICE Trigger Board

ATLAS A Toroidal LHC ApparatuS

BC Bunch Crossing

CBL Configurable Logic Block

CDC Clock Domain Crossing

CERN European Organization for Nuclear Research

CMOS Complementary Metal-Oxide-Semiconductor

CMS Compact Muon Solenoid

CPU Central Processing Unit

CPV Charge Particle Veto

*CR*4 Counting Room 4

CRU Common Readout Unit

CTP Central Trigger Processor

CTS Central Trigger System

DAQ Data AcQuisition

DCS Detector Control System

DPRAM Dual Ported Random Access Memory

EAS Extensive Air Shower

ECC Error Correction Code

ECS Experiment Control System

EM ElectroMagnetic

EMCal ElectroMagnetic Calorimeter

ENOB Effective Number of Bits

EPN Event Processing Node

*EUSO − SPB*2 Universe Space Observatory-Super Pressure Balloon 2

FBK Fondazione Bruno Kessler

FEC Forward Error Correction

FEE Front End Electronics

FIT Fast Interaction Trigger

FLP First Level Processor

FMC FPGA Mezzanine Card

FPA Focal Plane Assembly

FPGA Field Programmable Gate Array

FSM Finite State Machine

GBT GigaBit Tranciever

GEM Gas Electron Multipliers

INFN National Institute for Nuclear Physics

IOB Input Output Block

IP Interaction Point

ITS Inner Tracking System

LHC Large Hadron Collider

LHCb Large Hadron Collider beauty

*LS*2 Long Shutdown 2

LTU Local Trigger Unit

LVDS Low Voltage Differential Signaling

MAPS Monolithic Active Pixel Sensors

MCH Muon Chamber

MFT Muon Forward Tracker

MHPID High Momentum Particle Identification Detector

MID Muon IDentifier

MIZAR Multi-channel Integrated Zone-sampling Analogue-memory based Read-out

MRPC Multi-gap Resistive Plate Chamber

MWPC Multi Wire Proportional Chamber

NRE Non Recurring Engineering

NUSES NeUtrino and Seismic Electromagnetic Signals

*O*2 Online Offline

PBR POEMMA Balloon with Radio

PHOS PHOton Spectrometer

PLL Phase Locked Loop

PMT Photomultiplier

PSM Programmable Switch Matrix

QGP Quark Gluon Plasma

RDH Raw Data Header

RPC Resistive Plate Chamber

RTL Register Transfer Level

SEB Single Event Burnout

SEE Single Event Effect

SEGR Single Event Gate Rupture

SEL Single Event Latchup

SET Single Event Transient

SEU Single Event Upset

SFP Small Form-factor Pluggable

SiPM Silicon Photo-Multiplier

SoC System on Chip

SOI Silicon On Insulator

SPS Super Proton Synchrotron

SSMC Sub-Miniature version C

SWT Single Word Transfer

TCS Thermal Control System

TID Total Ionizing Dose

TMR Triple Modular Redundancy

TMRG Triple Modular Redundancy Generator

TOF Time Of Flight

TPC Time Projection Chamber

TRD Transition Radiation Detector

TTS Trigger and Timing System

TTYPE Trigger Type

UHECR Ultra-High Energy Cosmic Ray

UHENU Ultra-High Energy tau Neutrino

VME VERSABUS Module Eurocard

XTMR Xilinx Triple Module Redundancy

ZDC Zero Degree Calorimeter

ZN Neutron Calorimeter

ZP Proton Calorimeter

Chapter 1

Introduction to digital readout for particle physics experiments

This chapter will briefly describe what a detector for particle physics is (section 1.1) and what are the main challenges for its readout (section 1.2). In section 1.3 a brief overview of the main challenges of operating integrated circuits electronics under ionizing radiation is presented. In section 1.4 the Field-Programmable Gate Array (FPGA) device is introduced.

1.1 An introduction to radiation detection

From the end of the XIX century scientists from all over the world started conducting experiments involving radiation. In 1897 British scientist J.J. Thomson with his team conducted experiments with cathode tubes which led to the discovery of the electron [1]. In 1911 scientist E. Rutherford formulated the 'planetary theory' of atoms, as opposed to the 'plum pudding theory' commonly accepted at the time [2]. Another great example is the Bohr experiment of 1913 which led the scientist to the publication of the, now obsolete, Bohr model [3]. Detectors used by physics experiments performed during late XIX and the majority of XX century can be roughly divided into two main categories [4]:

- imaging detectors i.e. cloud chamber and bubble chamber photograph:
  which historically used special photographic film to impress the path of the

particles into paper. The images created were then analyzed and counted by humans. Those systems have excellent position resolution and the fact that target and detecting volume are the same makes the bubble chamber almost unbeatable for reconstruction of complex decay modes. The drawback of those chambers is the low rate capability (a few tens/second) and the fact that it cannot be triggered selectively means that every interaction must be photographed. In addition, analyzing millions of images by 'operators' was a quite laborious task. Photographic emulsions with automated track search are still used in modern systems nowadays.

- electronic readout logic detectors i.e. Geiger counters, scintillators:
  which used physics phenomena like the drift of electrons caused by the ionization of a gas mixture to produce a discharge measurable by an electroscope [5] or the scintillation of materials on passage of particles. The signal produced could then be processed with electronic gates and used, for example, for counting purposes. An example of such device can be the Geiger–Müller tube, the Multi-Wire Proportional Chamber (MWPC) [6] and the scintillation counters.

In the 70's, thanks to the advancement in electronic and computing power, those two categories (imaging and logic detectors) merged together to form 'Electronics Imaging Devices'. In figure 1.1 a Pb–Pb event recorded at the A Large Ion Collider Experiment (ALICE) experiment at CERN is shown. The data generated by hundreds of thousands of channels is collected, decoded and used to generate an image displaying all the recorded tracks; all by means of computers, electronic devices and specialized software.

1.2 Detector readout overview

When designing an experiment for particle physics, generally a lot of effort is put into making the detecting part as thin as possible, using as few material as possible. This is done in order to minimize the material budget and therefor the interactions between detector and the particle that is being measured. In this context the electronics present on the detector itself usually manage only basic signal amplification and the digitization of the data. Most of the modern detectors have external Front End Electronics (FEE) modules which manage the data flow and do more complex

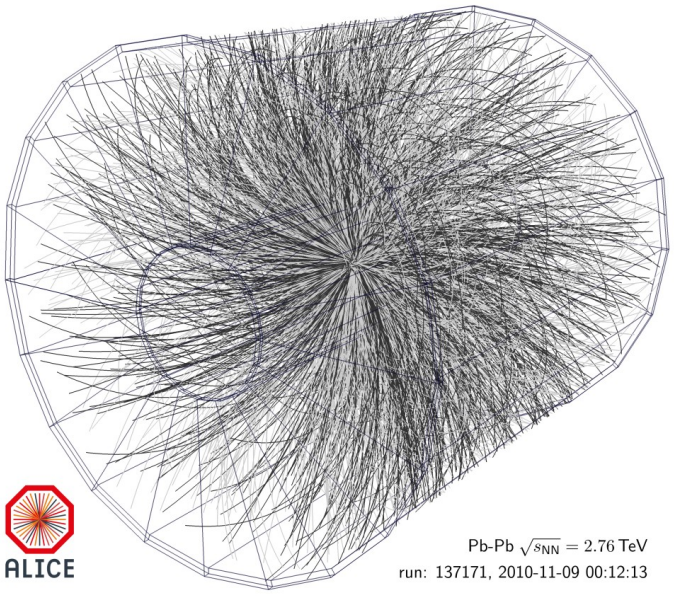


Fig. 1.1 ALICE Event Display electronically generated image. Figure taken from [7]

operations such as elaborate trigger detection, zero suppressing, data compression, track reconstruction and data interface management. In figure 1.2 a simple diagram depicting a common structure for a particle physics experiment detector can be seen. In the image four main block are depicted:

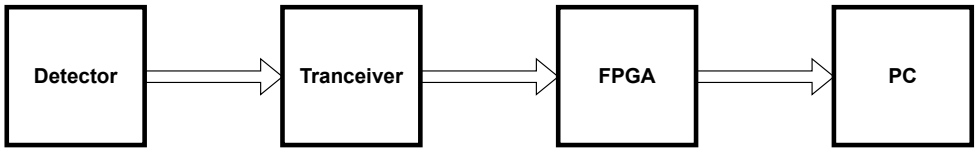


Fig. 1.2 Coarse block diagram of an hypothetical detector.

- the detector is the sensitive part of the experiment. Usually it is designed leveraging specific material properties, triggered by particle interactions, in order to transform a physical phenomena in a, in most cases electrical or optical, signal. This can be, for example, a Photomultiplier tube which is fed light by a scintillator [8], a Silicon Photomultiplier [9] or a Resistive Plate Chamber [10, 11].

- the analog signal, in order to be processed by a computer, must be transferred into the digital domain. For experiments with a large amount of channels or

with extremely tight power and space constraints, usually Application Specific Integrated Circuitss (ASICs) are developed 'ad hoc' for the particular use case and signal characteristics. An example of such architecture that is currently being developed will be presented in detail in chapter 4. In modern state of the art detectors, the sensitive volume and the transceiver electronics can be combined into a single wafer of silicon. This is called a monolithic design [12]. An example of such technology is the ALPIDE pixel sensor chip used for the upgrade of the ALICE Inner Tracking System [13].

- usually the digitized data of multiple channels (or ASICs) is sent to a FPGA based readout unit (or module) where it can be sorted, processed and sent to a computer for further analysis. This is considered the best approach for three main reasons:
  - designing, verifying and producing an ASIC capable of elaborate data processing is extremely complex, time consuming and expensive. From an architectural point of view it is much more efficient to cluster those calculations in a programmable multi purpose device that can be obtained commercially.
  - generally the FEE, being in close proximity to the detector itself, is exposed to radiation. This can cause a wide range of faults which will be discussed in section 1.3. Designing a chip capable of operating in a radiation environment requires much more designing effort, chip area and production cost. For those reasons, usually, the readout ASIC are designed with only the minimum necessary features for digitizing the events and the complex calculation are performed on different modules further away from the radiation.
  - when designing the experiment itself a lot of studies goes into minimizing the material budget of the detector. The observed particles should not be deviated or slowed down by the FEE. For this reason it is always preferable to put the bulk of the electronic outside the path of the particles. In addition this approach makes easier to design a simpler power and cooling infrastructure for the readout system.

  Example of this type of approach are: the readout of the upgraded ALICE-ITS [14], the readout of the new ALICE-TPC [15], and the readout system of the ATLAS tracker [16].

- the data is then finally sent to a computer were it can be reconstructed, analyzed and stored for further processing. The amount of data produced by an experiment can have huge variations. On one hand, detectors that were designed for example for space applications usually have a low data throughput, in the order of the tens of Gb per day due to the limitations in the satellite-hearth communication. On the other hand, the enormous experiments which are currently operating at CERN are producing data in the order of hundreds of GB per second, each.

If the detector has a relatively low number of channels (a few hundreds or less) or it is being developed as a prototype or a technology demonstrator the cost, time and complexity overhead, of designing and implementing a custom ASIC can be prohibitive. In this scenario the most cost effective approach is often to use a completely commercial solution which combines a digitizer and an user programmable logic (FPGA or CPU) in a modular form factor. Examples of such products can be the DT2730 from CAEN s.p.a. [17] or the ADQ214-DCLN from TELEDYNE sp devices [18].

1.3 Challenges of operating electronics in a radiation environment

When a charged particle passes through a material it can knock loose thousands of its electrons, this is called ionizing radiation [19]. Modern electronic circuits rely on microscopic electrical components which drive (or are driven) by and incredibly tiny amount of current. It is then possible that a charged particle passing through a CPU, memory cells, FPGA chip, logic gate or some sort of ASIC can cause a change in the state of the logic which can led to a temporary or even permanent damage of the device. The higher is the amount of radiation the more probable are this kind of events. For example, at plane cruising altitude the cosmic radiation levels are higher with respect to sea level. For this reasons the electronics circuits of planes are designed to be able to sustain the exposure to a radiation environment. On the contrary, common items such as personal computers, cellphones and cars do not need this kind of protection.

1.3.1 Effects of radiation on electronic circuits

In this subsection a brief list of the possible effects caused by ionizing radiation on electronic devices is presented.

- Cumulative effects; when the integral of the radiation is directly proportional to the damage, this can be divided into:
  - Bulk Damage: damage caused by the displacement of one or more crystal atoms by the interaction of the incident particles with the nuclei of the atoms of the lattice.
  - Total Ionizing Dose (TID): degradation of the performance of the irradiated devices due to the homogeneous accumulation of charge in the oxide layers and interfaces of the electronic components.

  TID degradation is usually measured in terms of a time or dose until failure.

- Single Event Effects (SEEs) are individual events which occur when a single incident ionizing particle deposits enough energy to cause an effect in a device. SEEs could be categorized into soft errors and hard errors. Soft errors are nondestructive to the device and may appear as a bit flip in a memory cell or latch, or as transients occurring on the output of an I/O, logic, or other support circuit. Hard errors may be physically destructive to the device, and, most importantly, are permanent functional effects [20]. The probability of a SEE is strongly correlated with the technology node used for the chip. In table 1.1 two parameters related to SEE are shown with respect to the technology used. The critical charge is the amount of deposited charge needed to change the logic state of a cell. The sensitive geometry is the volume in which the deposited charge is effective to generate a perturbation in the device. In addition, the number of elements (transistors) and the complexity of the design take a crucial role in the probability of a SEE. SEEs are usually measured in terms of a probability that an event will happen within a fixed time window.
  - Permanent (hard error):
    - Single Event Burnout (SEB) is a condition which can cause device destruction due to a high current state in a power transistor.

Technology node [nm]	Sensitive volume [μm^3]	Critical charge [fC]
250	0.245	8
130	0.025	2.5
90	0.02	0.2
65	0.0035	0.8

Table 1.1 Sensitive volume and Critical charge of a design with respect to the technology node used for the production.

    * Single Event Gate Rupture (SEGR) occurs when a single ion induced condition in power MOSFETs which may result in the formation of a conducting path in the gate oxide.
    * Single Event Latchup (SEL) is a condition which causes loss of device functionality due to a single event induced high current state. An SEL may or may not cause permanent device damage, but requires power strobing of the device to resume normal device operations [21].

– Transient (soft error):
    * Single Event Transient (SET) is a soft error in that a reset or rewriting of the device causes normal device behavior thereafter.

– Static (soft error):
    * Single Event Upset (SEU); it occurs in storage elements when the charge collected from an ion interaction exceeds the critical charge required to upset the circuit. The circuit then changes state, and the stored information gets lost. However, the circuit still functions normally, and it can be restored to its original operating state by rewriting or re-initializing the logic [22].

1.3.2 Common type of mitigation

This section will briefly delve into the most common approaches used to design electronics capable of sustaining operation in a radiation exposed environment. It is worth noting that all of the potential SEE mitigation methods may require that either additional hardware or software be added to the system design and architecture.

The complexity and, in most cases, the increase in system overhead caused by the additions are fairly linear with the power of the mitigation gained [23]. Several mitigation techniques can be applied across different design levels. In this section the technology, cell, and system levels will be presented.

- Technology level
  - Minimize the sensitive depth: reducing the sensitive depth in components, such as by using Silicon-on-Insulator (SOI) [24] technology, helps decrease the sensitive volume of the component.
  - Use a bigger technology node: when dealing with radiation and electronics not always smaller is better. In fact it is the opposite. Most radiation hard chips produced today use fabrication nodes that are considered obsoleted in consumer grade devices.
- Cell level
  - Increase the number of substrate contacts: adding more substrate contacts improves the charge dissipation, thus reducing the impact of SETs.
  - Surround the critical areas with guard rings: guard rings can be used to isolate critical circuit areas, thus providing protection by collecting charges from ionizing particles and preventing them from reaching sensitive cells.
  - Increase the node capacitance: this can help to absorb the charge deposited by radiation, thus reducing the risk of transient errors.
  - Store the information on multiple nodes, in this way the impact of a single fault can be mitigated as an error is not likely going to corrupt the data in all nodes at the same time.
- System level
  - Power-Cycling: any time an electronic device is power-cycled, all of the data stored in its cells are refreshed (if not designed with static memory). This means that a power cycle is the fasted, cheapest and simplest way of performing a complete system reset. If the system can withstand a downtime period and the expected dose is low this mitigation can sometimes be enough.

- Encoding: Error Correction Codes (ECC) allow detection and correction of certain radiation-induced errors [25]. In computer science, telecommunications and electronic finite state machine logic, hamming codes allows the detection and correction of one-bit and two-bit errors. When transmitting data usually 8b10b and 64b66b encoding are used to balance in DC the transmission line, to allow for clock recovery and to detect single-bit errors. To be noted that 8b10b and 64b66b do not allow error correction.
- Triple Modular Redundancy (TMR): this involves replicating critical components three times and implementing a majority voting logic in order to ensure correct operation even if one component experiences an error. The replication can be performed at different levels. The most robust configuration is to have three identical devices (ASICs or FPGAs) configured in the same way, the output of which is compared by and external radiation resistant majority voter. This approach, however, is also the most power hungry, costly, and area aggressive. Not considering the overhead added by the additional complexity in the PCB design. Another solution is to use one single device, but with all the logic replicated internally and with an integrated majority voter. In most cases this also means replicating all clock and reset nets. When doing so it is good practice to introduce a phase shift between the different clock edges in order to achieve a greater resilience to SET, since the bits are latched at different instants. Replicating all the logic is not always feasible, an ASIC can be area constrained, while FPGAs have a limited number of usable resources. In this case a solution could be to replicate only the most critical logic of the design and leaving everything else as is. For FPGA firmware design vendors offer specific tools to help the engineers implementing this kind of solutions. For examples XILINX provides the Xilinx Triple Module Redundancy (XTMR) software tool which can partially or fully triplicate a design, insert voters and allow synchronized feedback path loops. At the same time CERN developed the Triple Modular Redundancy Generator (TMRG) [26]; a tool-set designed to assist in the process of creating digital designs immune to single event upsets.

- Checkpointing: it involves storing periodic checkpoint states which enables the system to recover to a known good state in case of an error.
- Scrubbing: when operating in a radiation environment errors can happen not only in the FSM registers, but also in the S-RAM configuration memory of the device. This second case is harder to mitigate, since a complete knowledge of the correct memory state and the protocols to access it are necessary to perform any action. Modern systems have additional logic that, in background during normal operation, periodically inspect the memory (or part of it) for any error. This operation is called scrubbing. As for TMR, scrubbing can be implemented in different flavors and levels of complexity. The more sophisticated and robust approach is to have an external, radiation resistant device that performs the start-up of the chip and reads back its memory periodically. The data is then compared with a safe copy kept in a radiation resistant memory. If an error occurs only the relevant memory block would be re-written. Another option is the so called "blind scrubbing", which consist in the periodical re-configuration of the device without checking for errors. This second approach is much cheaper and simpler to implement since it does not requires additional external components, but it is less resistant to faults. In addition, given that the configuration phase is usually the most delicate during normal operation, this repeated procedure could led to additional errors. From a certain point of view a power-cycle could be considered as a light form of scrubbing [27, 28].

TMR, encoding, scrubbing and checkpointing can always be used in combination. The truth is that a golden rule which can be applied to all systems currently does not exists and most probably it never will. Each use case is unique and the engineers should carefully find the best mitigation scheme answering questions as: How much the application is tolerant to errors? For example an imaging capturing device for earth topography is more more tolerant than the International Space Station flight computer. What is the required time window of operation? This can span from a few minutes up to several years. What are the performance, power and cost requirements? The application can withstand downtime? If yes, how much? In figure 1.3 a mitigation scheme table overview is presented [29].

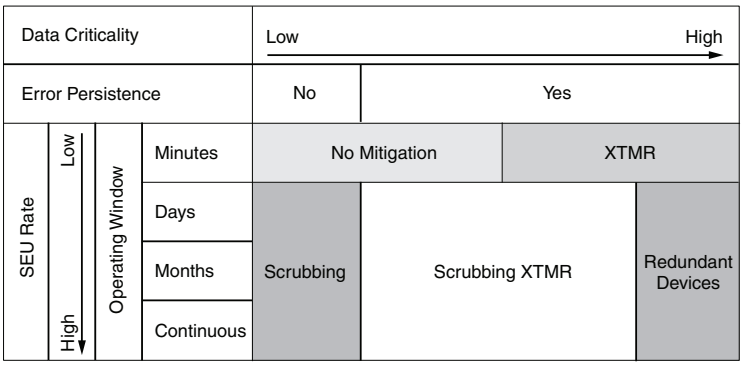


Fig. 1.3 Mitigation scheme table overview. To be noted that this table does not take in account neither power consumption, cost or performance evaluation. Figure taken from [29]

1.4 Introduction to FPGA devices for readout applications and Firmware

In figure 1.2 it was shown that, before the final processing step that involves a computer, most complex detectors use a readout module(s) which takes advantage of a FPGA chip for data processing. This section will briefly describe what this kind of device is, whats is purpose and what are its most relevant advantages and disadvantages.

1.4.1 Introduction to FPGA

A Field-Programmable Gate Array is an integrated circuit which is designed to be configured by the user after manufacturing a, theoretical, infinite amount of times, allowing thus a great flexibility to reprogram the hardware for different tasks. FPGAs were invented in the 1980s, with the first commercially available FPGA introduced by Xilinx in 1985, the XC2064, shown in figure 1.4. FPGAs consist of a matrix of

Fig. 1.4 XC2064 FPGA DIP chip. Figure taken from [30]

Configurable Logic Blocks (CLBs) which are interconnected by a Programmable Switch Matrix (PSM) which can be though as a configurable routing network and

interfaces with the outside world using Input/Output Blocks (IOBs). The resulting device can be configured to implement complex logic functions, processing tasks, or even entire systems-on-chip (SoCs). FPGA are widely used in telecommunications, aerospace and defense sectors, data centers, AI applications and, context of this thesis, data acquisition systems for scientific research.

1.4.2 Advantages and disadvantages

In this subsection a list of the main advantages and disadvantages will be presented.

- advantages:
  - unlimited programmability and blazing fast prototyping: this kind of device is excellent for a quick developing and testing phase. The operating logic of the device can be updated or completely changed by 'simply' modifying the Register Transfer Level (RTL) code on a computer and re-implementing it.
  - reduced power and better performance for custom tasks: when optimized for specific tasks (data concentrator, logic operations...), FPGAs can be much more power-efficient and quick than general-purpose processors like CPUs.
  - availability of huge IP and module libraries which allow to implement complex transmission protocol such as PCIe, AXI and 64b66b transmission lines at Gb speed over fiber in a matter of hours.
- disadvantages:
  - higher power consumption and slower performance compared to ASICs: FPGAs generally consume more power with respect to an ASICs built for the same application and, due to more complex routing and some additional logic overhead, the maximum performance achievable is generally lower.
  - larger physical size and Footprint: FPGAs generally have a larger die area than ASICs, this could be an issue for extremely space constrained designs.

– High Cost for High-Volume Production: On the one hand ASICs have usually a large Non-Recurring Engineering cost (NRE), which includes the cost of the design, the simulations and, eventually, the prototyping. Once all the development is done, the per-unit cost of a chip is very low. On the other hand, for the end user, the per-unit cost of a commercial product like an FPGA is much higher with respect to ASICs. The total price, and thus the final choice between ASIC and FPGA can be roughly summarized with a graph like the one shown in figure 1.5, where it can be seen that at a large enough scale the ASIC is always more cost effective. However, if the number of required chip is below a Break-Event point, the FPGAs are most cost effective.

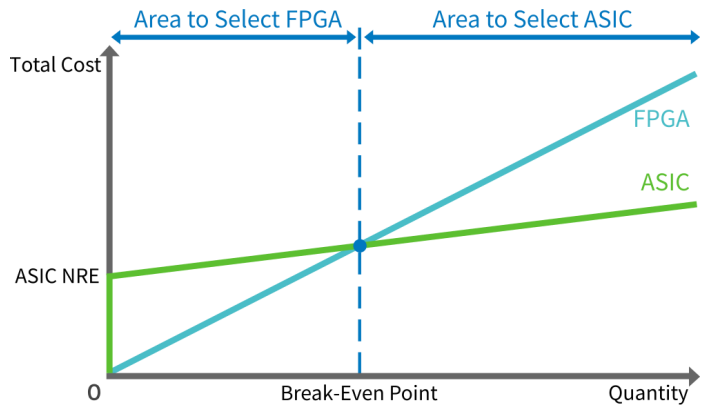


Fig. 1.5 Cost vs Quality diagram showing the point at which ASICs became more cost effective with respect to FPGAs. Figure taken from [31]

1.4.3 FPGA used in a radiation environment

As an ASIC can be designed in order to be withstand operation in a radiation environment, the same principles apply to FPGAs. In particular three approaches can be considered. The first one is to use Radiation-Hardened FPGAs. This kind of devices are built using materials and processes (SOI technology for example) that make them resilient to the effects of ionizing radiation. They are primarily used in space missions such as satellite systems, deep-space probes, and other aerospace applications. An example of this kind of device is the Microchip RTG4 [32] and the Xilinx Space-Grade UltraScale [33]. The second approach is to use Radiation-Tolerant FPGAs. With respect to the Rad-Hard types those are less robust and

only designed to withstand a moderate radiation exposure at the same type they are way less expensive. They are generally used for ground-based nuclear facilities, high-altitude aircraft, and medical imaging devices. The third and last approach is to use a normal consumer grade chip, but with mitigation techniques implemented into the logic as TMR, ECC and Checkpointing. The cost of the device in this case is the same of a consumer grade FPGA, the only overhead is added into the firmware design process. This approach is generally used for non critical applications where the total cost is highly constrained.

Chapter 2

The new readout system of the ZDC detector

This chapter begins with a short description of the complex of accelerators of the CERN (European Organization for Nuclear Research) and the ALICE (A Large Ion Collider Experiment) experiment in sections 2.1 and 2.2. The ZDC (Zero Degree Calorimeter) detector and its readout system will be analysed in more detail in section 2.3. Section 2.4, the most important of this chapter, contains my main contribution to the ALICE experiment, thus the description of the new FPGA readout firmware of the ZDC detector.

2.1 The CERN complex

The European Organization for Nuclear Research [34] was founded in 1954 and is located at the Franco-Swiss border near Geneva. At CERN physicists and engineers from all over the world use the world's largest and most complex scientific instruments ever built to study the basic constituents of matter. In order to do so particles are accelerated close to the speed of light and are then collided with each other or with fixed targets. Sets of particle detectors built on purpose are used to record the results of these collisions. Those particles, generally protons and lead nuclei, are accelerated by means of an elaborate system of linear and circular machines [36] which, step by step, increase they energy before injecting it into the next machine in the sequence. In the Large Hadron Collider (LHC), the last element in this chain

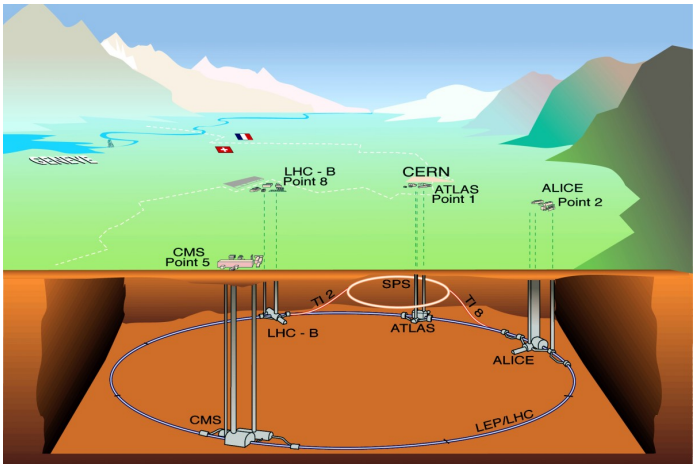


Fig. 2.1 LHC map diagram in 3D; Lake Leman and the city of Geneva are on the left, while the two main accelerator rings, SPS and LHC, can be seen in orange and blue. Figure taken from [35]

with a circumference of about 27 Km, particle beams are accelerated up to the record energy of 6.8 TeV per beam (during proton operations). In figure 2.1 the LHC ring, lake Leman and the city of Geneva are shown. All the controls for the accelerator are located at the CERN Control Centre, in the Prevessin site, on French territory [37]. The beams circulating in the LHC are put in collision in four locations around the accelerator ring, corresponding to the positions of the four main particle detectors: ATLAS [38] (A Toroidal LHC ApparatuS), CMS [39] (Compact Muon Solenoid), ALICE [40] and LHCb [41] (Large Hadron Collider beauty). In figure 2.2 a diagram of the accelerator complex and the experimental areas are shown.

2.1.1 The LHC orbit structure

In the LHC particles travel close to the speed of light, this means that an orbit around the 27 Km circumference takes 89.1 μs. The radiofrequency of the accelerator divides the orbit period into 35640 virtual positions in which particle transport is possible called buckets. Only 1/10 is usable, resulting into 3564 locations separated by $\approx$ 25 ns. Each bucket can hold a bunch of particles or can be empty. The bucket filling configuration will determine where the protons or nuclei in the two beams will cross over and collide and thus in which detectors specific bunches will be colliding. Those buckets are named Bunch Crossings (BC) [42] and, along the orbit period can have colliding beams, transiting beams or be empty.

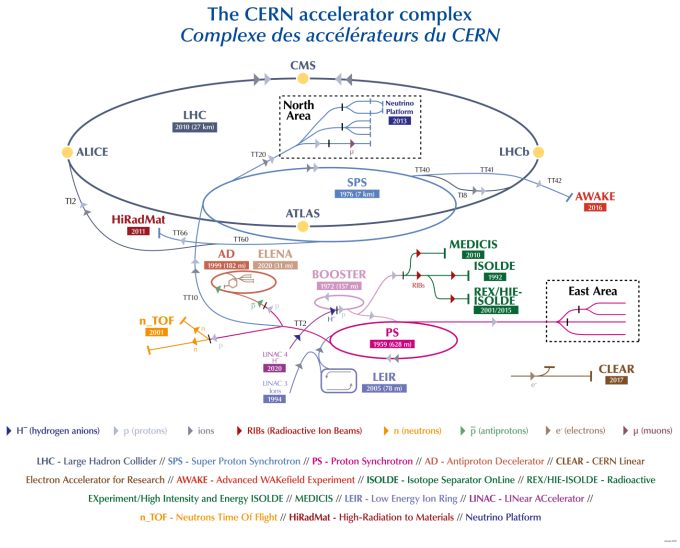


Fig. 2.2 The CERN accelerator complex diagram as of 2022. Figure taken from [36]

2.2 The ALICE experiment

The ALICE experiment [40, 43] is one of the four main experiments at CERN. The detector is specially designed to study heavy-ion (^{208}Pb) collisions at ultra-relativistic energies. The aim is to study and characterize a state of nuclear matter called Quark-Gluon-Plasma (QGP), in which quarks and gluons are not confined in hadrons [44]. Figure 2.3 shows a diagram of the ALICE detectors as before Run 3 [43]. The experiment is located at interaction point 2 of LHC in an underground cavern 56 m below the surface [45]. The detector is 26 m long, 16 m wide and 16 m high; the shape is similar to a cylinder (the so called central barrel), where the beams come from the two flat sides; On one side, called 'side C', an extension of the main detector allows to measure muons, the other side is named 'side A'. The experiment has a weight of about 10^7 Kg (more than the Eiffel Tower). ALICE can be divided into a central barrel and a forward region.

2.2.1 Central barrel detectors

The detectors of the central barrel are placed around the interaction point and include the following subsystems:

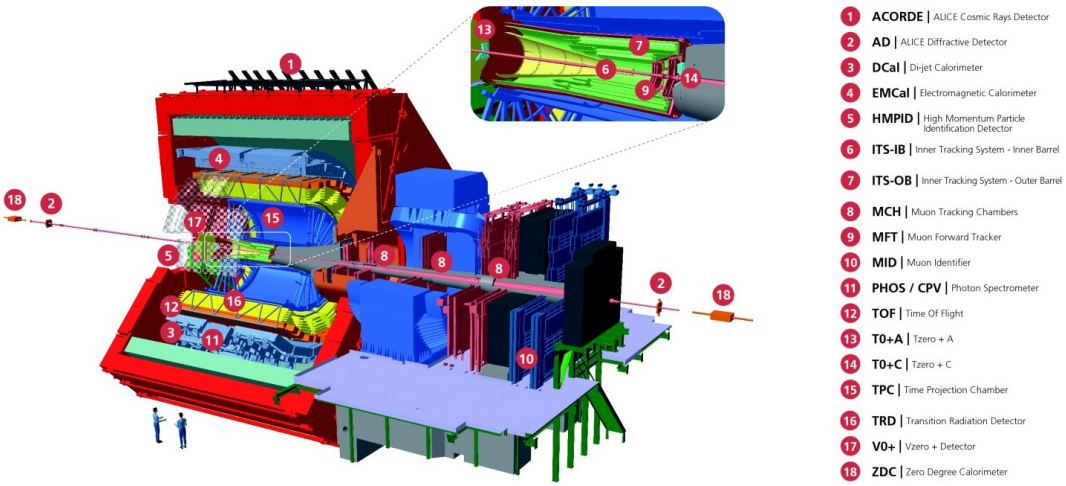


Fig. 2.3 The ALICE experiment. Figure taken from [46]

- The Inner Tracking System (ITS) [47] is the sub-detector closest to the interaction point. The current version is made of seven layers of Monolithic Active Pixel Sensors (MAPS) [48]. This machine is used for primary and secondary vertex determination and to track produced particles.
- The Time Projection Chamber (TPC) [49], a 90 m^3 time projection chamber, is the main detector used for tracking and particle identification.
- The Transition Radiation Detector (TRD) [50] is used for electron identification.
- The Time of Flight System (TOF) [51] is composed by an array of Multi-gap Resistive Plate Chambers (MRPCs) [52] and is used to measure time of flight and particle identification.
- The Electromagnetic Calorimeter (EMCal) [53] is a large lead tungsten alloy calorimeter; it measures the energy of pions, electrons and photons.
- The PHOton Spectrometer and Charge Particle Veto (PHOS/CPV) [54, 55] . PHOS is an electromagnetic spectrometer for photon detection, while the CPV is used to mitigate the contamination in PHOS from charged particles.
- The High Momentum Particle Identification Detector (HMPID) [56] is used for identification of particles such as pions, kaons and protons that have high momentum.

2.2.2 Forward detectors

Most of the detectors of the forward region, with the exception of FV0 and ZDC, are placed on side C and include the following subsystems:

- The Muon Forward Tracker (MFT) [57] is a silicon-based tracker which was installed during LS2 (Long Shutdown 2) i.e. before Run 3, and is used to measure the forward charged particle multiplicity and to improve momentum resolution of the tracks of muons detected in the muon arm.
- The muon arm is composed by the Muon IDentifier (MID) [58] and the Muon CHamber (MCH) [58]. The two subsystems are used to tag and detect muons.
- The Fast Interaction Trigger (FIT) [59, 60] detector is divided into three sub-systems, FV0, FT0-A and C and FDD-A and C. The first one, FV0, is a disk-shaped scintillator used for centrality, multiplicity and event-plane measurements. FT0 is made by quartz radiators placed on both side of the interaction point and are used as luminometers during pp collisions. FDD consists of plastic scintillators and are used to study diffractive events.
- The Zero Degree Calorimeter (ZDC) [61–63] is used for centrality estimation, event-plane measurements and as luminometer in Pb–Pb collisions. The whole system will be explained in detail in section 2.3.

2.2.3 The ALICE Upgrade for Run 3

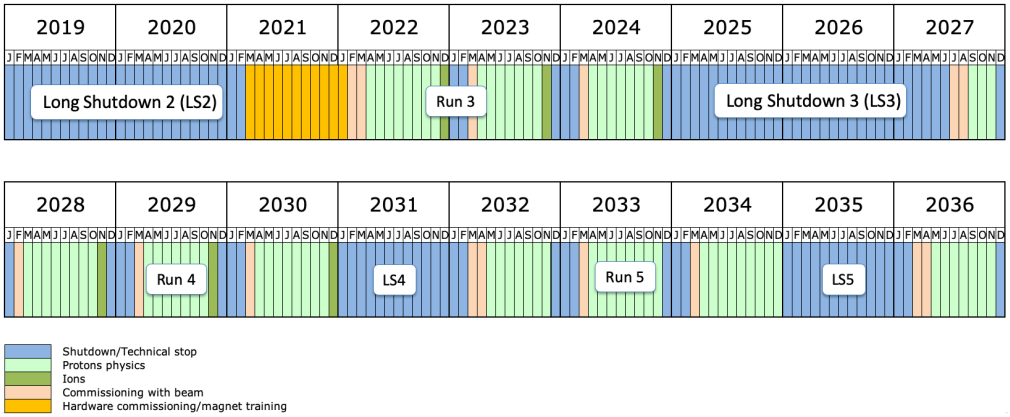


Fig. 2.4 Long term LHC schedule as of September 2020. Figure taken from [64]

After Run 2 the LHC machine and all the four main experiments went through a shutdown phase called Long Shutdown 2 that lasted from November 2018 to April 2022. In figure 2.4 a diagram with the shutdown and Run phases planned until 2036 is shown. During this period the LHC upgraded its injection and collimation [65] systems which allowed to reach an energy of 6.8 TeV per beam and to increase the interaction rate in Pb–Pb collisions at IP2 to about 50 kHz with respect to the 8 kHz achieved in Run 1 and Run 2. In order to cope with the increased event rate foreseen for Run 3 and to allow for new analysis techniques that aim to process all collisions, using a continuous readout method, many subdetectors of the ALICE experiment went through major upgrades [66]. The ALICE computing infrastructure was redesigned in order to cope with the amount of data expected during Run 3 data taking (more than 800 GBps at 45 kHz hadronic interaction rate). In figure 2.5 a diagram with the most relevant upgrades can be seen:

- The dimension of beampipe close to the interaction point was reduced to an internal radius of 18.2 mm with a 0.8 mm thickness.
- The ITS was totally rebuilt; the new system has an active pixel silicon area of 10 m^2 which makes it the largest pixel detector ever built.
- The TPC readout plates were upgraded using high granularity GEM (Gas Electron Multipliers) [67] technology instead of the previous wire chambers [68].
- The MFT detector was added to allow precision tracking in a region that was not previously covered.
- In Run 3 the ALICE experiment is taking data in continuous readout mode [69]. With this new acquiring mode each detector will have to run in a self triggering mode and all the information on the event will be put together when reconstructing the data.
- In order to cope with the new interaction rate and to be compliant with the continuous readout mode the ZDC collaboration had to completely redesign the readout system of the detector. This thesis will analyse my contribution to the project, in particular the development of the FPGA firmware of the new ZDC readout boards.

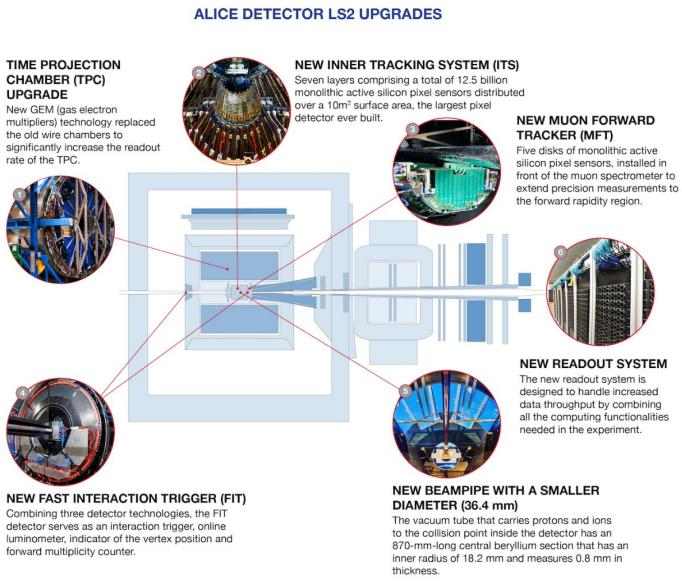


Fig. 2.5 Main upgrades of the ALICE experiment during LS2. Figure taken from [70]

2.3 The ZDC detector

The Zero Degree Calorimeter system (ZDCs) of the ALICE experiment consists of two identical sets of calorimeters located on both sides of the interaction point IP2 (side A and C), 112.5 m away from it in the LHC tunnel. In this region the two LHC beams circulate in two different beam pipes. Each set of detectors consists of a neutron (ZN) and a proton (ZP) calorimeter. A picture of the position of the two calorimeters in the LHC setup can be seen in figure 2.6. The ZN is placed at zero degree with respect to the LHC axis, between the two beam pipes, while the ZP is positioned externally to the outgoing beam pipe, where positive charged particles are deflected by the machine magnetic fields. The two calorimeters sit on top of a moving platform, used to adjust the position of the ZN and ZP to follow the beam crossing angle [72]. In addition, when not in use and during beam injection, the whole detector system can be lowered in a "garage" position outside the path of the particles. In nucleus-nucleus collisions the ZDC is mainly sensitive to spectator nucleons. The spectator protons are separated from the ion beams by means of the LHC dipole separator magnet, while spectator neutrons fly at zero degrees. The ZN, given the space constraints due to its location between the two beam pipes, has a 7.04x7.04 cm front face, made of a high density tungsten alloy and it is segmented into 4 towers, plus common channel. The position of the moving platform is set in

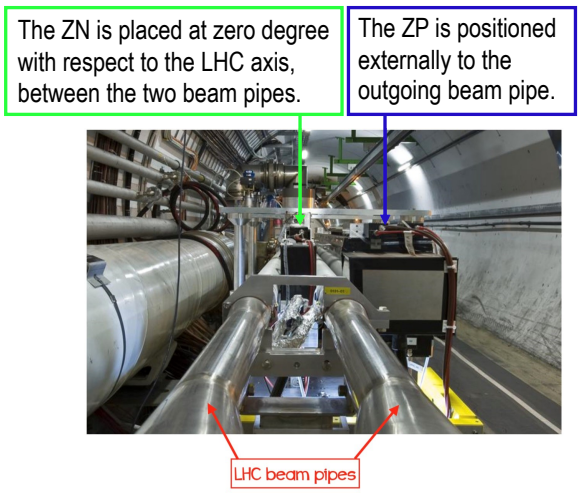


Fig. 2.6 Front view of the ZDC detector without the shielding that covers it during normal operation. Figure taken from [71]

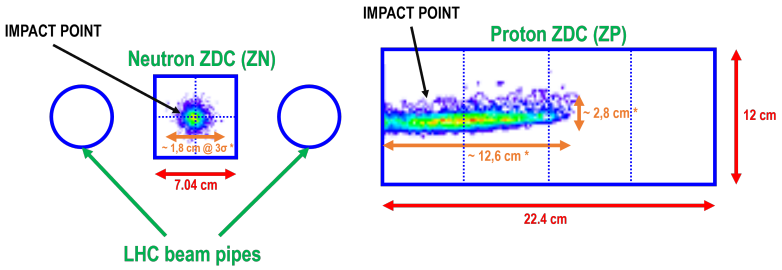


Fig. 2.7 Diagram of the ZDC geometry and segmentation togheter with a simulation of impact point of spectator neutrons and protons on ZN and ZP front faces respectively [61]

such a way to have the neutron impact point vertically centered in the front face, as in figure 2.7. The ZP has less severe space constraints, and this allowed for the use of a less dense passive material as brass and for greater footprint of 12x22.4 cm. The ZDC detector is completed by 2 forward EM calorimeters (ZEM) placed at about 7.35 m from IP2, on side A. The ZDCs are quartz-fiber spaghetti calorimeters with silica optical fibers as active material. In figure 2.8 it can be seen a diagram of the different components of the detector. During the data taking period particles emitted in the very forward directions traveling at relativistic speed enter the dense absorber, produce prticle showers, and generate Cherenkov light in the radiation resistant quartz fibers. These fibers then guide the light to a photomultiplier that produces a proportional electrical pulse. The electrical signal coming from each photomultiplier, one for each ZDC channel, travels through a thick low loss copper

wire up to the Counting Room 4 (CR4) where a patch panel adapts the cabling to a standard LEMO [73] 50 Ω connector. At this point the signal enters a FAN-IN/FAN-OUT chain which tunnels it to the correct readout module. The ZDC readout system is made of 8 readout modules, each one is capable of handling 4 channels. The channels were configured taking into account that the light produced by the hadronic showers is collected in such a way that half of the signal is readout by a common photomultiplier (TC) and the other half by fibers that are bundled to divide the detector into four towers readout by four different photomultipliers (T1, T2, T3, T4), as in figure 2.7. During usual operations the auto triggering of each calorimeter

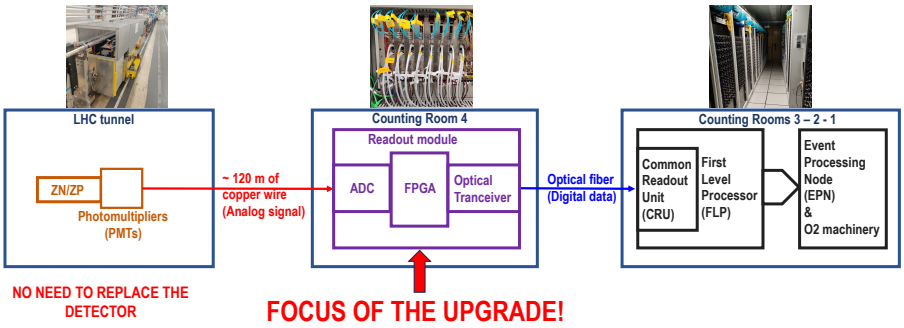


Fig. 2.8 Detector overview sketch

is based on the TC signals since it has the best energy resolution. These channels are therefore fanned-out to two different readout modules for redundancy purposes, but are readout only once. The signals from the remaining four towers are summed (SUM) with a commercial Phillips model 740 quad linear FAN-IN/FAN-OUT [74] to provide a redundant trigger in case of malfunction of a common photomultiplier. The SUM signals are therefore fanned-out to two readout modules too. The signal from each of the four towers is duplicated with a commercial CAEN model 401 quad linear FAN-IN/FAN-OUT [75], one is sent to a readout module and the other is used to form the SUM. Given that the redundancy requirement is less stringent for the proton calorimeters its SUM signals are sent to a single readout module, this gives spaces for the two ZEM calorimeters which are also configured for trigger. A cabling scheme for the 8 readout modules is shown in table 2.1. The FAN-IN/FAN-OUT chain is composed taking into account that the delay from the input of the photomultiplier signal to the input of the ADC must be equal for every channel for timing calibration purposes. From tests in laboratory it was measured that the CAEN model 401 have an intrinsic delay of 5.5 ns and a maximum configurable

	MODULE 0	MODULE 1	MODULE 2	MODULE 3
ch 0	ZNA_TC(T)	ZNA_TC (OT)	ZNC_TC(T)	ZNC_TC(OT)
ch 1	ZNA_SUM(OT)	ZNA_SUM(T)	ZNC_SUM(OT)	ZNC_SUM(T)
ch 2	ZNA_T1	ZNA_T3	ZNC_T1	ZNC_T3
ch 3	ZNA_T2	ZNA_T4	ZNC_T2	ZNC_T4

	MODULE 4	MODULE 5	MODULE 6	MODULE 7
ch 0	ZPA_TC(T)	ZPA_TC (OT)	ZPC_TC(T)	ZPC_TC(OT)
ch 1	ZEM1(T)	ZPA_SUM(T)	ZEM2(T)	ZPC_SUM(T)
ch 2	ZPA_T1	ZPA_T3	ZPC_T3	ZPC_T1
ch 3	ZPA_T2	ZPA_T4	ZPC_T4	ZPC_T2

Table 2.1 Cabling of each IFC1211 module of the upgraded ZDC readout system. T = (Trigger) triggering channel. OT = (Only Trigger) channel used for trigger, but not for readout.

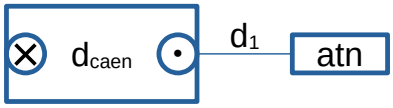
offset of 100 mV, while the PHILLIPS model 740 has a delay of 3.5 ns and 500 mV of maximum configurable offset. The selected digitizer for the ZDC readout has a range of 1 V peak to peak; the ZDC signal has negative polarity, thus, in order to use as much as possible the ADC range, the baseline of the signal is increased using the FAN-IN/FAN-OUT offset. From simulations it was estimated that the signals with the higher amplitude, corresponding to about 60 neutrons, could saturate the ADC. In order to mitigate this issue an attenuator (atn) was added just before the ADC input. Three main configurations can be found.

Configuration 1

The signal enters a CAEN FIFO with a 100 mV offset.

$$TOTAL\ DELAY\ = (5.5+6)\ ns = 11.5\ ns$$

Added delay with cables (d_1) = 6 ns. This applies for channels ZNA T1-T2-T3-T4, ZNC T1-T2-T3-T4, ZPA T1-T2-T3 and ZPC T2-T3-T4.

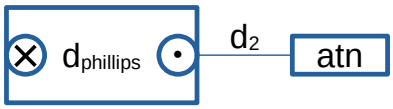


Configuration 2

The signal enters a PHILLIPS FIFO with a 450 mV offset.

$$TOTAL\ DELAY\ = (3.5+8)\ ns = 11.5\ ns$$

Added delay with cables (d_2) = 8 ns. This applies for channels ZEM1, ZEM2, ZNA TC, ZNC TC, ZPA TC and ZPC TC.

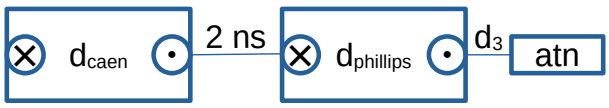


Configuration 3

The signal enters a CAEN FIFO with a 100 mV offset and the a PHILLIPS FIFO with a 350 mV offset.
This applies for channels ZPA T4, ZPC T1.
The signal enters a CAEN FIFO with a 100 mV offset and the a PHILLIPS FIFO with a 100 mV offset.
This applies for channels ZNA SUM, ZNC SUM, ZPA SUM and ZPC SUM.

$$TOTAL\ DELAY\ = (5.5+2+3.5+0.5)\ ns = 11.5\ ns$$

Added delay with cables (d_3) = 0.5 ns.



The four ZN towers are expected to receive approximately the same signal once the calorimeter is centered. On the contrary, the ZP towers receive very different amounts of energy, the one closer to the beam pipe receiving much more signal than the others, as can be seen in figure 2.7. For this reason the two ZP towers that are

closer to the beam pipe (ZPA T4 and ZPC T1) have an higher offset in order to allow for bigger signals without saturating the ADC. The attenuators do not add a significant delay to the signal. The current attenuator configuration is summarized in table 2.2.

	TC	SUM	T1	T2	T3	T4
ZNA	4	4	2	2	2	2
ZNC	4	4	2	2	2	2
ZPA	4	4	0	0	2	2
ZPC	4	4	2	2	0	0

ZEM1	4
ZEM2	4

Table 2.2 ZDC attenuator value in dB for each channel.

2.3.1 The readout module

Given the low amount of detector channels (26 in total), the new readout system can be based on commercial digitizers. These allow a continuous sampling of the signal waveform, followed by digital processing on a FPGA. The readout module hardware was bought from IOxOS Technologies SA [76], a Swiss company based in Gland. The carrier board is a VME [77] IFC_1211 [78] as the one in figure 2.10. The VME format was chosen in order to exploit the existing infrastructure; the VME bus is used only to power the module. Each unit hosts three FPGAs, one of which is a Xilinx Kintex Ultrascale xcku040-1ffva1156 [79], a PowerPC processor and two FMC connectors. The chosen digitizer is the ADC_3112, also from IOxOS Technologies SA, shown in figure 2.9. Each module mounts two TI ADS5409 ADCs [80], has 4

Fig. 2.9 ADC_3112 front panel with the SSMC connectors.

SSMC [81] input channels, a maximum sampling rate of 1 GSps, 12 bit resolution with an Effective Number of Bits (ENOB) of 10 bit, 1 V peak to peak amplitude and

can be configured with DC coupling 50 Ω termination that is therefore adapted to readout a photomultiplier signal. The ADC's nominal sampling rate is 900 GSps, but it can be over-sampled up to 1000 MSps. For the ZDC application it was configured at 960 MSps with decimation by two enabled; this means that the ADC averages two subsequent samples, thus reducing the noise and the data throughput giving an effective sampling rate of 480 MSps (12 samples for each event; each event is 25 ns). The other FMC connector hosts a FM-S14 [82] Quad SFP/SFP+ transceiver. Only

Fig. 2.10 IFC_1211 from IOxOS with two FPGA Mezzanine Card (FMC) modules. The one at the top hosts 4 Small Form-factor Pluggable+ (SFP+) connectors of which only two are populated. The one at the bottom is an ADC_3112 from IOxOS.

two of the four ports are in use, as it can be seen in figure 2.13.

2.3.2 The ALICE data taking infrastructure

In the context of the ALICE upgrade for Run 3, the expected volume of data of heavy-ion events is 3.5 TBps; 100 times more than during Run 1 and Run 2 [83]. In order to cope with the continuous readout requirements and the new data throughput, a sub-set of detectors have been upgraded to use the Common Readout Unit (CRU). The CRU is the interface between the Front-End Electronic (FEE), the Online-Offline facility (O^2), the Detector Control System (DCS) [84, 85] and the Trigger and Timing

System (TTS); a diagram is depicted in figure 2.11. The CRU is a PCIe-gen3 based

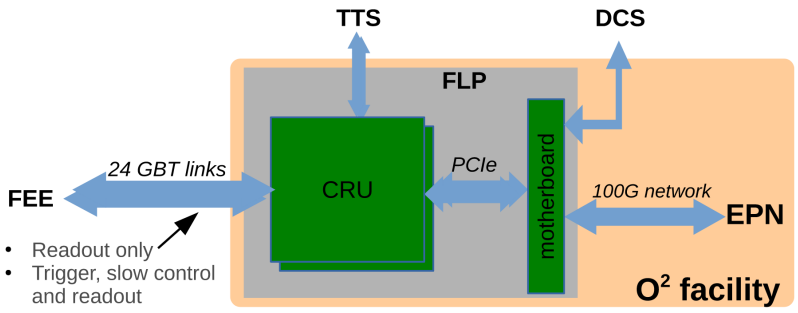


Fig. 2.11 Diagram of the O^2 facility structure. Figure taken from [83]

FPGA processor board capable of handling up to 24 bidirectional optical links with the FEE, one trigger link with the TTS and one PCIe connection with the hosting computer, the First Level Processor (FLP). The O^2 facility is composed of two types of machines FLPs and Event Processing Nodes (EPN); the first ones being used to collect data from up to three CRUs (for the ZDC only 16 links are needed, thus one FLP with just one CRU) and to send them through a 100 Gbps InfiniBand network to the second ones, which are used for event reconstruction and data analysis. The 5 Gbps GBT protocol [86, 87] developed at CERN is used for the FEE-CRU communication. The GBT link can be used to receive data from the detector (Readout), to send clock, trigger, and timing information to the detector and for slow control to and from the detector using by-directional Single Word Transfer (SWT) messages. For what concerns the ZDC specific case, each readout module has two links, namely Link 0 and Link 1. Link 0 is used for the data transmission of channels 0 and 1 and for receiving the trigger messages, the orbit number and the BC number information, while Link 1 is used for the data transmission of channels 2 and 3 and for slow control through the SWT commands. The 24 links of the CRU are bundled into 2 endpoints with 12 links each. In order to spread the data throughput evenly, the 16 ZDC links are divided into two groups of 8, this is summarized in table 2.3. The TTS information that the ZDC modules receive through Link 0 is managed by the Central Trigger Processor (CTP) [88]. The Central Trigger System (CTS) is composed of a CTP board that distributes clock and trigger to 18 Local Trigger Unit (LTU) boards, one for each ALICE detector. During normal operation the CTP controls the LTUs, which act as a communication layer with the detector CRUs. A common ALICE Trigger Board (ATB) using the 6U VME form factor was designed to act as either, CTP or LTU given the appropriate firmware. To summarize,

ZDC MODULE number	PCB id	CRU endpoint	Link numbers
0	1	0	0x00-0x01
1	3	0	0x02-0x03
2	4	1	0x10-0x11
3	5 (2)	1	0x12-0x13
4	6	0	0x08-0x09
5	7	0	0x0A-0x0B
6	8	1	0x0C-0x0D
7	9	1	0x0E-0x0F

Table 2.3 Available link numbers from 0x00 to 0x17.

a single CTP is connected several LTUs, one for each detector, and they are all located in the ALICE experimental cavern. The LTU of each detector is connected via fiber with to its respective CRUs (in the ZDC case just one). This structure is depicted in figure 2.12. The CRU takes the clock, orbit number and BC number from the LTU and tunnels it to the FEE of the detector, in the meantime, during data taking, it receives the event packets from the readout modules and sends it to the FLP. The FLP funnels the data from up to three CRUs and sends it to the EPN farm where it can be stored or processed. In figure 2.13 the complete ZDC readout apparatus is shown; the 32 fibers, 16 TX and 16 RX, are linked directly to the CRU; The RJ45 connectors are used for uploading the new firmware versions and rebooting the readout modules; the 16 brown wires carry the ZDC analog data.

2.3.3 Continuous readout

The trigger message that is sent to the CRUs and subsequently to the detector FEE consists of 80 bits arranged as follows: trigger type TTYPE (32 bits), BC counter (12 bits), spare (4 bits) and ORBIT counter (32 bits) [89]. Every 25 ns the FEE receives either a trigger message or a filler word. In table 2.4 a list of all trigger types available; the most important trigger messages are the ones starting and stopping the runs, in particular:

- 0x6A03 -> for starting a run with continuous readout.
- 0x6C03 -> for ending a run with continuous readout.
- 0x4883 -> for starting a run with triggered readout.

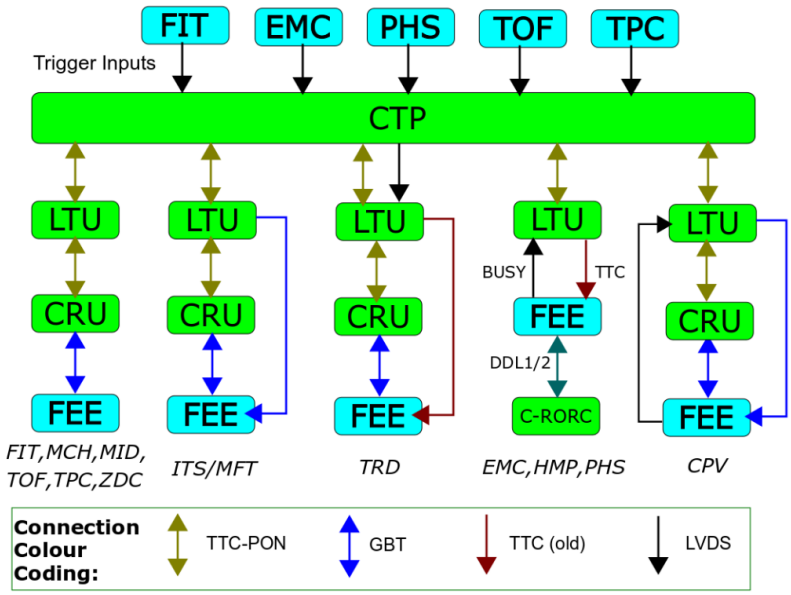


Fig. 2.12 Central Trigger System overview showing trigger inputs to the CTP and the connection types between the CTP and the detector FEE. Figure taken from [88]

- 0x4903 -> for ending a run with triggered readout.

The presence of two distinct run types requires for the acquisition logic to react to either:

- a trigger received from the CTP through the optical fiber with fixed (programmable) latency with respect to the absolute event time (ALICE L0 or L1 trigger).
- an autotrigger event which is generated internally by the logic in the FPGA.

During C (Continuous) readout operation the FEE reacts to both triggers, during T (Triggered) operations only to the CTP received ones.

2.3.4 Autotrigger algorithm

The algorithm, which has been developed and tested by INFN Torino, involves differences between digital samples that are compared to a threshold t. Indicating with y_i the i^{th} ADC sample and considering that the signal has negative polarity, the trigger can be evaluated with a double (eq 2.1) or triple (eq 2.2) condition, the

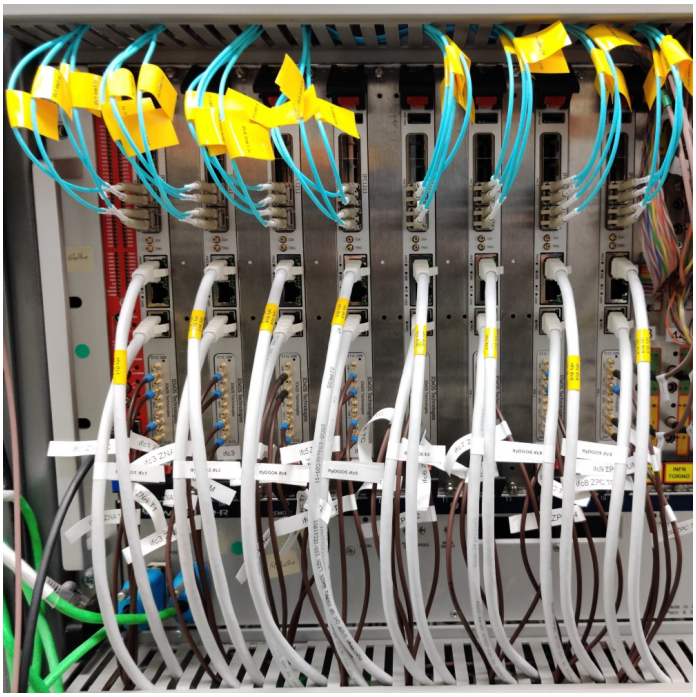


Fig. 2.13 The 8 ZDC readout modules installed at P2 completely cabled and ready for data taking. The leftmost being MODULE 0 and the rightmost being MODULE 7.

first one being less affected by pile-up, while the second one offers an improved protection from the electronic noise.

$$T = (y_i - y_{i+k} > t) \wedge (y_{i+1} - y_{i+k+1} > t) \tag{2.1}$$

$$T = (y_i - y_{i+k} > t) \wedge (y_{i+1} - y_{i+k+1} > t) \wedge (y_{i+2} - y_{i+k+2} > t) \tag{2.2}$$

Typical values for the algorithm parameters are $k = 4$ and $t = 10$ (obtained from simulation and laboratory tests). The trigger algorithm runs on every channel and sets the autotrigger flags independently for each channel. The logic reads the autotrigger flags of the different channels of each module and puts them in logic AND with a configurable trigger mask (autotrigger mask). If the result is not zero this provides the global trigger decision for the module.

Bit	Name	Comment
0	ORBIT	ORBIT
1	HB	Heart Beat flag
2	HBr	Heart Beat reject flag
3	HC	Health Check
4	PhT	Physics Trigger
5	PP	Pre Pulse for calibration
6	Cal	Calibration trigger
7	SOT	Start of Triggered Data
8	EOT	End of Triggered Data
9	SOC	Start of Continuous Data
10	EOC	End of Continuous Data
11	TF	Time Frame delimiter
12	FErst	Front End reset
13	RT	Run Type; 1=Cont, 0=Trig
14	RS	Running state; 1=Running
...	...	Spare
27	LHCgap1	LHC abort gap 1
28	LHCgap2	LHC abort gap 2
29	TPCsync	TPC synchronisation/ITSrst
30	TPCrst	On request reset
31	TOF	TOF special trigger

Table 2.4 Trigger Types. Table taken from [89]

2.4 The ZDC readout firmware

The ZDC readout firmware development can be divided into three main areas, each one provided by a separate entity:

- IOxOS Technologies SA [76] provided the readout boards, the ADCs and the part of the firmware that interfaces with the hardware.
- CERN which provided the IP for the GBT link.
- INFN unit of Turin implemented the interface layer between the digitized samples and the optical link. This comprehends the triggering logic, the calibration procedures, the slow control behaviour and all the configuration options.

As a INFN associate member my primary task was the design of the part of the firmware under INFN responsibility, the integration of the three portions, testing and commissioning of the ZDC readout firmware as well as the final detector commissioning at P2.

Firmware structure summary

Fig. 2.14 Firmware design sources from Vivado 2018.3 project.

The firmware was coded entirely in VHDL using Xilinx VIVADO 2018.3; in figure 2.14 a tree diagram with the most relevant design source files. In figure 2.15 a diagram with the main logic blocks can be seen. A brief description for each component will be given. Looking at figure 2.15 and considering as inputs the analog waveforms and the timing information from the fibers, it can be noticed that:

- from the right, the analog signal enters the readout module and gets digitized by the IOxOS ADC.
- from the left, from Link 0 (SFP0) the timing information arrives from the fiber and gets decoded in the decoder module, the relevant information is sent to each of the four acquisition module.
- the digitized data and the timing information are combined together following the scheme from table 2.6. The various delays in the chain are all taken into

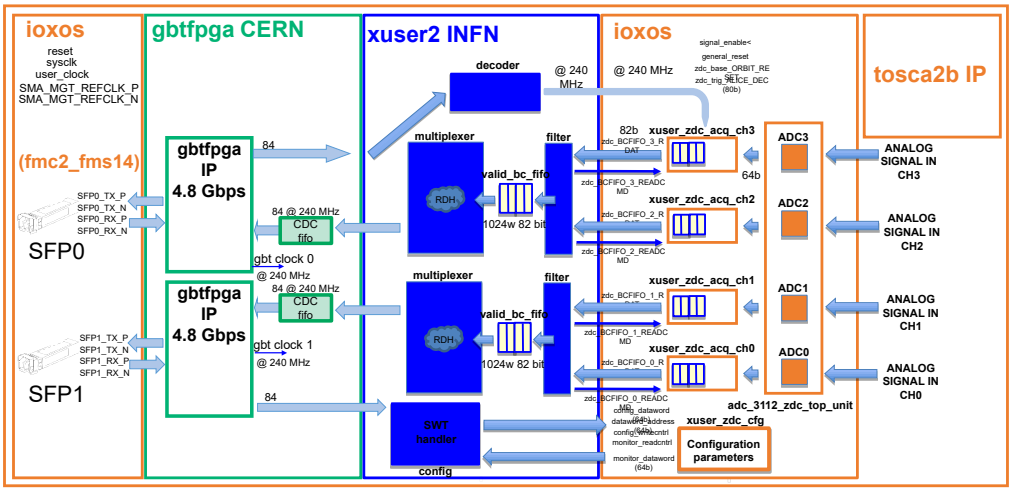


Fig. 2.15 ZDC firmware diagram

account by means of two configurable delays that operate in the acquisition module.

- each acquisition module will generate a 4 words packet, each one 82 bits long at each BC. This corresponds to a data transfer inside the FPGA of 52.5 $Gbps$:

$$\frac{4\,channels * 4\,words * 82\,bits/word}{25\,ns} = 52.48\,bit/ns$$

$$52.48\,bit/ns \approx 52.5\,Gbps$$

- the internal packets are sent to the filter module, where, following the scheme in figure 2.18, all the non relevant BCs are discarded.
- the remaining BCs are sent to the multiplexer module, here the data format is stripped from all the unnecessary and redundant information in order to save up bandwidth (the result is reported in table 2.8) and then saved in a Clock Domain Crossing (CDC) FIFO ready to be transferred to the CRU.
- the slow control is managed by the SWT handler module, a complete list of all available commands will be presented.
- all the configurable parameters are stored in the configuration parameters module and can be changed at any time, in theory even during data taking.
- the main container module (xuser2 INFN) hosts the FSM procedure for the automatic timing calibration. This procedure is performed at the beginning

of each fill and is needed to center the peak of the signal in the middle of the waveform. More detail on this subject will be presented later later.

2.4.1 Acquisition module

The acquisition module was developed by IOxOS, following the specifications requested by the ZDC, and is a wrapper for four sub-modules. In figure 2.16 it is shown an exhaustive color coded diagram with the logic structure of the acquisition block. The four sub-modules (zdc_acq_adcfe, zdc_acq_atrig, zdc_acq_rgbuf and zdc_acq_evtbuilt), arranged in a pipelined configuration, take the raw data from the ADC and assemble a packet with timing and trigger information.

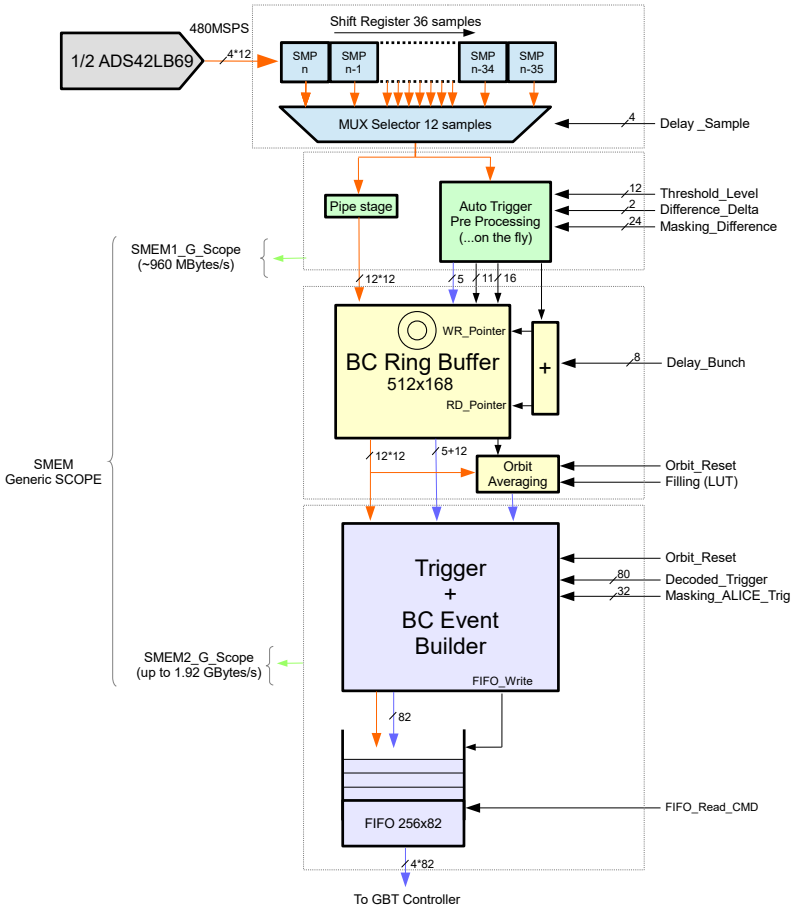


Fig. 2.16 ZDC data acquisition chain diagram. Figure taken from [90]

zdc_acq_adcfe [90]

The ZDC ADC Front-end (in blue in figure 2.16) allows to align the ADC samples to the ALICE reference BC center. The ADC_3112 supplies four consecutive ADC samples at 120 MHz which are continuously fed into a 36 samples shift register. A multiplexer controlled by the "CTL_X_DELAY_SAMPLE[3:0]" configuration parameter allows to select the 12 sample that will be part of the BC. This is the equivalent of adding (or removing) 2.08 ns of delay for each unit of DELAY_SAMPLE. In figure 2.17 an illustration with the effect of the DELAY_SAMPLE modification. Each channel has an independent delay, namely:

- CTL_0_DELAY_SAMPLE[03:0] -> default value= 0x0
- CTL_1_DELAY_SAMPLE[03:0] -> default value= 0x0
- CTL_2_DELAY_SAMPLE[03:0] -> default value= 0x0
- CTL_3_DELAY_SAMPLE[03:0] -> default value= 0x0

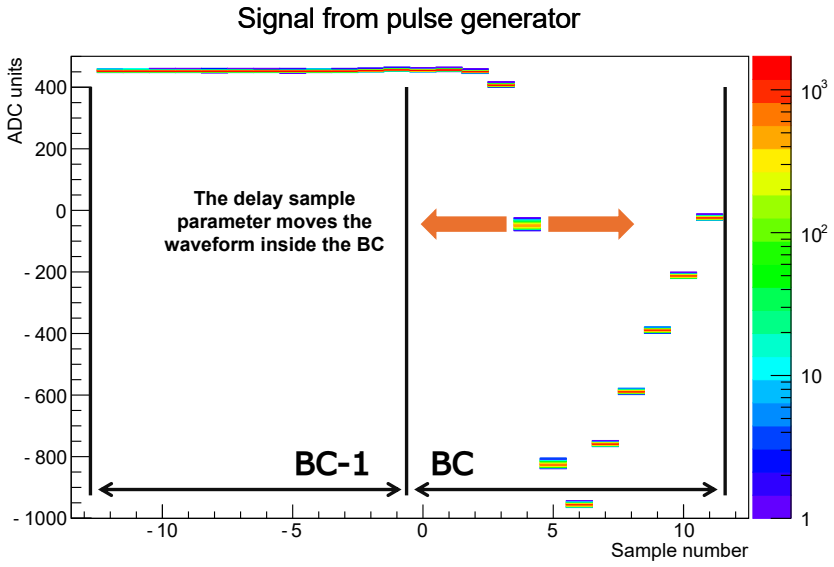


Fig. 2.17 ZDC signal from Pulse Generator.

zdc_acq_atrig [90]

The ZDC autotrigger module (in green in figure 2.16) is designed to perform two tasks, the sum of the 12 samples composing the BC and the evaluation of the autotrigger algorithm threshold comparisons.

- The sum of the 12 samples is computed in the module and the output is a 16 bit vector which is used to compute the average baseline for the current orbit. This is done by adding together all the relevant BCs into a bus named 'ORBIT sum' and then dividing the values by the number of events. This is calculated in the evtbuilt module.
- The autotrigger processing applies the threshold discrimination to $y_i - y_{i+k} > t$ with i from 0 to 11. The result is a 12 bit vector where each bit represents the i^{th} check. A subsequent configurable priority encoder logic can be set to extract:
  - The first 3 consecutive match detected (triple condition).
  - The first 2 consecutive match detected (double condition).
  - The last 3 consecutive match detected (triple edge condition) - disabled.
  - The last 2 consecutive match detected (double edge condition) - disabled.

  The module provides as output the position of the first detected match that will be called autotrigger sample value.

The selection between autotrigger modes is done via the ZDC_TRIGGER _CONDITION[01:0] parameter; in table 2.5 the list of all possible values. The 12 ADC

Parameter Value	Trigger Condition
0b00 (0x0)	Double condition (11)
0b01 (0x1)	Triple condition (111)
0b10 (0x2)	Edge condition (011) - disabled
0b11 (0x3)	Reverse double condition (11) - disabled

Table 2.5 Trigger condition configuration parameter. In later revisions of the firmware the configurations 0x2 and 0x3 have been disabled to save up resources.

samples of the current BC and the output from the autotrigger algorithm are pushed to the ZDC ring buffer module.

zdc_acq_rgbuf [90]

The ZDC ring buffer module (in yellow in figure 2.16) implements the "DELAY COARSE" functionality which is a delay with 1 BC granularity. Given the length of the copper cabling from the tunnel to CR4, the signals arrive at the ZDC FE input $\approx$ 1260 ns after the event. In the meantime the L1 triggers are dispatched from the CRU after $\approx$ 7830 ns. A maximum possible delay between trigger and data can be estimated between 5970 and 6570 ns. The "DELAY COARSE" feature was thus designed in order to guarantee a configurable delay between 1 BC and 512 BC. The functionality is implemented with a 512x128 Dual-Ported RAM (DPRAM) managed as a ring buffer in order to act as a pseudo programmable depth FIFO. The DPRAM Port A WRITE address pointer is incremented at every BC event, while the DPRAM Port B READ address pointer is derived as:

$$READ_{address} = WRITE_{address} - CTL\_X\_DELAY\_COARSE[08:0] - \beta$$

$$\beta = correction\,factor = 7$$

zdc_acq_evtbuilt [90]

The ZDC event builder module (in light purple in figure 2.16) builds and pushes to the output FIFO (BC_FIFO_ACQ) a 4 word frame containing the BC event information. In addition it calculates the average baseline of each orbit and manages the trigger information. For the baseline averaging calculation a pseudo 4095x1 DPRAM (only 3564 bit are used) is used to include or exclude each BC from the calculation in order to use only bunch crossings where no signal can influence the baseline. From the zdc_acq_atrig module the sum of the 12 samples is received. If the BC is enabled then the valued is added to zdc_BASE_SUM[27:0]. By knowing the number of enabled BCs the computation of the average is trivial. The BCs that are used for the average calculation are selected prior to the start of the run. Only the BCs where LHC has no collisions (Empty Bunches) are selected for the baseline averaging. By selecting the number of BCs only as a power of 2 (2, 4, 8, 16, ...) the division by shifting the zdc_BASE_SUM sum vector is easy. A series of tuned shift registers are used to calibrate the average information, the ADC samples, the trigger bits and the timing information. In table 2.6 is shown the BC event frame (4 words X 82 bit), the first word being the TTC information from the CTP and bits [01:00] being the

delimiters for each word. In table 2.7 a brief description of each domain in the data

01	00	
0	0	
0	1	
1	0	
1	1	

21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02
TRIGGER DATA FROM ALICE (19-0)																			
AVG OFFSET (11-0) CHx												T3	T2	T1	T0	A3	A2	A1	A0
SAMPLE 0												TX	TM	SPARE					lst
SAMPLE 6												SPARE		rt_ERROR_STA_FF					

41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22
BC CNT CTP (7-0)								TRIGGER DATA FROM ALICE (31-20)											
BC CNT (3-0)				N EVENTS OVER TH												AVG OFF (15-12)			
SAMPLE 2 (7-0)								SAMPLE 1											
SAMPLE 8 (7-0)								SAMPLE 7											

61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42
ORBIT CNT CTP (11-0)												SPARE		CH ID		BC CNT CTP (11-8)			
ORBIT CNT (11-0)												BC CNT (11-4)							
SAMPLE 4 (3-0)				SAMPLE 3												SAMPLE 2 (11-8)			
SAMPLE 10 (3-0)				SAMPLE 9												SAMPLE 8 (11-8)			

81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62
ORBIT CNT CTP (31-12)																			
ORBIT CNT (31-12)																			
SAMPLE 5												SAMPLE 4 (11-4)							
SAMPLE 11												SAMPLE 10 (11-4)							

Table 2.6 IOxOS data format from the ADC module to the filter. In black the bit number. In red word 0. In blue word 1. In green word 2. In magenta word 3.

frame.

2.4.2 Filter module

The filter module is the first entity in the data path entirely coded by the INFN ZDC team. This block manages the read sequence from the FIFOs of each IOxOS acquisition module and selects the interesting BCs. Each module manages 2 ZDC channels. Two parallel processes read the 4 words for each channels and store the information in a set of registers. A combinator logic circuit, in figure 2.18, generates a TRG (*readout_logic_enable*) signal and a *FORCE_write* signal. The first one "flags" the event for readout; the second one identifies the BC that must always be transmitted (first and last one). To summarize:

- in case of ALICE physics triggers the triggering BC is always sent, in addition:
  - the bunch crossing previous to the trigger (bunch -1) is always sent for pedestal estimation;
  - in case there is a signal (autotrigger) two bunch crossings previous to the trigger (bunch -2) the bunch is readout;

Name	Comment
TRIGGER DATA FROM ALICE	Trigger information from CTP
BC CNT CTP	Bunch Crossing counter from CTP
CH ID	Channel identifier (0x0,0x1,0x2,0x3)
ORBIT CNT CTP	Orbit counter from CTP
A0	ALICE trigger current BC
A1	ALICE trigger BC-1
A2	ALICE trigger BC-2
A3	ALICE trigger BC-3
T0	autotrigger current BC
T1	autotrigger BC-1
T2	autotrigger BC-2
T3	autotrigger BC-3
AVG OFFSET	Average orbit baseline signal
N EVENTS OVER TH	Number of trigger of the channel per orbit
BC CNT	BC counter
ORBIT CNT	Orbit counter
lst	Lost data error
TM	autotrigger BC+1
TX	autotrigger current BC (this channel)
rt_ERROR_STA_FF	Error messages

Table 2.7 ZDC IOxOS data types comments

  - in case there is a signal (autotrigger) three bunch crossings previous to the trigger (bunch -3) both bunch -3 and bunch -2 are readout;

  This logic ensures that it is transmitted enough information to correct for pile-up event event if it is identified only by ZDC and not by ALICE trigger.

- in case of autotrigger:
  - if there is an autotrigger the previous BC and the current BC are readout;
  - if the channel is configured for trigger and there is an autotrigger then bunch +1 is read;
- the last and first BC for each orbit are always readout in order to transmit:
  - the counter for the autotriggered BCs for each channel;
  - the mean baseline (pedestal) for each orbit;

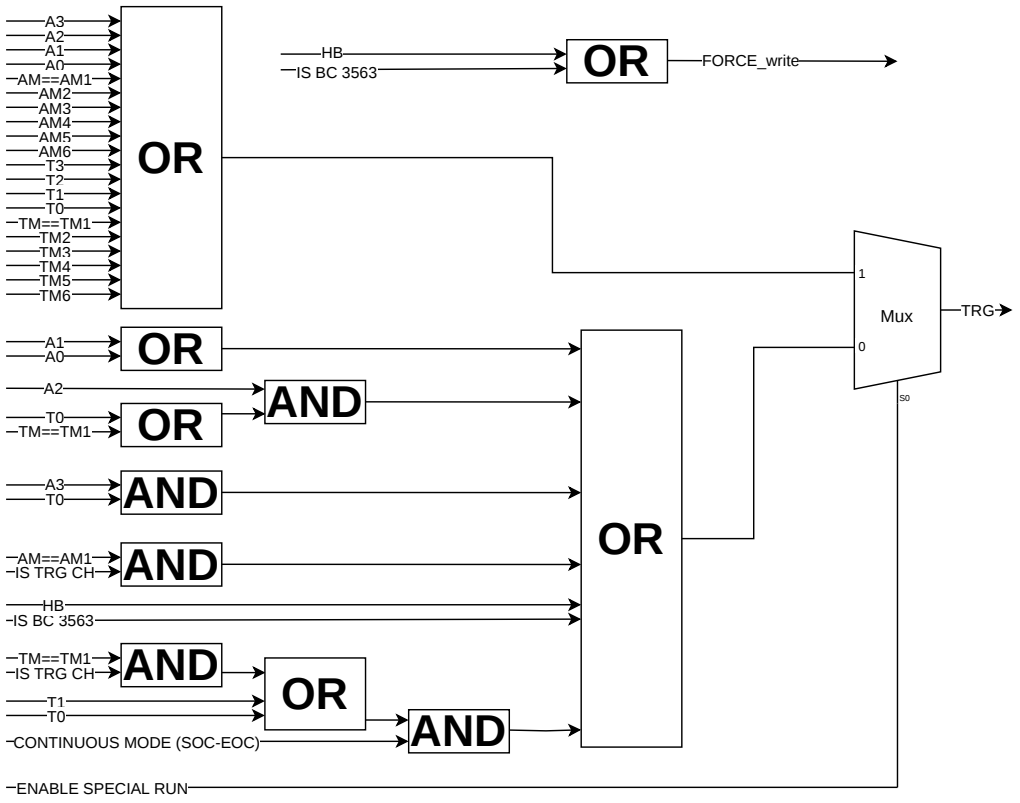


Fig. 2.18 ZDC trigger logic diagram.

- during the commissioning phase it has been important to acquire the full waveform (for example to create templates for Monte Carlo simulations). This was enabled or disabled via the ENABLE_SPECIAL_RUN signal;

This logic enforces that the minimum BC rate for each channel is 22492 Hz. Once the BCs in the registers and the trigger signals are ready, the information can be sent to the next module. In figure 2.19 the diagram of the main filter FSM. In table 2.1 it was shown that not all channels are configured for readout; some are spare or used only for trigger. This information is stored in the TRIG_0123_READOUT_MSK[3:0] register (table 2.10) where each bit represents a channel, if the bit is HIGH the channel is configured for readout. The bus is then divided, the first two bits going to filter entity 01 and the second two going to filter entity 23. The FSM uses a 240 MHz clock, considering the worst delay scenario where both channel are processed, the system needs $8\,x\,4.2\,ns \approx 33\,ns$ to send a BC for two channels. This is more than the 25 ns of a BC and this can lead to a possible throughput bottleneck. However, it will be shown that the 4.8 Gbps GBT link will saturate way before this condition

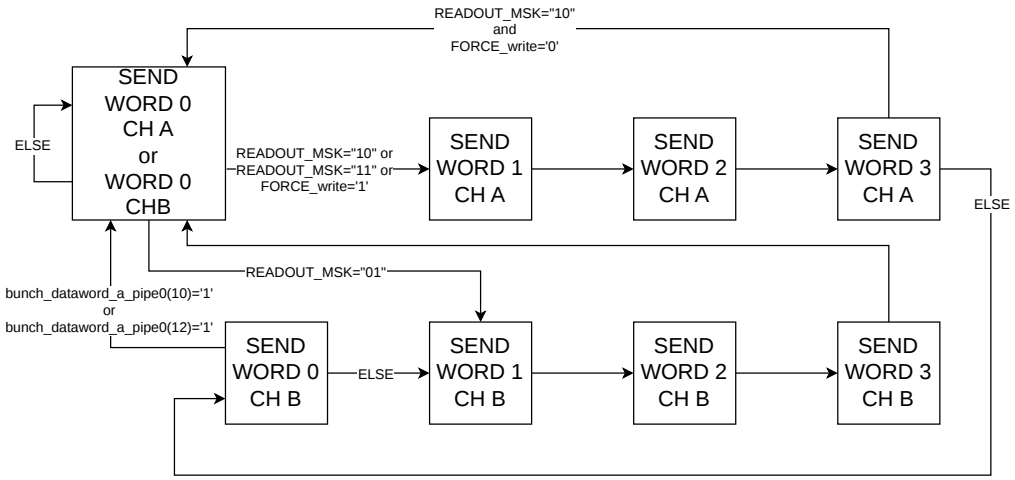


Fig. 2.19 ZDC FSM push filter module.

is reached. The selected BCs are then saved in a FIFO, ready to be read by the multiplexer module.

2.4.3 Multiplexer module

The multiplexer module reads the BCs from the VALID BC FIFO, encodes the words according to the ZDC data format reported in table 2.8, builds the RDH (Raw Data Header) words for the CRU communication and writes all the resulting data into a CDC FIFO. In figure 2.20 a diagram of the Multiplexer FSM that builds and writes into the FIFO the data words is shown. It is worth highlighting that, in order to

19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
AVG OFFSET (11-0)												CH		BOARD ID				0	0
SAMPLE 0												A3	A2	A1	A0		lst	0	1
SAMPLE 6												T3	T2	T1	T0	TM	TX	1	0

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20
BC CNT (3-0)				N EVENTS OVER TH												AVG OFF (15-12)			
SAMPLE 2 (7-0)								SAMPLE 1											
SAMPLE 8 (7-0)								SAMPLE 7											

59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40
ORBIT CNT (11-0)												BC CNT (11-4)							
SAMPLE 4 (3-0)				SAMPLE 3												SAMPLE 2 (11-8)			
SAMPLE 10 (3-0)				SAMPLE 9												SAMPLE 8 (11-8)			

79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60
ORBIT CNT (31-12)																			
SAMPLE 5												SAMPLE 4 (11-4)							
SAMPLE 11												SAMPLE 10 (11-4)							

Table 2.8 ZDC data format from the readout module to the FLP. In black the bit number. In red word 0. In blue word 1. In green word 2.

reduce the total data throughput, the ZDC data format is composed of 3 words, each one 80 bit long.

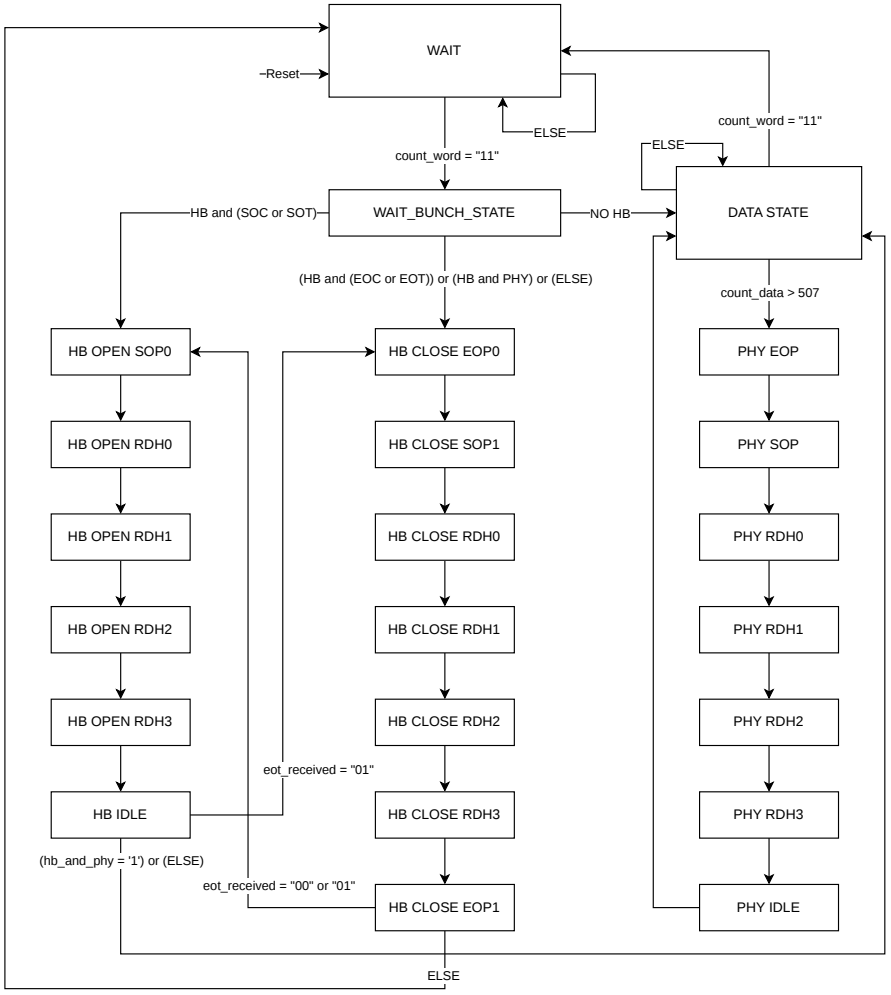


Fig. 2.20 ZDC Multiplexer FSM diagram.

2.4.4 Configuration module

The configuration module stores and allows read/write operation on all the registers needed for the module configuration. The framework was built by IOxOS, and it has been expanded at INFN by adding several new registers and additional options for the DELAY_COARSE parameter settings. In tables 2.9 and 2.10 a list of all the configurable parameters is displayed. Below a brief description for all of them:

Parameter name	Address	Default value
CTL_0_DELAY_SAMPLE [03:0]	0x00	0x0
CTL_1_DELAY_SAMPLE [03:0]	0x01	0x0
CTL_2_DELAY_SAMPLE [03:0]	0x02	0x0
CTL_3_DELAY_SAMPLE [03:0]	0x03	0x0
CTL_0_DELAY_COARSE [08:0]	0x04	0xE
CTL_1_DELAY_COARSE [08:0]	0x05	0xE
CTL_2_DELAY_COARSE [08:0]	0x06	0xE
CTL_3_DELAY_COARSE [08:0]	0x07	0xE
TRIG_0_AUTO_THRES_LEV [11:0]	0x08	0xA
TRIG_1_AUTO_THRES_LEV [11:0]	0x09	0xA
TRIG_2_AUTO_THRES_LEV [11:0]	0x0a	0xA
TRIG_3_AUTO_THRES_LEV [11:0]	0x0b	0xA
TRIG_0_AUTO_DIFF_DELTA [02:0]	0x0c	0b100
TRIG_1_AUTO_DIFF_DELTA [02:0]	0x0d	0b100
TRIG_2_AUTO_DIFF_DELTA [02:0]	0x0e	0b100
TRIG_3_AUTO_DIFF_DELTA [02:0]	0x0f	0b100
TRIG_0_AUTO_MASK_DIFF [23:0]	0x10	0x001fff
TRIG_1_AUTO_MASK_DIFF [23:0]	0x11	0x001fff
TRIG_2_AUTO_MASK_DIFF [23:0]	0x12	0x001fff
TRIG_3_AUTO_MASK_DIFF [23:0]	0x13	0x001fff
FILLING [63:0]	0x14 $\rightarrow$ 0x4b	0x0000000000000000

Table 2.9 Configurable parameters part 1.

- DELAY_SAMPLE CH_X
  Each one of the four channels has an independent register; it is used to set the position of the waveform inside the BC, thus correcting for the differences in the cabling length and the phase shift from the LHC clock and ZDC generated clock.

- DELAY_COARSE CH_X
  Each one of the four channels has an independent register; it is used to compensate for the delay of the analog signal coming from the LHC tunnel into CR4 and the ALICE trigger coming from the CTP.

- AUTO_THRES_LEV CH_X
  Each one of the four channels has an independent register; it is the value of the threshold t used in the autotriggering algorithm from equations 2.1 and 2.2.

Parameter name	Address	Default value
TRIG_0123_ALICE_MSK [31:0]	0x4c	0x00000010
TRIG_0123_AUTO_MSK [03:0]	0x4d	0xC
TRIG_0123_READOUT_MSK [03:0]	0x4e	0xF
ZDC_BASELINE_DIV [03:0]	0x4f	0x1
ZDC_TRIGGER_CONDITION [01:0]	0x50	0b00
SELECTED_TRIGGER_SAMPLE [03:0]	0x51	0x6
DELAY_COARSE_DEFAULT [08:0]	0x52	0xB
FIRST_BC_COLLISION [11:0]	0x53	0x010
AUTO_RESET_SYNC [00:0]	0x54	0b1
AUTO_CAL_0123_MASK [03:0]	0x55	0xF
AUTO_CAL_AVG_CFG [03:0]	0x56	0x4
EN_DS_FINE [00:0]	0x57	0b1
DEBUG_SIGNAL_CONFIG [03:0]	0x58	0x6
EN_TRG_AUTO_CALIB [00:0]	0x59	0b1
BC_ABORT_GAP_START_1 [11:0]	0x5a	0x0E6
BC_ABORT_GAP_STOP_1 [11:0]	0x5b	0x152
BC_ABORT_GAP_START_2 [11:0]	0x5c	0xB57
BC_ABORT_GAP_STOP_2 [11:0]	0x5d	0xBC3
CAL_TRG_COSTANT [31:0]	0x5e	0x0000EA60
STA_0123_SIGNATURE [63:0]	0x1e0	0x0000001020201022
TST_0123_RWREG [63:0]	0x1e1	0x7766554433221100

Table 2.10 Configurable parameters part 2.

- AUTO_DIFF_DELTA CH_X
  Each one of the four channels has an independent register; it is sample separation k used in the autotriggering algorithm from equations 2.1 and 2.2.

- AUTO_MASK_DIFF CH_X
  Each one of the four channels has an independent register; each bit is associated with a ZDC sample. When the bit is 1, the sample is used for the autotrigger algorithm, if it is 0 then the sample is disabled. This allows to configure the module in such a way to be sensible only in selected portions of the waveform. This feature is extremely useful during debugging and testing. During data taking the system is generally configured in order to be sensible over the entire waveform length.

- FILLING
  In order to calculate the average baseline for each orbit the readout mod-

ules needs to know where are located the BCs that, for sure, won't have any collision. Before each new fill, the LHC publishes a string with the beam configuration; for example, number of bunches of beam A and beam C, position and number of the colliding bunches in P2 and position and number of the empty bunches in IP2. This string is decoded by the ZDC DCS WINcc panel and converted into a set of 64 bit words. Each bit refers to a BC, if the BC is empty the bit is set to 1. Given that the LHC orbit is divided into 3564 BCs, 56 words are needed (3564/64), thus from address 0x14 to 0x4b. The firmware uses this information to select the BCs enabled for the average calculation. This is done through the pseudo 4095x1 DPRAM (mentioned in section 2.4.1), of which only the first 3564 locations are used.

- ALICE_MSK
  This 32 bit register is used to mask the trigger message coming from the CTP. This is performed by doing an AND bitwise operation between the mask and the trigger message. The resulting vector is or_reduced. The output of this operation is a single signal. If the bit is 1 then the BC is flagged with ALICE trigger, otherwise it is not. During usual operation the mask is 0x00000010, thus only one trigger type is accepted, in this case this is the Physics Trigger (bit number 4).

- AUTO_MSK
  This 4 bit mask is used to select which channels are configured to give the autotrigger signal to the readout module. If a channel detects an event, the TX bit of the respective channel will always be set to high, however, if the channel is enabled for autotriggering by this mask, the trigger information will be propagated to all other channels via the T3-T2-T1-T0-TM bits. If more channels are enabled they are put in an OR configuration. During normal operation channels 0 and 1 (COMMON and SUM) give the trigger. Channel 0 -> Bit 3; Channel 1 -> Bit 2; Channel 2 -> Bit 1; Channel 3 -> Bit 0. Thanks to this logic a triggering channels can force the readout of the entire module.

- READOUT_MSK
  This 4 bit mask is used to select which channels are configured for readout. If the bit of the respective channel is 1 then it is sent to the multiplexer, otherwise it gets discarded in the filter. The READOUT_MSK and AUTO_MSK are completely independent. A channel can be disabled for readout, but still give

trigger and another channel can be enabled for readout, but not for triggering. The readout masking is done in the filter module and the autotrigger masking is done in the acquisition module. Channel 0 -> Bit 3; Channel 1 -> Bit 2; Channel 2 -> Bit 1; Channel 3 -> Bit 0.

- BASELINE_DIV
  This register is used for the baseline average calculation. In the acquisition module all the selected BCs (in which there is only baseline) are summed in a single, 28 bit whide, register. During the configuration phase the DCS algorithm takes care of marking those BCs only in power of two (2, 4, 8, ... up to 2048); in this way the division can be done by simply shifting the 12 bit output average starting from the sum. This means that:
  - If BASELINE_DIV = 0x0 then the average will be (15 downto 00) of the sum vector.
  - If BASELINE_DIV = 0x1 then the average will be (16 downto 01) of the sum vector.
  - ....
  - If BASELINE_DIV = 0xB then the average will be (26 downto 11) of the sum vector.

  The value of the BASELINE_DIV parameter is calculated and sent by the DCS ZDC slow control logic.

- TRIGGER_CONDITION
  This register refers to table 2.5 and is used to select which trigger algorithm the module is using.

- SELECTED_TRIGGER_SAMPLE
  This register is needed for the auto calibration procedure described in section 2.4.7. The parameter, usually 0x6, indicates the sample in which the peak of the signal should be located at the end of the calibration procedure.

- DELAY_COARSE_DEFAULT
  This register, needed for the auto calibration procedure, is used at the beginning of the procedure to set a default known value almost good for every channel. The delay coarse is not expected to change between one fill and another, the

± 1 tolerance is given just to extend the range of the delay sample adjustments. During normal operation the default value is 0xB.

- FIRST_BC_COLLISION
  This register, needed for the auto calibration procedure, is uploaded by the DCS and is a BC in which a collision is expected (usually the first one in the orbit or an isolated collision). This is used as a reference in order to tune the delay coarse accordingly.

- AUTO_RESET_SYNC
  This is a single bit parameter, when it is set to 1 it enables a check on the phase of the recovered clock during the auto reset procedure which will be described in section 2.4.5. During normal operation this feature is enabled.

- AUTO_CAL_MSK
  This 4 bit mask register is needed for the auto calibration procedure. It enables or disables the calibration procedure for each channel. During normal operation all the channels are enabled for calibration. Channel 0 -> Bit 3; Channel 1 -> Bit 2; Channel 2 -> Bit 1; Channel 3 -> Bit 0.

- AUTO_CAL_AVG_CFG
  The auto calibration average configuration parameter is used to select the number of events that the auto calibration procedure uses to find the average trigger sample position. The corresponding values are shown in table 2.12. During normal operation the most precise setting is used (4096 events), however, if it is needed to calibrate in running conditions with very low event rate, this value can be reduced up to 256 events. The integration time (TIMEOUT) is fixed to ≈ 1.1 s.

- EN_DS_FINE
  This single bit register enables or disables the second stage of the delay sample calibration procedure. During normal operation this feature is enabled.

- DEBUG_SIGNAL_CONFIG
  The IFC1211 module has a GPIO port on the front panel. This register allows to select a bit from the trigger message and send it to the output port. Not all trigger bits are enabled for this feature; for debug purposes it was mainly used with CAL and PP from table 2.4. During normal operation this feature is not used.

- EN_TRG_AUTO_CALIB
  This register enables the activation of the auto calibration procedure when the module receives a CAL trigger from the CTP. During normal operation this feature is enabled. To be noted that a single CAL trigger is not sufficient to start the calibration procedure, a configurable amount of CAL triggers must be received in a fixed time span.

- BC_ABORT_GAP_START_1
  This is the BC where the first abort gap area of the LHC orbit start. The abort gap is a section of the LHC filling scheme which is always empty. The reason for that is to enable the LHC beam dump magnets to ramp up in current and direct the beam towards the abort beam dump area. Given that this area is always empty and there is no physics signal the firmware is enabled for counting CAL triggers, and thus calibrating, only in this region. During normal operation this value is 0x0E6.

- BC_ABORT_GAP_STOP_1
  This is the BC where the first abort gap area of the LHC orbit ends. During normal operation this value is 0x152.

- BC_ABORT_GAP_START_2
  This is the BC where the second abort gap area of the LHC orbit start. During normal operation this value is 0xB57.

- BC_ABORT_GAP_STOP_2
  This is the BC where the second abort gap area of the LHC orbit ends. During normal operation this value is 0xBC3.

- CAL_TRG_COSTANT
  This value represents the number CAL triggers that must be received by the readout module in order to start the calibration procedure. The integration time is fixed to $\approx$ 2 s. The value is 0x0000EA60 -> 0d60000. Given that each event increments the counter by 6 units -> 60000/6=10000. 10000 events / 2 s = 5 kHz. This is equivalent to having a threshold on the rate of the CAL trigger. If the rate is greater than 5 kHz then the auto calibration procedure starts.

- STA_SIGNATURE
  Test register used for debugging.

- TST_RWREG
  Test register used for debugging.

2.4.5 SWT handler module

The SWT handler module (config.vhd in the Vivado project; to not be confused with tosca2b_xuser_zdc_cfg which is the parameter configuration module) manages all the SWT words received and transmitted by the readout module and operates the auto reset procedure.

SWT commands

Every GBT word decoded by the ZDC firmware is 80 bit long [91]; When the detector sends physics data, a flag is raised and the whole 80 bit data field is used to transfer the information. When this flag is 0, the CRU considers the GBT word a control word and it uses the four most significant bits (79 down to 76) of the data field to distinguish between the different control words [83].

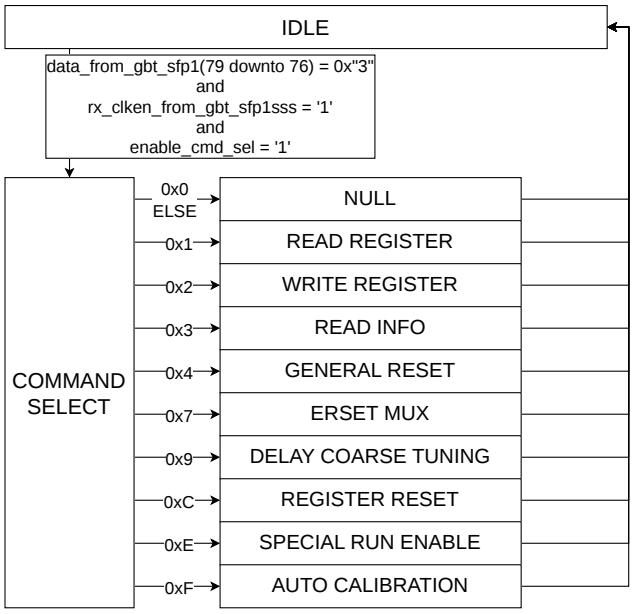


Fig. 2.21 ZDC SWT handler FSM diagram.

- IDLE: 0X0. This word is used to pause the data transfer in case the FEE has no data to send to the CRU.

- SOP: 0X1. This word identifies the start of packet during a data transfer.
- EOP: 0X2. This word identifies the end of packet during a data transfer.
- SWT: 0X3. Single word transfer, it is used to identify a configuration word. The CRU extracts the SWT information from the data stream and stores it in a dedicated FIFO which cab be accessed by the DCS.

In figure 2.21 the FSM logic for the selection of the command when receiving a SWT command. The ZDC SWT data format was chosen to be:

- 4 bits of header (0X3) of the SWT.
- 4 bits of COMMAND.
- 8 bits of ADDRESS.
- 64 bits of DATA.

Below a list of all available commands and a brief explanation:

- 0x0 -> NULL
  This does nothing; it was used just for testing.
- 0x1 -> READ REGISTER
  This is used to read the registers from tables 2.9 and 2.10. For example, the command 0x208_00000000_00000000 is used to read address 0x08, thus the TRIG_0_AUTO_THRES_LEV.
- 0x2 -> WRITE REGISTER
  This is used to write into the registers from tables 2.9 and 2.10. For example, the command 0x213_00000000_000003FC is used to write into address 0x13, thus the TRIG_3_AUTO_MASK_DIFF, the value 0x3FC.
- 0x3 -> READ INFO.
  This command is used to read different status words. It is used for monitoring during commissioning. Below a list of info that can be read for each address:
  - 0x00 -> READ LINK STATUS
    28 bit word which contains the information on the clock and links status. The value that represents a ready board is 0xd4ff2ff; where:

* clock status → 0xd4ff2ff:
  0 → LMK4803 PLL 1 & PLL2 locked
  1 → LMK4803 holdover
  2 → ADC Acquisition PLL locked
  3 → ZDC Clock Generator PLL locked

Bus received from IOXOS

* system status → (0xd4ff2ff):
  · sync_status signal that indicates the phase between the sampling clock and the GBT clock.
* link1 status → 0xd4ff2ff:
  0 → latOptGbtBankTx_from_gbtExmplDsgn
  1 → not rxDataErrorSeen_from_gbtExmplDsgn(1)
  2 → not gbtRxReadyLostFlag_from_gbtExmplDsgn(1)
  3 → gbtRxReady_from_gbtExmplDsgn(1)
  4 → rxFrameClkReady_from_gbtExmplDsgn(1)
  5 → mgtReady_from_gbtExmplDsgn(1)
  6 → txFrameClkPllLocked_from_gbtExmplDsgn
  7 → '1'

Signals received from CERN GBT

* system status → (0xd4ff2ff):
  0 → '0' (spare bit)
  1 → state of run (1 - run ongoing / 0 - run not ongoing)
  2 → links_ready_signal ('1' means that the auto-reset procedure has terminated successfully and the board is ready to receive commands and send data)
  3 → auto_calibration_working ('1' means that the auto-calibration process for the delay sample and delay coarse is in progress.)
* link2 status → 0xd4ff2ff:
  0 → latOptGbtBankTx_from_gbtExmplDsgn
  1 → not rxDataErrorSeen_from_gbtExmplDsgn(2)
  2 → not gbtRxReadyLostFlag_from_gbtExmplDsgn(2)
  3 → gbtRxReady_from_gbtExmplDsgn(2)

4 → rxFrameClkReady_from_gbtExmplDsgn(2)

5 → mgtReady_from_gbtExmplDsgn(2)

6 → txFrameClkPllLocked_from_gbtExmplDsgn

7 → '1'

Signals received from CERN GBT

It is to be noted that the link status is read using the link itself through the SWT words. This means that the only value that can be properly transmitted on the link has a format like 0xd4ff2ff. Everything else should be considered as an indication that the board is not yet ready.

– 0x10 -> READ AUTOTRIGGER SAMPLE CH0 STATUS
the AUTOT values are used during the calibration phase. In particular they are needed in order to properly align the signal and thus to configure the delay sample parameters. A possible output is for example 0x4fff. Each character represents the sample at which the autotrigger algorithm fires in a particular BC.

$$AUTOT_{CH0} =$$

$$Spl_{CH0}(BC_i)\&Spl_{CH0}(BC_{i+1})\&Spl_{CH0}(BC_{i+2})\&Spl_{CH0}(BC_{i+3})$$

$Spl_{CH0}(BC_i) = f$ means that the algorithm did not fire. There is a similar register for each channel and they are all latched simultaneously when:

$$Spl_{CH0}(BC_i) \neq f \; or$$

$$Spl_{CH1}(BC_i) \neq f \; or$$

$$Spl_{CH2}(BC_i) \neq f \; or$$

$$Spl_{CH3}(BC_i) \neq f$$

This means that the user can obtain a still image of the last sample that triggered the algorithm.

– 0x11 -> READ AUTOTRIGGER SAMPLE CH1 STATUS
The same for Channel 1.

– 0x12 -> READ AUTOTRIGGER SAMPLE CH2 STATUS
The same for Channel 2.

- 0x13 -> READ AUTOTRIGGER SAMPLE CH3 STATUS
  The same for Channel 3.
- 0x20 -> READ ORBIT SOX CH 01
  This is the orbit recorded at the last SOX (Start of Continuous / Start Of Triggered) received and must be the same for every board during data taking. To be noted that this value should not be taken into consideration right after a reset and before the SOX. This register is for link 0.
- 0x21 -> READ ORBIT SOX CH 23
  The same for link 1.
- 0x22 -> READ LAST ORBIT CH 01
  This is the last orbit number latched by the firmware.
- 0x23 -> READ LAST ORBIT CH 23
  The same for link 1.
- 0x30 -> READ BC AUTOTRIGGER CH0 STATUS
  This word represents the BC value of the leftmost event displayed in the AUTOT section above. Fox example if $AUTOT_{CH0} = 0x4fff$ and $BCT_{ch0} = 0x56$ it means that at the bunch crossing 0x56 there was an autotrigger event at the sample 0x4.
- 0x31 -> READ BC AUTOTRIGGER CH1 STATUS
  The same for Channel 1.
- 0x32 -> READ BC AUTOTRIGGER CH2 STATUS
  The same for Channel 2.
- 0x33 -> READ BC AUTOTRIGGER CH3 STATUS
  The same for Channel 3.
- 0x40 -> READ AVG EVENT SAMPLE CH0
  This register was extensively used during the commissioning phase of the auto calibration procedure. Is is divided into three sections:
  * 0x605e2:
    This is the rounded up average trigger position during the auto calibration procedure to the nearest sample.
  * 0x6005e2:
    This is always 0x0.

    * 0x605e2:
      This is the average trigger position during the auto calibration procedure.

- 0x41 -> READ AVG EVENT SAMPLE CH1
  The same for Channel 1.
- 0x42 -> READ AVG EVENT SAMPLE CH2
  The same for Channel 2.
- 0x43 -> READ AVG EVENT SAMPLE CH3
  The same for Channel 3.
- 0x50 -> READ EVENT COUNTER CH0
  This is a 64 bit counter that increases by one each time the autotrigger algorithm fires. It is used for monitoring the event rate.
- 0x51 -> READ EVENT COUNTER CH1
  The same for Channel 1.
- 0x52 -> READ EVENT COUNTER CH2
  The same for Channel 2.
- 0x53 -> READ EVENT COUNTER CH3
  The same for Channel 3.
- 0x54 -> READ EVENT COUNTER CH0 AND CH1
  The same for the logic AND between Channel 0 and Channel 1. Those two channels are usually the TC and SUM signals. By doing the AND in the firmware it is possible to reject most of the photomultiplier noise.
- 0x60 -> READ BC OVERFLOW COUNTER CH0
  This is a 32 bit counter which increases by one each time the latched BC value is outside the LHC nominal values (0d0 to 0d3563). This is useful to understand possible issues in the data taking and during the commissioning phase of the readout.
- 0x61 -> READ BC OVERFLOW COUNTER CH1
  The same for Channel 1.
- 0x62 -> READ BC OVERFLOW COUNTER CH2
  The same for Channel 2.
- 0x63 -> READ BC OVERFLOW COUNTER CH3
  The same for Channel 3.

- 0xFF -> READ FIFO STATUS
  this value, which for example for board number 4 can be 0x64000, is divided into:
  * PCB id → 0x64000:
    This value comes from a set of four dip switches on the FPGA carrier and is used to identify the different boards.
  * board number → 0x64000:
    This value is used by the reconstruction software to to identify the different boards and thus the different channels. The conversion between PCB id and board number can be seen in table 2.3.
  * FIFO status → 0x64000:
    00 → BC_FIFO_0_FULL
    01 → BC_FIFO_1_FULL
    02 → BC_FIFO_2_FULL
    03 → BC_FIFO_3_FULL
    04 → VALID_BC_FIFO_0_FULL
    05 → VALID_BC_FIFO_1_FULL
    06 → BC_FIFO_WAS_FULL
    07 → VALID_BC_FIFO_WAS_FULL
    08 → GBT_FIFO_0_FULL
    09 → GBT_FIFO_1_FULL
    10 → GBT_FIFO_WAS_FULL
    11 → '0'

- 0x4 -> GENERAL RESET
  Full reset of the GBT links and the ZDC firmware logic.

- 0x5 -> RESET TX (COMMAND DISABLED)
  Reset of the TX section of the GBT link.

- 0x6 -> RESET RX (COMMAND DISABLED)
  Reset of the RX section of the GBT link.

- 0x7 -> RESET MUX
  Reset of the ZDC firmware logic and NOT the link.

- 0x8 -> RECOVERED CLOCK (COMMAND DISABLED)
  This was used during the development phase to select the clock source.
- 0x9 -> DELAY COARSE TUNING
  During the commissioning phase this was used to increase/decrease the delay coarse value for all the channels simultaneously.
  - 0x10 -> ADD 1 BC TO ALL CHANNELS
  - 0x01 -> REMOVE 1 BC TO ALL CHANNELS
- 0xA -> TEST PATTERN (COMMAND DISABLED)
  Command used during the development phase to send a predefined test pattern.
- 0xB -> LOOP-BACK (COMMAND DISABLED)
  Command used during the development phase to configure the GBT link into a loop-back mode.
- 0xC -> REGISTER RESET
  Command used to reset all the registers from tables 2.9 and 2.10 to the default parameters.
- 0xD -> CHANNEL RESET (COMMAND DISABLED)
  Command used as an interface for the channel reset command intoduced by IOxOS.
- 0xE -> SPECIAL RUN
  Command used to activate the special run mode for acquiring long waveform.
  - 0xE00_00000000_00000001 -> to enable the special run.
  - 0xE00_00000000_00000000 -> to disable the special run.
- 0xF -> AUTO CALIBRATION
  Command used to manually start the auto calibration procedure at any time.

Auto reset procedure

The configuration parameters, slow control and reset commands are sent through the GBT link. If the link is down the communication with the firmware is no longer possible. For this reason a logic that automatically check the status of the link and, if

needed, proceeds to a full reset is needed. This is done using the logic of the FSM in figure 2.22. In the main CHECK STATUS state three checks are performed:

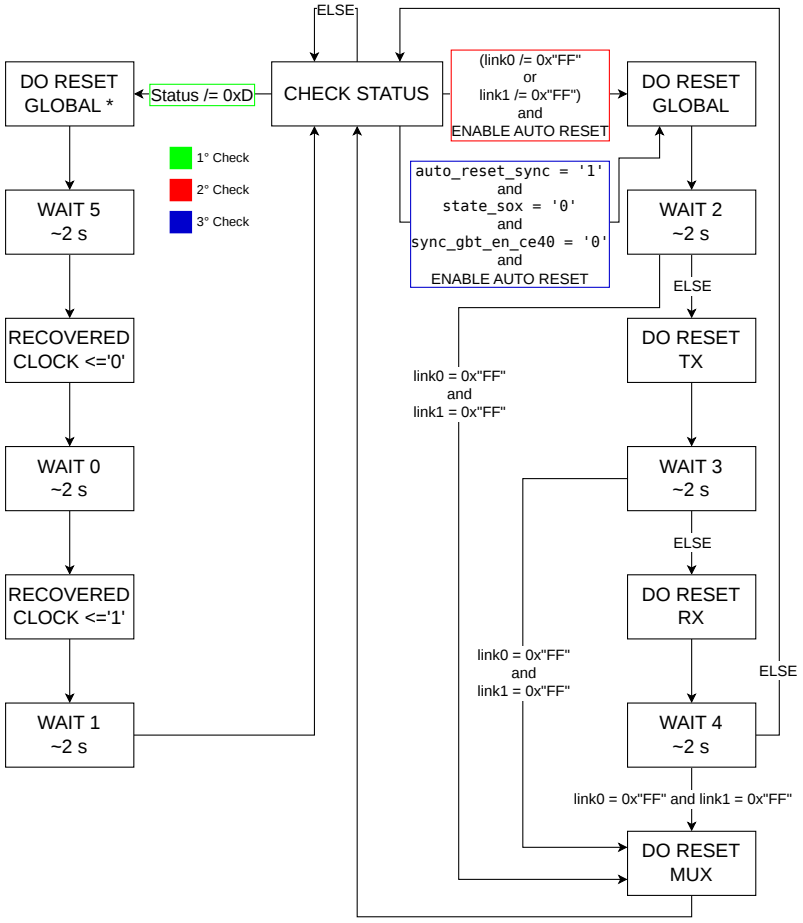


Fig. 2.22 ZDC auto reset FSM diagram.

- 1 → The clock status must be 0xD, thus the ZDC clock generator PLL must be locked.
- 2 → The status of the two links must be 0xFF and the auto reset feature must be enabled.
- 3 → If the AUTO_RESET_SYNC parameter is HIGH and the is no run ongoing and the auto reset is enabled then a check on the phase of the clock is performed. A new GBT word is dispatched every 25 ns (40 MHz), meanwhile the ZDC clock runs at 240 MHz (a factor 6 higher). This means that the GBT word could be theoretically sampled in six different positions. In order to not

have bit flips errors it is preferable to have the word sampled in the middle. During this check the logic compares the ZDC_CE40 shift register bus with the GBT link en signal. If there is a match between GBT_link_10_en and zdc_CE40_ss or zdc_CE40_sss or zdc_CE40_ssss then the phase is correct, otherwise a reset is needed. This can be seen in figure 2.23.

It is important to note that the auto reset feature is not controlled by a configuration parameter. In order to enable or disable this feature a complete firmware rebuild is necessary.

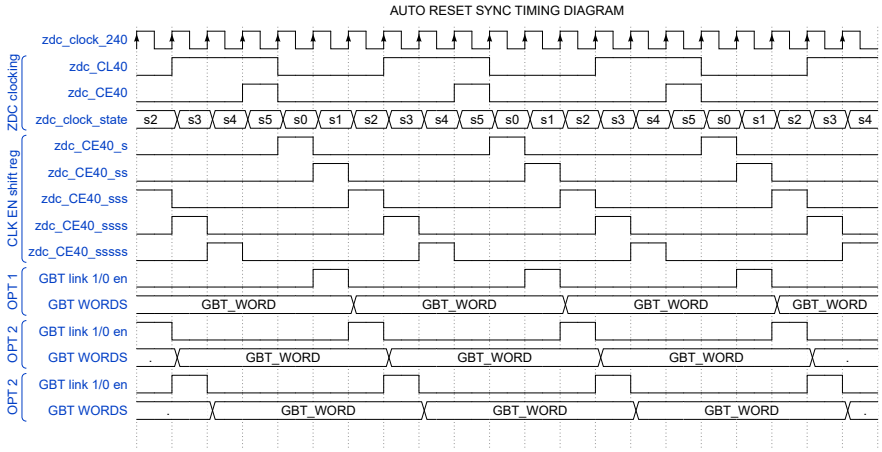


Fig. 2.23 ZDC auto reset timing diagram.

2.4.6 Decoder module

The decoder module receives the TTC signal from the CRU and, through a series of flip flops, synchronises the 240 MHz GBT clocked data with the 240 MHz ZDC clocked logic. A counter for the number of CAL triggers and a logic which decodes the start and the end of a run are present.

2.4.7 XUSER_ZDC module

The XUSER_ZDC module is the main container for the ZDC firmware. This entity houses the four acquisition modules, the configuration module and the GBT logic. In addition, all the logic for the auto calibration procedure is contained here.

Auto calibration procedure

The sampling clock is generated by the ADC PLL, however the ZDC logic works on the clock recovered from the GBT link. Every time that the link is restored the phase between the two clock changes. This is not ideal since, for proper time measurements, the peak of the nominal signal must be centered relative to the bunch crossing. In order to shift the waveform in its proper position, some modification to the two delay parameters (sample and coarse) are required. Given that the clock can change multiple times per day, it is impractical to perform this operation manually. For this reason a system that can automatically calibrate the channels using a reference signal sent by the CTP was developed. In this way the reference signal is always in sync with the collisions. Given that each photomultiplier has a different high voltage setting and thus a different timing response, in the patch panel outside CR4 there is a delay line for each channel calibrated in order to align all the signals in time with the slowest one, and this ensures that all the physics signals are synchronous at the FEE level. In table 2.11 the delay used for pp tests and Pb–Pb data taking is shown. The CAL trigger from the CTP come in CR4 through a LVDS cable. The

	TC	T1	T2	T3	T4
ZNA	6 / 5	4 / 4	5 / 3	5 / 4	0 / 0
ZNC	6 / 5	1 / 0	6 / 5	5 / 4	5 / 3
ZPA	5 / 3	8 / 5	5 / 1	2 / 0	3 / 1
ZPC	4 / 3	2 / 0	2 / 0	4 / 2	5 / 4

ZEM1	-5 / -10
ZEM2	-13 / -18

OUT	20 / 13

Table 2.11 ZDC delay added to synchronise all channels. Values are in ns. In black values for Pb–Pb collisions, in red values for pp collisions. OUT = delay before the passive splitter star.

signal is then converted with a modified discriminator to a very short NIM pulse (under 4 ns width). This signal is sent to a passive splitter star which sends a copy to every input of the FIFO chain that feeds the readout modules. In this way the physics and reference signal end up in the readout module with always the same time shift. In figure 2.24 it is shown a diagram with the path of the physics and

reference signals. In figure 2.25 a diagram with the FSM states of the calibration

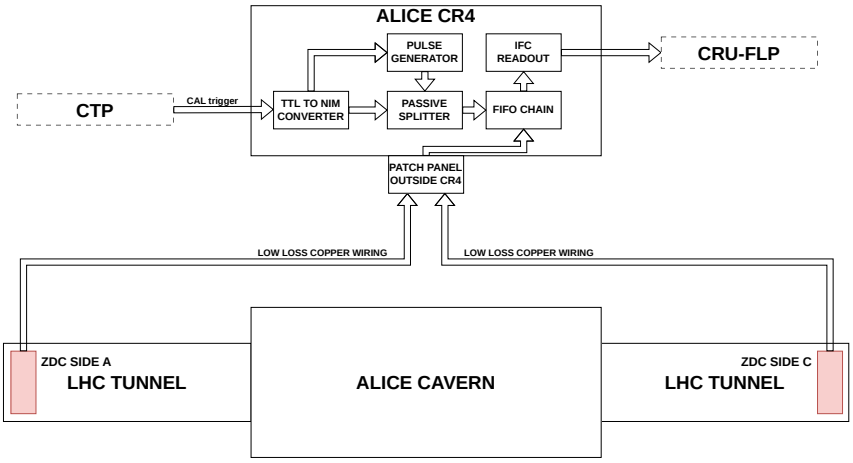


Fig. 2.24 ZDC signal path

procedure is displayed. The calibration procedure exploits the LHC abort gaps to avoid interference from physical signals on calibration signals. This is the standard sequence of calibration steps:

- IDLE
  The system waits for a triggering event to start the calibration. This event can be of two types:
  - an SWT command $\rightarrow$ 0xF00_00000000_00000000.
  - a rate of CAL triggers higher than a configurable threshold. The parameters for the threshold modification are an integration time of ≈ 2 s fixed in firmware at the compilation time and the CAL_TRG_COSTANT. In the decoder module a counter is increased by one unit each time the trigger message has the CAL bit set to high. However, the counter logic works at 240 MHz while the trigger message is refreshed at 40 MHz. This means that every BC with a CAL increases the counter by 6. If the thresholds needs to be configured for 10 kHz (i.e. approximately, considering one CAL trigger per orbit) the value should be:

$$10\,kHz \cdot 2\,s \cdot 6 = 0d120000 = 0x0001D4C0$$

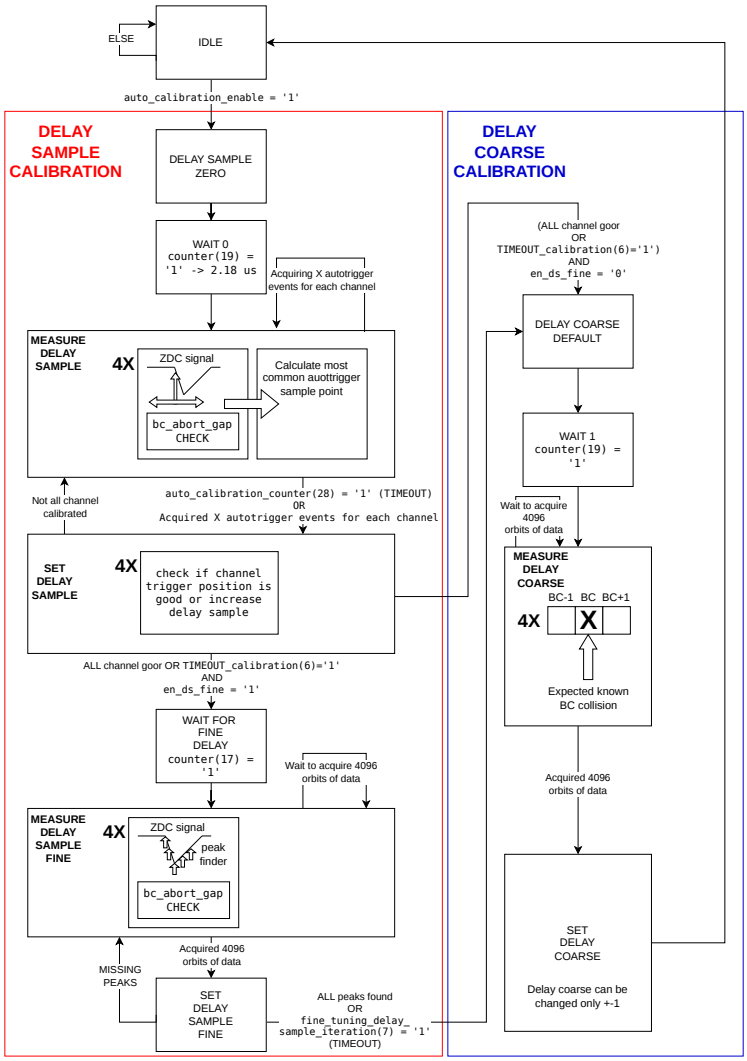


Fig. 2.25 ZDC auto calibration FSM diagram.

During normal operations the threshold is set at 5 kHz as mentioned in section 2.4.4. In addition, the auto calibration starts only if the links are ready, there is a run ongoing, and the auto calibration setting is enabled.

- DELAY SAMPLE ZERO
  The delay sample parameter is set to 0x0 for every channel in order to start from a known value.

- WAIT 0
  A 2.18 μs wait time to let the new configuration apply.

- MEASURE DELAY SAMPLE
  For each channel the autotrigger sample and BC values are continuously monitored. If an event (trigger sample with value different from 0x0 or 0xF) occurs in the abort gap region of the LHC orbit then three actions are done:
  - A counter that keeps track of the number of events increases by one. This value is configurable according to table 2.12; during normal operation the highest value is used.
  - The autotrigger sample value is added to a SUM register.
  - For each possible value of the sample value the respective counter is incremented if an autotrigger is detected in that position. In this way an histogram with the statistics of the autotrigger sample is formed.

AUTO_CAL_AVG_CFG[3:0]	Number of autotrigger
0b0000 (0x0)	256
0b0001 (0x1)	512
0b0010 (0x2)	1024
0b0011 (0x3)	2048
0b0100 (0x4)	4096

Table 2.12 Auto calibration averaging configuration parameter.

  When a channel has reached the desired number of events the calibration is stopped for that channel. After all channels stopped counting, then the most probable autotrigger sample is calculated from the histogram, the average trigger position is calculated by dividing the SUM register by the number of events, and the FSM goes to the next state. Every channel has a timeout timer in case of extremely low rate and can be deactivated by means of a mask parameter.

- SET DELAY SAMPLE
  If the average autotrigger sample has reached the target value, then the logic does not perform any further action, otherwise the delay sample parameter is increased by one. If all the enabled channels have found the right delay value then the FSM goes to the next state, otherwise is goes back to the measurement phase. If after 64 cycles the system does not find the correct combination a timeout counter is triggered. If the delay sample fine tuning

option is enabled the the FSM moves to the WAIT FOR FINE DELAY state, otherwise it switches to the delay coarse calibration. During normal operations the fine tuning option is enabled.

- WAIT FOR FINE DELAY
  A 0.55 μs wait time to let the new configuration apply.

- MEASURE DELAY SAMPLE FINE
  The fine tuning measurement analyses the waveform of the signal itself. If an event is detected, a first check on the position with respect to the LHC orbit is done. If it is in the abort gap area, then a comparison between the central sample of the BC is done. Taking into account that the signal has negative polarity. A three bin histogram is filled is such a way that:

  – when the sample with the lowest value is located before the center of BC the first bin is filled;
  – when the sample with the lowest value is exactly in the center of the BC the middle bin is filled (i.e. the correct position);
  – when the sample with the lowest value is located after the center of BC the last bin is filled;

  After 4096 LHC orbits the number of entries of the histogram are compared and the FSM moves to the next state.

- SET DELAY SAMPLE FINE
  Using the results from the previous calculation the logic can increase or decrease by one unit the delay sample value of each enabled channel. If all channels are correctly centered, the FSM moves to the next state, otherwise the delay parameter of the wrongly calibrated channel gets modified and the system repeats the measuring state until all channels are good. After 128 iterations, if the procedure does not converge then a timeout counter triggers and the system moves to the next state.

- DELAY COARSE DEFAULT
  The delay coarse parameter is set to the default value for every channel in order to start from a known almost good initial condition. The default, in this case 0xB, was measured by sending pulse in a known BC and measuring the shift recorded by the FEE without alignment.

- WAIT 1
  A 2.18 μs wait time to let the new configuration apply.

- MEASURE DELAY COARSE
  During the calibration procedure the reference signal is always sent to the same known BC which is also saved in a configurable parameter. During this state the logic records 4096 LHC orbits with particular attention to the BC where the calibration event is expected and the two BCs before and after that. As in the previous measurement state, an histogram is formed with each bin representing a BC and the number of entries representing the number of events recorded for each BC. At the end of the 4096 orbits the bin with the higher number of entries is extracted and the FSM moves to the next state.

- SET DELAY COARSE
  The delay coarse parameter is modified using the information from the previous state, if needed, and the calibration procedure ends returning to the IDLE state. To be noted that the delay coarse calibration is not iterative, the measurement phase is done just once, because, thanks to the accuracy of the links synchronization procedure, the expected variation from fill to fill of the delay coarse parameter is maximum ± 1, thus a single step correction is enough.

Chapter 3

ZDC data taking and measurements

In this chapter the different data taking periods of the ALICE experiment when the ZDC detector were switched on are presented. In section 3.1 the test setup realized in Turin and CERN are shown. In sections 3.2, 3.3, 3.5 and 3.7 the data taking in proton proton collisions is presented. In sections 3.4 and 3.6 the data taking in lead lead collisions is shown.

3.1 Laboratory tests and commissioning

3.1.1 Setup in Turin

A complete test system for the ZDC readout was assembled in laboratory at INFN Turin. The system is shown in figure 3.1. It has two readout modules, but, in theory, it could be cabled to handle up to 12 links, thus 6 boards. The clock and the trigger information are provided by a LTU VME module with the same hardware used at CERN P2 and the data is collected by the CRU and stored/processed by the FLP (same DELL server as in ALICE CR1). The carrier firmware is stored and loaded on the readout modules by an additional server (not visible in the figure) connected to a network switch.

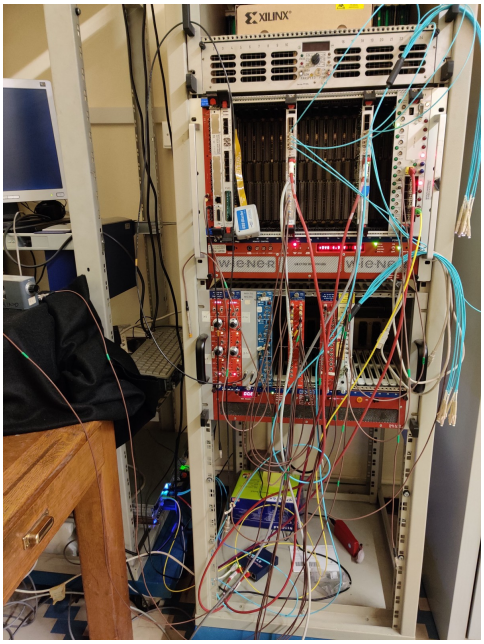


Fig. 3.1 Turin test setup with two readout modules, an LTU, a FLP and a CRU.

3.1.2 Setup at CERN

During the commissioning of the readout system in 2022 a new test setup was assembled at CERN, jointly with the ALICE DAQ team, with the same components of the one tested in Turin. The setup hosted up to two redout modules, the FLP and CRU, the LTU and the server used to load and debug the firmware. The system was then dismantled after the Pb–Pb low intensity run of November 2022.

3.1.3 Maximum trigger rate measurements

During the commissioning phase, a series of studies have been carried out to evaluate the performance of the new readout system. In table 3.1 the maximum achievable rate, in kHz, for different autotrigger and readout configurations is reported. The different settings were chosen taking into account table 2.1. In fact it can be observed that:

- channels 3 and 4 of every board are always transmitted, but they do never provide a trigger; this means that the readout mask will be 0b11 and the autotrigger mask 0b00.

- for boards 0 and 2, channel 1 is used for trigger but it is not transmitted while channel 0 is both transmitted and used as trigger. Thus, for those links, the readout mask will be 0b10 and the autotrigger mask 0b11.
- for boards 4 and 6, channels 0 and 1 are both transmitted and used as trigger. Thus, for those links, the readout mask will be 0b11 and the autotrigger mask 0b11.
- for boards 1, 3, 5 and 7, channel 0 is used for trigger but it is not transmitted while channel 1 is transmitted and used for trigger. This correspond to a readout mask of 0b01 and an autotrigger mask of 0b11.

It must be noted that the maximum data throughput is always around 5 Gbps per link. However, since the number of sent BCs decreases from 3 to 2 when a channel is not enabled for triggering and not all channels are sent, the maximum trigger rate can vary in each configuration accordingly to table 3.1. The worst case scenario (with both channels transmitted and used as trigger) has a maximum trigger rate of ≈ 2100 kHz. The data throughput was measured using the roc-bench-dma tool for a board

	ALR	Readout	Autotrigger	DATA	Data loss
00	1500	0b11	0b11	3.62	NO
01	1900	0b11	0b11	4.48	NO
02	2000	0b11	0b11	4.70	NO
03	2100	0b11	0b11	4.94	NO
04	2200	0b11	0b11	5.08	YES
05	2100	0b10	0b10	2.49	NO
06	4000	0b10	0b10	4.70	NO
07	4050	0b10	0b10	5.08	YES
08	2100	0b01	0b10	1.66	NO
09	4200	0b01	0b10	3.49	NO
10	5000	0b01	0b10	3.92	NO
11	5500	0b01	0b10	4.48	NO
12	5750	0b01	0b10	5.08	YES
13	2100	0b11	0b00	3.30	NO
14	2900	0b11	0b00	4.82	NO
15	3100	0b11	0b00	5.08	YES

Table 3.1 Single link throughput performance evaluation; ALR= Alice Trigger Rate [kHz]; DATA in [Gbps]

with two links and then the result was divided by two in order to consider the single

link performance. The data loss was evaluated by monitoring the programmable_full signals coming from all the FIFOs memory of the readout pipeline. It is worth to highlight that:

- in rows 3 and 13 both channels are transmitted. However in the second case, since those channels are not used for triggering, only two BCs are sent. This can be seen in the data throughput. In fact, doing the arithmetic it results that $\frac{4.94\,Gbps}{3} \cdot 2 \approx 3.30\,Gbps$, as expected.
- in row 3 both channels are transmitted and used for triggering, while in row 5 only one channel is transmitted and used for trigger. Doing the arithmetic $\frac{4.94\,Gbps}{2} \approx 2.49\,Gbps$, as expected.
- in rows 5 and 08 only one channel is transmitted, in the first case it is used to trigger, thus 3 BCs are sent. In the second case it is not used as trigger, so only 2 BCs are transmitted. Doing the math $\frac{2.49\,Gbps}{3} \cdot 2 \approx 1.66\,Gbps$, as confirmed by the test.
- in rows 8 and 13 no channel is nor being transmitted and used as trigger, but in the second case the number of sent channels is two times larger. Doing a short calculation $\frac{3.30\,Gbps}{2} \approx 1.66\,Gbps$.

It appears that in the best case scenario, with only one channel transmitted and the other one used for triggering, the maximum achievable trigger rate is $\approx$ 5500 kHz. In the worst case scenario, with both channels transmitted and used for trigger, the maximum trigger frequency is $\approx$ 2100 kHz. This achieves the designed safety factor with respect to the maximum expected trigger rate of 1200 kHz

3.1.4 Data throughput calculation

In order to predict the behaviour of the firmware under data taking conditions, a way to calculate the expected data throughput was developed. Starting from the assumption that:

$$Throughput = 128b * (Na + Nb * \frac{80b}{128b}) * \frac{1000}{1/11.2455kHz} * \frac{1}{1000000} * \frac{1}{8} \quad (3.1)$$

This corresponds to:

$$Throughput = 128 * (Na + Nb * \frac{80b}{128b}) * \frac{11245.5}{8000000} MBps \tag{3.2}$$

with 128b equals to the number of bits for each transmitted word, N is the number of words sent for each orbit, $\frac{1000}{1/11.2455kHz}$ is the conversion factor to bit/s, $\frac{1}{1000000}$ is the conversion factor from bit to Mb and $\frac{1}{8}$ is the conversion factor from bits to bytes. Na being the number of RDH words and Nb the number of data words. All the constants can be reduced to a single value. Therefore, by knowing the number of sent words by the i^{th} board, its throughput can be calculated with equation 3.3:

$$Throughput_i = \frac{128b}{711.4\frac{b}{MBps}} * (Na_i + Nb_i * \frac{80b}{128b}) \tag{3.3}$$

Given that the current setup has 8 readout board, the complete formula for the throughput calculation becomes:

$$Throughput_i = \sum_{i=0}^{7} \frac{128b}{711.4\frac{b}{MBps}} * (Na_i + Nb_i * \frac{80b}{128b}) \tag{3.4}$$

For a usable equation Na_i and Nb_i must be divided into components. Na_i, the number of RDH words, it has a minimum fixed number of 8 per link, due to the RDH protocol (4 at the beginning of each packet and 4 at the end), plus a R_i factor which keeps track of the extra RDH words used when an orbit has more than 508 data words. It has to be noted that in the CRU each RDH word becomes a 128 bit message, thus the different conversion factor ($80b/128b$). Nb_i counts all the data words. Those can be divided into 24 for baseline throughput (6 words for the first BC and 6 words for the last BC over 2 link); n_i is the number of trigger for each orbit; c_i is the coefficient for the number of word send for each trigger bit for the link 0, for boards 0,1,2,3,5,7 it is 9 and for boards 4,6 it is 18; k is the coefficient for the number of word send for each trigger bit for the link 1, this value is always 12; ϕ_i is a noise compensation parameter; ω_i is used for considering the transmission efficiency gain when the filling scheme has dense trains. Putting everything together

it is obtained:

$$\begin{aligned} Throughput_i = &\sum_{i=0}^{7} \frac{128b}{711.4\frac{b}{MBps}} * (16 + R_i) + \\ &\sum_{i=0}^{7} \frac{80b}{711.4\frac{b}{MBps}} * (24 + c_i n_i + k n_i + \phi_i - \omega_i) \end{aligned} \tag{3.5}$$

Considering $\phi_i = (c_i + k) * noise_i$:

$$\begin{aligned} Throughput_i = &\sum_{i=0}^{7} \frac{128b}{711.4\frac{b}{MBps}} * (16 + R_i) + \\ &\sum_{i=0}^{7} \frac{80b}{711.4\frac{b}{MBps}} * (24 + c_i * (n_i + noise_i) + 12 * (n_i + noise_i) - \omega_i) \end{aligned} \tag{3.6}$$

R_i can be calculated as:

$$R_i = \left(int\left(\frac{c_i * n_i}{508}\right) + int\left(\frac{12 * n_i}{508}\right)\right) * 4 \tag{3.7}$$

The ω_i coefficient can slightly decrease the throughput, however, given the negligible effect and the computation difficulty, for the moment it was neglected in first approximation.

$$\begin{aligned} Throughput_i = &\sum_{i=0}^{7} \frac{128b}{711.4\frac{b}{MBps}} * \left(16 + \left(int\left(\frac{c_i * n_i}{508}\right) + int\left(\frac{12 * n_i}{508}\right)\right) * 4\right) + \\ &\sum_{i=0}^{7} \frac{80b}{711.4\frac{b}{MBps}} * (24 + c_i * (n_i + noise_i) + 12 * (n_i + noise_i)) \end{aligned} \tag{3.8}$$

The accuracy of the formula was validated with values measured in different data taking conditions; in table 3.2 a summary.

3.1.5 Baseline noise evaluation

In order to properly evaluate a correct threshold for the autotrigger algorithm, the standard deviation of the signal baseline was measured; this is shown in table 3.3, where it can be noticed that:

Calculated	Measured	Notes
52 MBps	54 MBps	Technical run low rate
78 MBps	79 MBps	Calibration run
149 MBps	150 MBps	Baseline run
307 MBps	304 MBps	VdM run instantaneous rate

Table 3.2 Comparison between calculated and measured data throughput for different run types.

board number	ch0	ch1	ch2	ch3
0	ZNA TC	ZNA SUM	ZNA T1	ZNA T2
	2.415	/ / /	1.926	1.921
1	ZNA TC	ZNA SUM	ZNA T3	ZNA T4
	/ / /	4.191	1.783	1.851
2	ZNC TC	ZNC SUM	ZNC T1	ZNC T2
	2.400	/ / /	1.876	1.834
3	ZNC TC	ZNC SUM	ZNC T3	ZNC T4
	/ / /	4.044	1.725	1.777
4	ZPA TC	ZEM 1	ZPA T1	ZPA T2
	3.242	3.169	1.865	1.762
5	ZPA TC	ZPA SUM	ZPA T3	ZPA T4
	/ / /	4.401	1.961	2.907
6	ZPC TC	ZEM 2	ZPC T3	ZPC T4
	3.137	2.863	1.798	1.737
7	ZPC TC	ZPC SUM	ZPC T1	ZPC T2
	/ / /	3.977	2.970	1.908

Table 3.3 Standard deviation for the baseline level in ADC units. Red name indicates triggering channel. Run number: 524661. Run duration: 1352 min. Signal events: 52908288.

- the SUMs have a larger noise, at around 4 ADC units, compared to the rest of the channels. This is due to the fact that the SUM is formed by adding 4 analog signals together with a Phillips FIFO NIM module, increasing thus the electronic noise.

- the towers, except for ZPA T4 and ZPC T1, all have a standard deviation lower than 2 ADC units. This is because in this case only one stage of CAEN FIFO is used, while for ZPA T4 and ZPC T1 the two stage setup with both the CAEN and Phillips FIFO is used.

- the channels for the common PMTs and ZEMs use a single stage of Phillips FIFO which have an higher configurable offset, but also a worser performance in terms of noise. The standard deviation of the baseline is around 3 ADC units with the exception of ZNA TC and ZNC TC which have a 4 dB attenuator that affects also the baseline standard deviation.

These measures were essential during the first runs and allowed a first estimation of a proper set of threshold to be used during the following periods of data taking.

3.2 The 2022 pp LHCf Run

On September 23^{rd} 2022 the first runs with beam of the new readout system were performed during the LHCf special run period. The readout system at the time was not completely finalized; in particular it was observed that:

- it was realized that the time calibration procedure was far more important and complex than estimated. At the time, the calibration procedure didn't have a reference signal and was based on the physics signals alone. This resulted in a series of problems, the first and most important one being that the calibration had to be performed after the stable beam declaration and thus loosing a portion of data taking time. During the two days of data taking the calibration procedure was performed by manually setting the correct delay value.

- it was discovered a bug in the filter module that interfered with the number of data packets sent. This was then fixed in the next weeks with a new firmware version.

- a bug was also present in the reconstruction software; this caused the majority of time frames to be discarded during reconstruction. This was fixed in the next weeks by updating the online code used in reconstruction.

In total more than 75 runs were acquired over the span of 4 fills in order to test different parameter configurations. The majority of the runs lasting less than 10 minutes. Most of the runs were performed over the span of a single long fill. In fact, fill n°8178 is the longest fill of the LHC to date, lasting 57.4 hours (this was

possible thanks to the conditions of the collisions which allowed a particularly low degradation of the beams). In figures 3.2 and 3.3 the ZDC raw data signal shapes of run 526174 are shown for the two neutron common photomultipliers. On the X axis the sample number and on the Y axis the amplitude in ADC units.

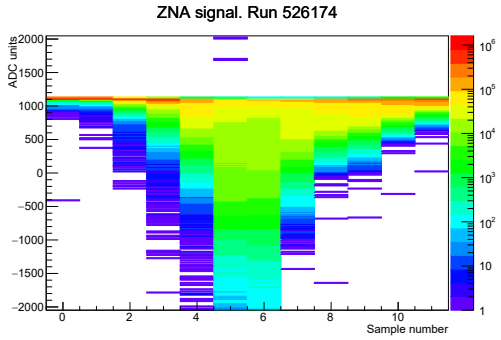


Fig. 3.2 Run 526174 waveform ZNA TC.

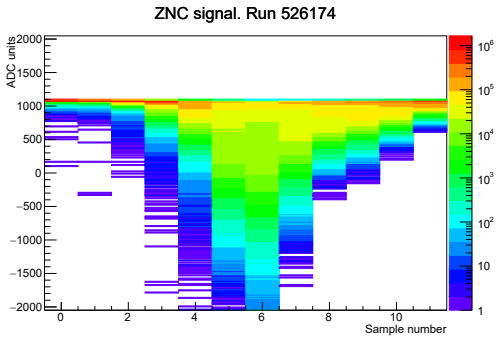


Fig. 3.3 Run 526174 waveform ZNC TC.

3.3 The 2022 pp Van der Meer scans

On November 9^{th} 2022, the first VdM [92] scan of Run 3 was performed with proton beams of 6.8 TeV. Given the special beam conditions during this type of data taking, in particular the limited crossing angle values of +73 μrad, the ZDC was able to acquire data in order to proceed with further commissioning. During this period the timing calibration was still performed manually at the beginning of each fill. The firmware and software bugs were fixed and the acquisition parameters were better tuned with respect to the LHCf runs. In figures 3.4 and 3.5 the ZDC raw data signal shapes of run 529039 are shown for the two neutron common photomultipliers. To be noted that the peak of the signal (considering the negative polarity) is centered on sample number 6.

3.4 The 2022 Pb–Pb low intensity Run

During August 2022 LHC sustained a 4 weeks long RF issue [93]. This, in addition with the worldwide energy crisis, forced the LHC committee to reshuffle the data taking schedule. The result was a shift of the first main Pb–Pb data taking to fall

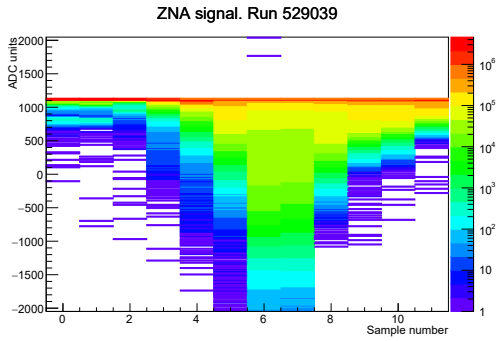


Fig. 3.4 Run 529039 waveform ZNA TC.

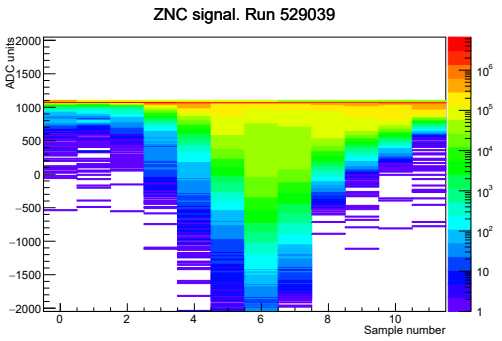


Fig. 3.5 Run 529039 waveform ZNC TC.

2023 and the decision to perform just two commissioning fills at low intensity for 2022. This period started November 18^{th} 2022. LHC delivered two fills; the first (8412) with single bunch injections, meanwhile the second (8413) was used by the LHC machine to validate the 50 ns Pb trains:

- fill n° 8412 with filling scheme Multi_20b_7_16_7_3bpi_8inj_PbTest2022
  Beta*=10m - L3/Dip +/+
  Effective crossing angle= -72(internal) +172(external) = +100 μrad
  Beam duration = 02:33:08

- fill n° 8413 with filling scheme 50ns_24b_8_24_5_8bpi_10inj_PbPbtrains
  Beta*=10m - L3/Dip +/+
  Effective crossing angle= -72(internal) +128(external) = +56 μrad
  Beam duration = 09:14:42

During both fills the ZDC detector was used as luminomiter for the ALICE experiment, providing the event rate of IP2 as a feedback for the LHC operation team. After some short standalone runs used for timing calibration, still performed manually at this stage, the ZDC was able to join global data taking. The ALICE experiment managed a good efficiency performing just one global run (a data taking run with all the detectors enabled) for each fill:

- Run 529403, taken during fill n° 8412 with average event rate of 718 Hz and 53 min duration which acquired 2.3 Mevents of EMD + hadronic on the most exposed calorimeters.

- Run 529418 during fill n° 8413 with average event rate of 770 Hz and 8 h and 29 min duration which acquired 23.5 Mevents of EMD + hadronic on the most exposed calorimeters (ZNA TC and ZNC TC).

In this section the data from run 529418 will be shown.

3.4.1 Channel transmitted and baseline evaluation

In figure 3.6 a 2D plot with the number of transmitted words for each channel. On the X axis the board number, while on the Y axis the channel number. From the colors it can be noted that the channels that send the most words are the ZN calorimeters (ZNA TC, ZNA SUM, ZNC TC, ZNC SUM). This is due to the difference in cross section from the neutron and proton calorimeters and from the different readout configuration of each channel; the TC and SUM (which are configured for triggering) send three BCs for each event, while the other towers only two. The channels that send less data are the one which are not configured for data transmission since they only send BC 0 and BC 3563. From the patterns of the plot it is easy to spot issues with the data taking. In figure 3.7 a summary of the baseline sent by the fee is presented. On the X axis is shown the channel name and on the Y axis the respective baseline value. This is useful for spotting issues with the cabling of the analog signals or the faults in the analog chain.

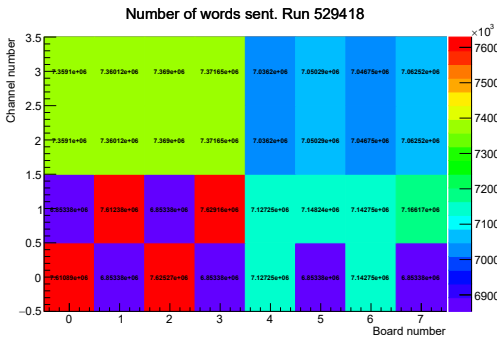


Fig. 3.6 Run 529418 Channels Transmitted.

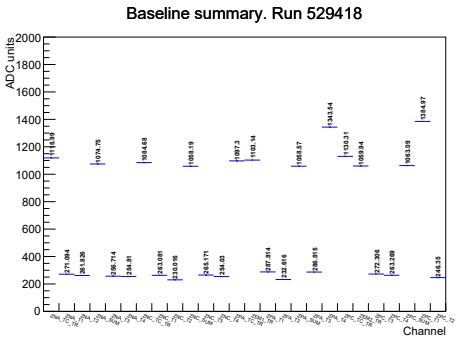


Fig. 3.7 Run 529418 Baseline Summary.

3.4.2 ZN trigger scheme

In figures 3.8 and 3.9 a 2D histogram with the number of triggers (ALICE trigger or autotrigger) for each BC is shown. The bunch crossing value of each pixel is given by the value of the X axis plus the value of the Y axis multiplied by -100. For example, for the pixel (12;-15) the respective BC is number 1512. From the plot the two Pb trains with 50 ns bunch spacing can be seen (the two lines with red squares at the center of the plot). It can also be observed that side C acquired more noise. This histogram is crucial when checking the delay coarse alignment; the check is done by comparing the entries in each bin with the highest number of entries with the known filling scheme provided by the LHC.

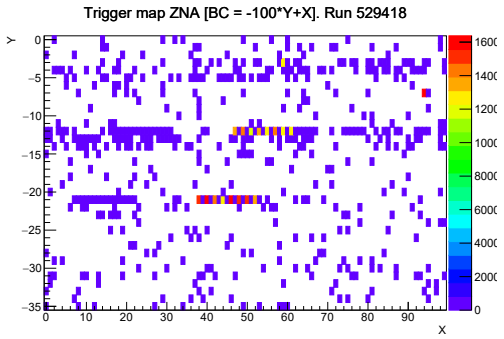


Fig. 3.8 Run 529418 ZNA TC trigger.

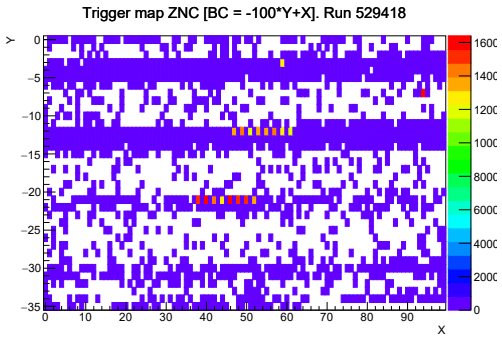


Fig. 3.9 Run 529418 ZNC TC trigger.

3.4.3 ZN common PMT signals

In figure 3.10 the digitized analog signal from ZNA TC is shown, while in figure 3.11 a zoom showing in detail the 1 neutron and 2 neutrons signals is reported. On the X axis the sample number is reported, while on the Y axis there is the amplitude value in ADC units. The single neutron peak has an amplitude of $\approx$ 27 ADC units, which corresponds to $\approx$ 8 mV; this is true for both sida A and side C. In figure 3.12 the digitized analog signal from ZNC TC is shown, while in figure 3.13 the zoom on the 1 neutron and 2 neutrons signals is reported. On the X axis the sample number, while on the Y axis the amplitude value in ADC units are reported.

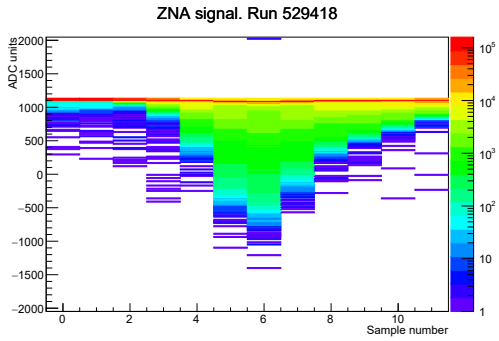


Fig. 3.10 Run 529418 ZNA TC signal.

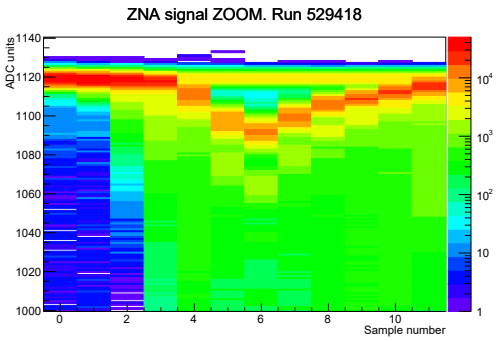


Fig. 3.11 Run 529418 ZNA TC signal zoom.

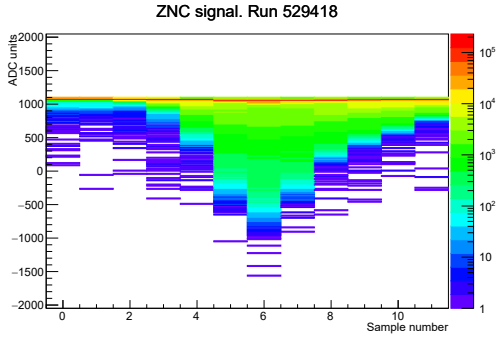


Fig. 3.12 Run 529418 ZNC TC signal.

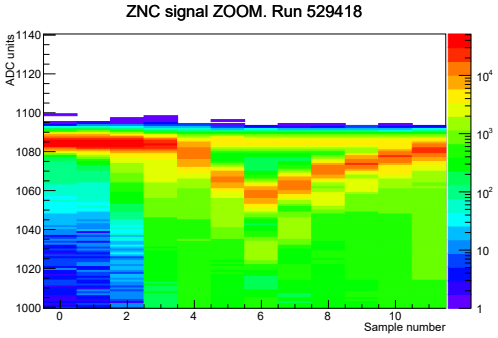


Fig. 3.13 Run 529418 ZNC TC signal zoom.

3.4.4 ZN energy spectra

Starting from the concept that the signal is aligned in such a way that the peak of the waveform is in sample number 6, then, the integral of the portion of the waveform with the highest signal to background ration can be obtained by summing the values from sample 4 to sample 8. This value can be plotted on an histogram and (after energy calibration) the energy spectra of the calorimeter is obtained. In figure 3.14 the uncalibrated spectra for ZNA TC, while in figure 3.15 the spectra for ZNC TC is reported. In both histograms the 1 neutron peak is clearly visible, being the one with less energy and the most entries, as well the other peaks up to 5 neutrons are distinguishable. By performing a fit with an asymmetrical Gaussian function and constraining the 2 neutrons and 3 neutrons peaks to respectively 2 and 3 times the value obtained by fitting the 1 neutron peak, the position and resolution of the 1 neutron signal can be calculated. In figure 3.16 the fit results for ZNA TC and in figure 3.17 the results for ZNC TC are reported together with the values of the fit parameters. From those parameter a preliminary resolution of the 1 neutron peak

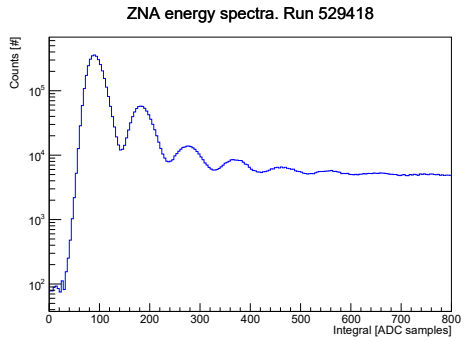


Fig. 3.14 Run 529418 ZNA TC energy spectra (log scale).

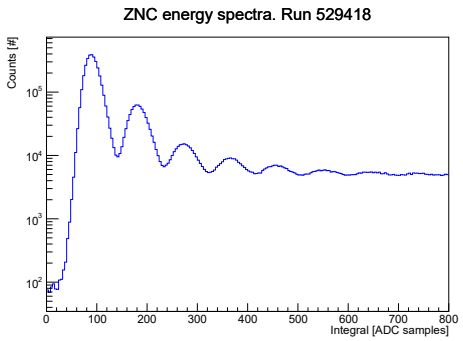


Fig. 3.15 Run 529418 ZNC TC energy spectra (log scale).

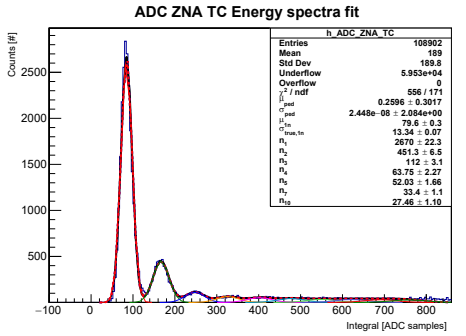


Fig. 3.16 Run 529418 ZNA TC energy spectra fit.

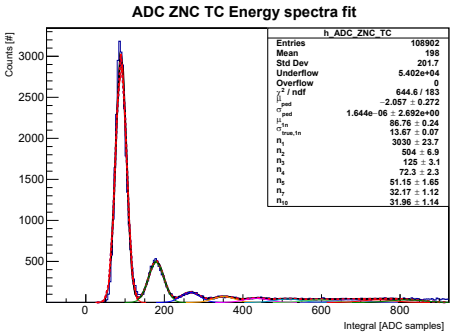


Fig. 3.17 Run 529418 ZNC TC energy spectra fit.

can be calculated as :

$$\sigma_{1n}/\mu_{1n} = 0.17\, for ZNA\, TC$$

$$\sigma_{1n}/\mu_{1n} = 0.16\, for ZNC\, TC$$

This value was $\approx 20\%$ in Run 1 and Run 2, thus the new readout system allows a better allows a better resolution thanks to the new digitization, even without chnging the detecting hardware.

3.4.5 ZN centroids

Using the signal from the 4 ZN towers, the average impact point of the beam on the front face of the neutron calorimeters can be calculated. This is done by computing the weighted average of the tower position relative to the center of the calorimeter

using as weight the amplitude of the signal with the formula shown in equation 3.9.

$$x_c = \frac{\sum_{i=1}^{4} PMT_i \cdot x_i}{\sum_{i=1}^{4} PMT_i} \quad y_c = \frac{\sum_{i=1}^{4} PMT_i \cdot y_i}{\sum_{i=1}^{4} PMT_i} \tag{3.9}$$

$$x_i = (-1.75, 1,75, -1,75, 1,75) \quad y_i = (1.75, 1,75, -1,75, -1,75)$$

The results are reported in figures 3.18 for side A and 3.19 for side C. On X and Y axis values are in cm. It appears that the beam on side C was half a centimeter shifted to the left, while on side A it was perfectly centered.

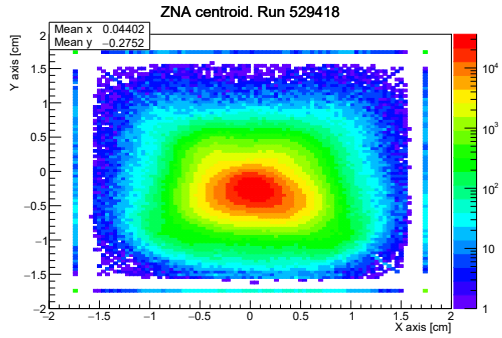


Fig. 3.18 Run 529418 ZNA centroid.

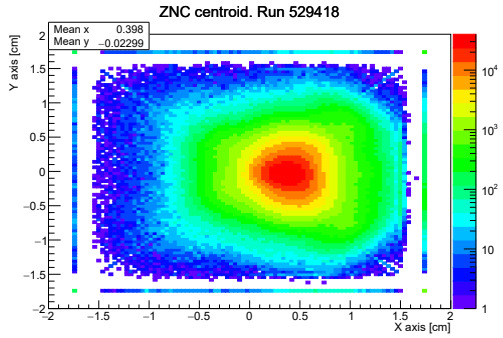


Fig. 3.19 Run 529418 ZNC centroid.

3.4.6 Separating ZN nominal interactions

In figure 3.20 a 2D histogram that has:

- on the X axis the sum of the arrival time of ZNC TC and ZNA TC minus a constant in ns; this is related to the event time.
- on the Y axis the difference of the arrival time of ZNC TC and ZNA TC in ns; considering that the speed of the neutrons is approximately c this can be related to the z position of the collision.

The center cluster, in red, represents the collisions occurring in the nominal interaction point, while on the diagonals the parasitic interactions are displayed. The LHC RF work at 400 MHz and nominally only one out of 10 buckets should contain a packet [94]. The parasitic interactions are caused by some ions falling in the adjacent RF buckets with respect to the nominal ones. This histogram is extremely useful for

the ALICE data analysis, since, a cut based on this histogram is applied to separate the signal from beam-beam interactions from the noise.

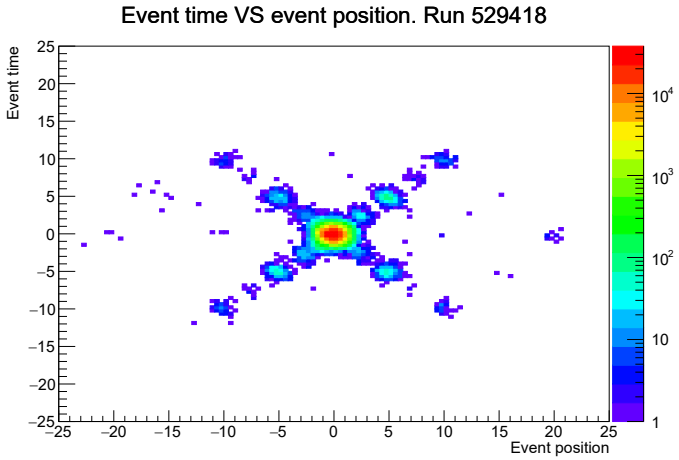


Fig. 3.20 Run 529418 nominal interactions.

3.4.7 Pb–Pb 2022 ALICE data taking team

In figure 3.21 is shown a group picture of the ALICE team the evening of the first Pb–Pb stable beams of Run 3. I am on the top row, center right.

Fig. 3.21 ALICE crew group photo the evening of the first Pb–Pb stable beams of Run 3. Figure taken from [95]

3.5 The 2023 pp Van der Meer scans

The first data taking period including the ZDC of 2023 started June 28^{th} with a VdM scan and lasted two days. During this period the ZDC acquired over 30 runs, the majority of them lasting a few minutes at the beginning of the first fill for commissioning purposes and the remaining ones lasting several hours, acquiring data in global run with the other ALICE detectors. The longest run (538964) lasted 06:24:24. In figures 3.22 and 3.23 the waveform signals from ZNA TC and ZNC TC are shown. During this period the automatic ZDC calibration procedure was

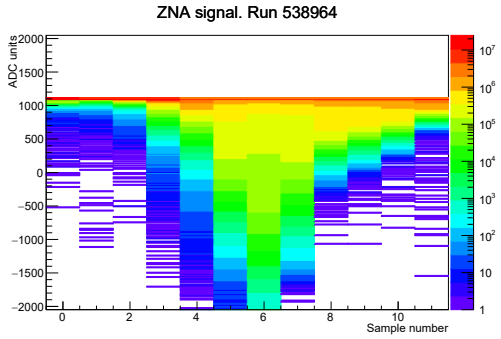


Fig. 3.22 Run 538964 waveform ZNA TC.

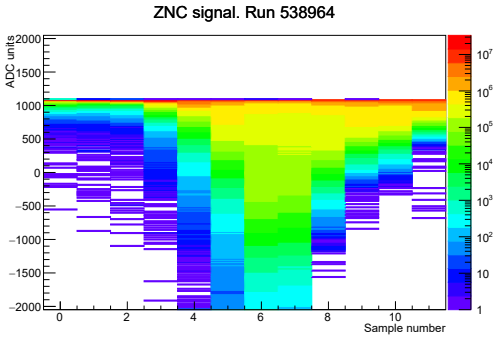


Fig. 3.23 Run 538964 waveform ZNC TC.

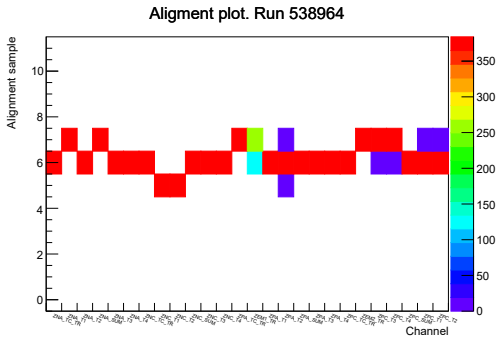


Fig. 3.24 Run 538964 alignment plot.

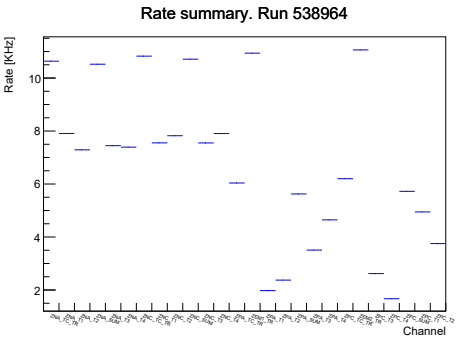


Fig. 3.25 Run 538964 rate summary plot.

successfully tested with physics data for the first time. A new 2D histogram was developed in order to check in real time the calibration status of each channels. This can be seen in figure 3.24. On the X axis the channel name is shown, while on the Y axis the average peak sample of the waveform is reported. Each minute a new value is added to that histogram. Ideally the result would be to have every channel locked stably on channel 6, as it happens when calibrating the detector with a reference

signal. This is shown in figure 3.26. However, the physics data is much less stable than the reference, thus, for pp and Pb–Pb data alignment values between 5 and 7 are considered good. Using the information of the number of events for each orbit sent by the FEE, a new plot was added which displays the average event rate for each channel during the run. This is shown in figure 3.25; with the ZDC channels reported on the X axis and the value of average event rate in kHz on the Y axis.

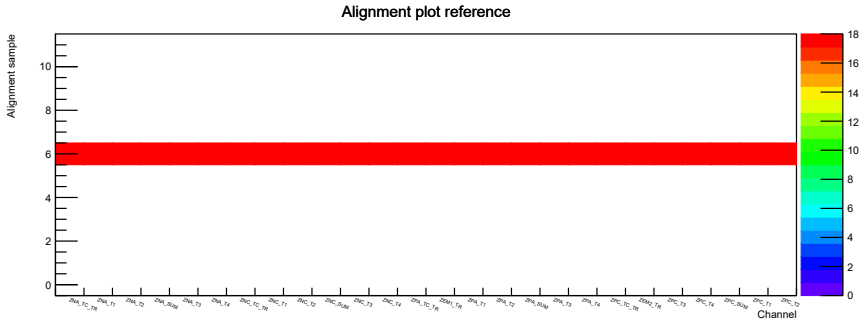


Fig. 3.26 ZDC ideal alignment plot using the reference signal for calibration.

The second data taking period of 2023 started on September 6^{th} with another VdM scan. At this point the operational parameters and calibration procedure were already tested, thus less commissioning runs were performed. Over the span of one fill the ZDC took 8 runs with the longest one (542756) lasting 02:32:22. To be noted that in

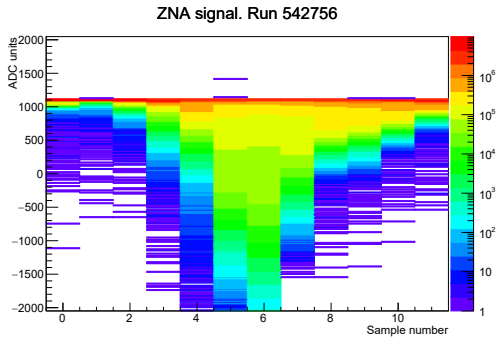


Fig. 3.27 Run 542756 waveform ZNA TC.

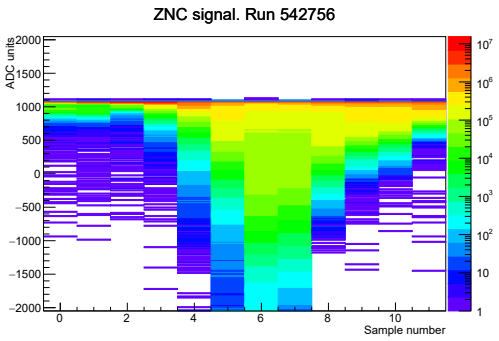


Fig. 3.28 Run 542756 waveform ZNC TC.

this case the calibration procedure, the results of which are reported in figure 3.29, did a much better job with respect to previous period, keeping the majority of the signals on sample 6. On the other hand the average rate, shown in figure 3.30, was lower with respect to run 538964 reported in figure 3.25; ≈6 kHz with respect to ≈10 kHz for the ZNs. Meanwhile, in figures 3.27 and 3.28 the waveform shapes from ZNA TC and ZNC TC are displayed.

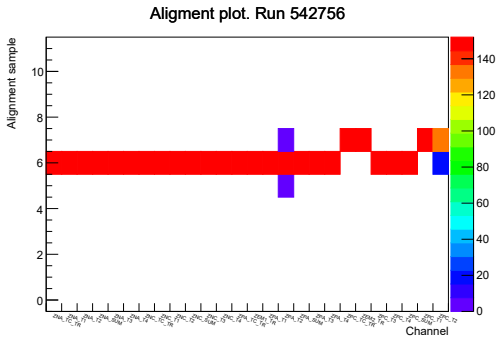


Fig. 3.29 Run 542756 alignment plot.

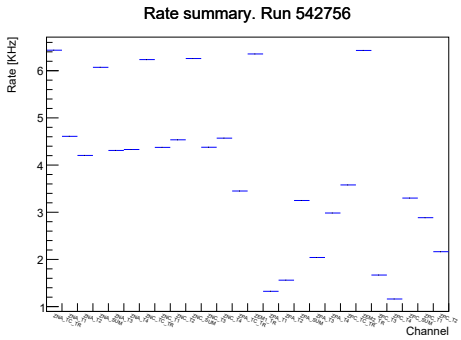


Fig. 3.30 Run 542756 rate summary plot.

3.6 The 2023 Pb–Pb Run 3 data taking and VdM

The evening of September 26^{th} 2023 officially the first Pb–Pb data taking campaign of Run 3 started and was carried out until October 29^{th}. During this five week period the number of collisions recorded by ALICE has been 40 times greater than the total amount recorded by the experiment in its previous periods of heavy-ion data taking, from 2010 to 2018 [96]. Thanks to the upgrades of the LHC machine and the new readout capabilities of the ALICE experiment, it was reached an energy of 5.36 TeV per pair of nucleons and a collision rate six times higher with respect to Run 1 and Run 2. The first days of the heavy-ion period were dedicated to a smooth ramp-up of

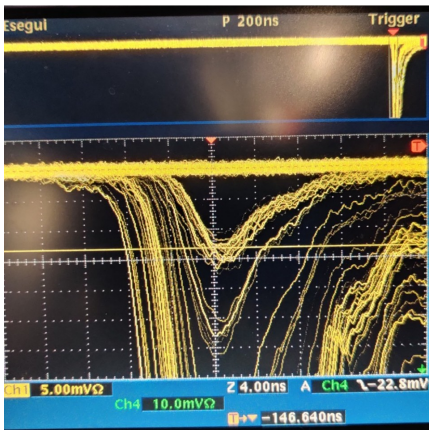


Fig. 3.31 Signal from ZNC TC seen with the oscilloscope during the time calibration performed the first night of Pb–Pb data taking.

Fig. 3.32 The ALICE data taking crew waiting in ACR for the first stable beam declaration of the 2023 Pb–Pb data taking campaign.

the number of bunches in the accelerator.

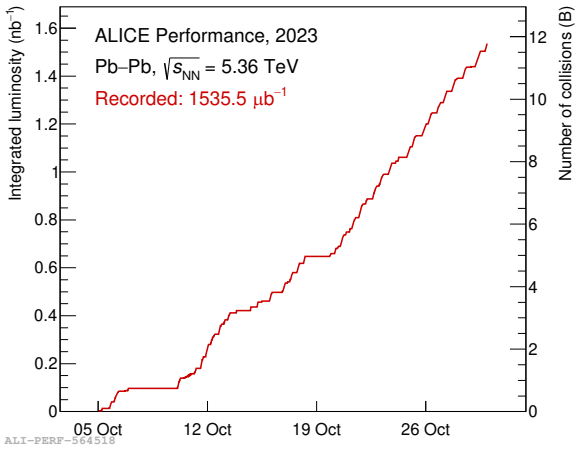


Fig. 3.33 The number of Pb–Pb collisions collected by ALICE in 2023 (beam background excluded), expressed in terms of the cumulative number of collisions (right vertical axis) and a related quantity called integrated luminosity (left vertical axis). Figure taken from [96]

3.6.1 The background studies

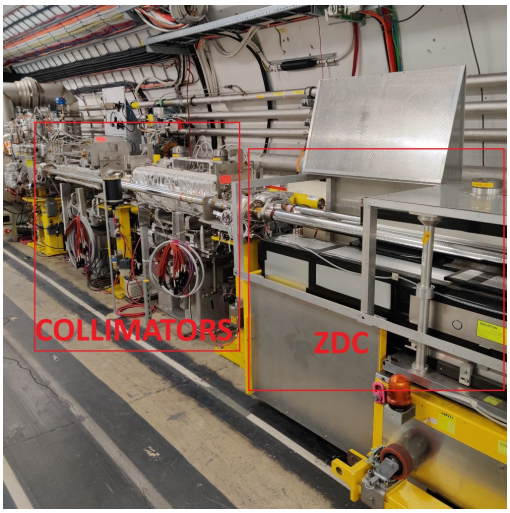


Fig. 3.34 The ZDC detector on side A and collimators.

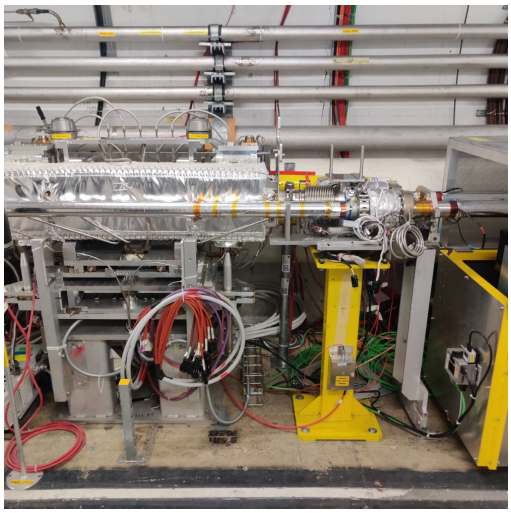


Fig. 3.35 Close up of the collimators placed before the ZDC on side A.

After the first runs it was clear that some background noise was present at P2, affecting in particular the signals of the inner layer of ITS and the ZDC. The first week of data taking was thus dedicated to the study of this problem. The particles circulating in the LHC do not all have the same energy; some particles in the beam have a slightly lower or higher energy than the nominal energy of all the particles. A beam of particles thus has an average energy and an energy spread. The beam is

curved by the LHC dipoles, which guide it along the 27-km-long accelerator, but the radius of curvature is different for particles with different energies. Therefore, the physical size of the beam at the exit of the dipoles will depend on the energy spread of the beam. This is what is called dispersion. In order remove from the beam the particles with large dispersion collimators are used. During Pb–Pb operation the collimators are also used to remove Pb ions that have incorrect Z/A ration due to processes that result in a loss of one or more neutrons or pick up of an electron. One of those collimators is placed right before the ZDC on side A, this can be seen in figures 3.34 and 3.35 and 3.36. The particles hitting the jaws of the collimator were producing some particles traveling nearly at zero degree which were hitting both the ZDC on side A and a section of the ITS. In figures 3.37 and 3.38 the ZNA TC and

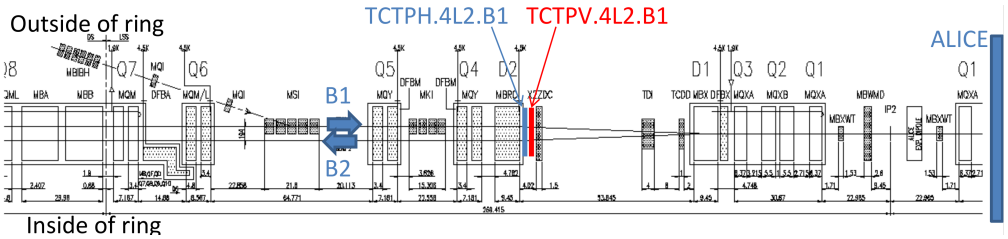


Fig. 3.36 ALICE collimation scheme. Figure taken from [97]

ZNC TC interaction rate during fill 9222 are shown. On the X axis the time, while on the Y axis the rate in Hz are reported. It can be seen how the interaction rate decreases during the fill; this is due to the beams losing intensity. In figure 3.39 the

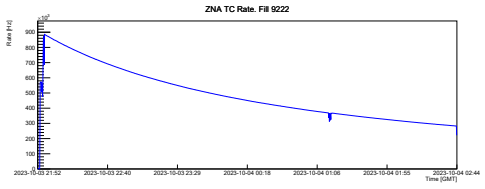


Fig. 3.37 ZNA TC interaction rate during fill 9222.

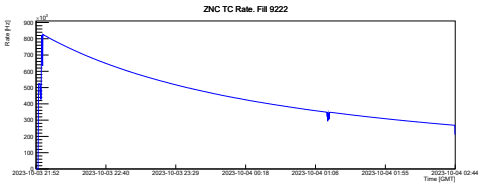


Fig. 3.38 ZNC TC interaction rate during fill 9222.

ratio between the two interaction rates seen before is shown. Ideally this value should be exactly 1. Here it can be seen that ZNA at the beginning of the fill sees ≈ 7 % more events and at the end of the fill this values lowers down to ≈ 5 %. In addition in figure 3.40 the ratio between ZNA TC and ZNC TC during fill 9212 is shown. During the first three quarters of the fill CMS performed its VdM scan, while during the last quarter it was the time of LHCb. From the plot it can be noted that during

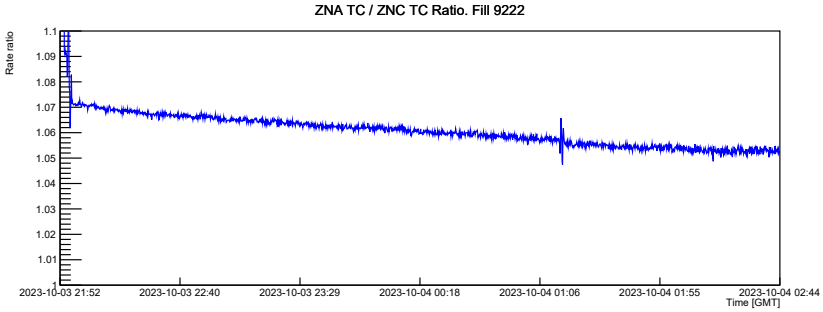


Fig. 3.39 ZNA TC / ZNC TC interaction rate ratio during fill 9222.

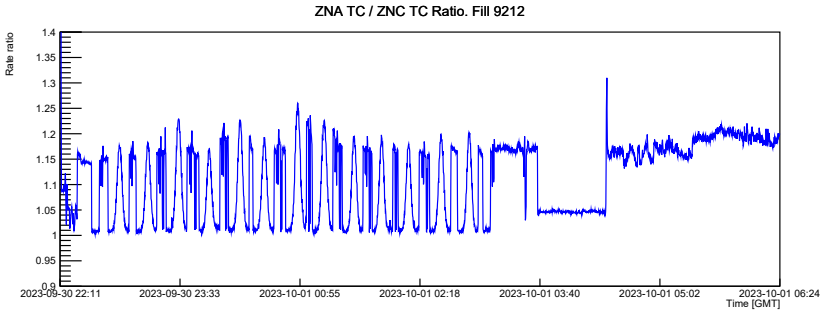


Fig. 3.40 ZNA TC / ZNC TC interaction rate ratio during fill 9212.

the CMS scan the operations performed at point 8 can be seen in the ratios between the two neutron calorimeters; the effect is much less visible during the LHCb scan. Experts from the ALICE experiment and the LHC machine collaborated closely to tackle the issue of background noise in the ALICE detector: many different remedial strategies were studied and tested during several fills over a period of more than 30 hours. Finally, the correction of residual dispersion reduced the background noise to a satisfactory level for ALICE to take physics data [98].

3.6.2 An overview on the data taking runs

After the background mitigation was put in place, the LHC machine restarted the ramp up procedure up to the nominal 50 kHz hadronic collision rate foreseen for Run 3. This resulted in a series of fills with a step by step increasing number of bunches. The ZDC detector took physics data during all of those runs working also as luminomiter of the experiment. In this section the plots from a typical Pb–Pb physics run will be shown; in particular from run number 545345 from October 29^{th} 2023 (fill n°9317) which lasted 8 h and 9 min.

3.6.3 Channel transmitted and baseline evaluation

In figure 3.41 the number of transmitted packets for each channel is shown. On the X axis the board number, while on the Y axis the channel number are reported. The absolute number of entries for each channel is related to the interaction rate and the duration of the Run. For quality control purposes, only the ratios between channels are relevant. In figure 3.42 the average baseline (in ADC samples) of the whole run for each channel is reported. The baseline value for each channel is similar, almost equal, to the values in figure 3.7 of the 2022 Pb–Pb low intensity Run. This means that the analog FIFOs performance and the cabling configuration remained unchanged, as expected.

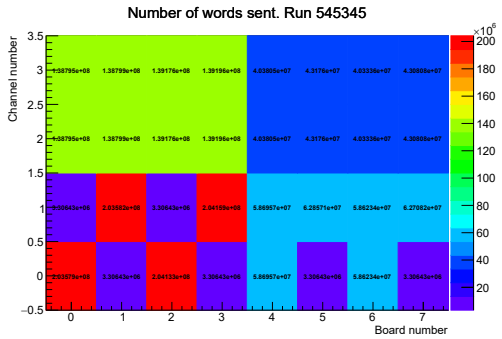


Fig. 3.41 Run 545345 Channels Transmitted.

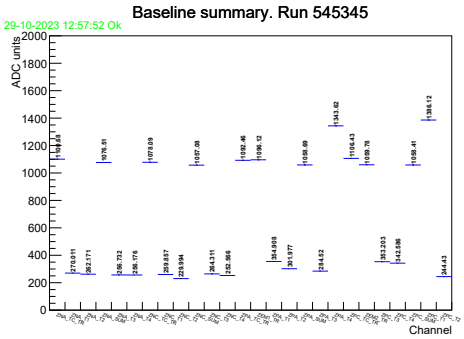


Fig. 3.42 Run 545345 Baseline Summary.

3.6.4 Channel fired and rate summary

In figure 3.43 the number of autotrigger events for each channel is shown. On the X axis the board number, while on the Y axis the channel number are reported. To be noted that the number of event for the eight ZN towers are similar to each other, while the number of events of the ZP towers are decreasing in relation with the proximity of the tower itself with the beam pipe (the closest the tower the higher the number of events). This can be also seen in figure 3.44 where the average rate in kHz (Y axis) shown for the whole run for each channel (X axis) is shown. During this Run the average event rate of the most exposed calorimeters was $\approx$ 450 kHz while during the 2022 Pb–Pb low intensity Run this values was $\approx$ 0.75 kHz.

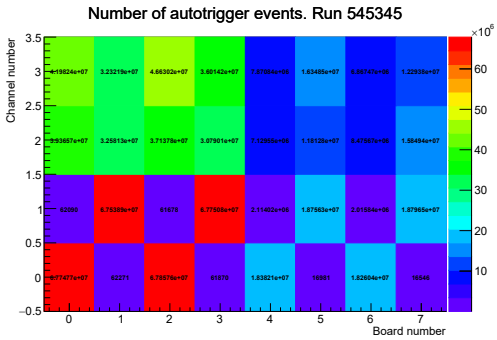


Fig. 3.43 Run 545345 Channels Fired.

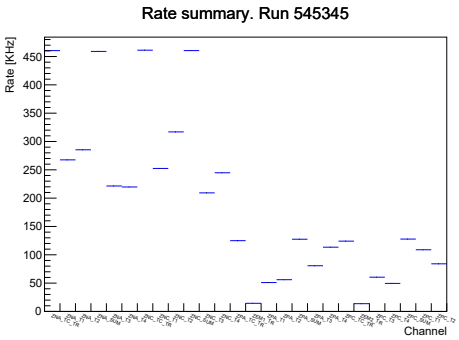


Fig. 3.44 Run 545345 Rate Summary.

3.6.5 Alignment plot

In figure 3.45 the integrated alignment plot of run 545345 can be seen. Using this plot the central data taking crew regularly checks the proper time calibration and stability of the detector during acquisition. Ideally all channels should be aligned on sample 6; a deviation of $\pm$ 1 is accepted; a deviation greater than $\pm$ 1 would result in the run being stopped and the ZDC to be re-calibrated. Each entry represents a minute of data taking, while the run number 545345 lasted $\approx$ 450 min.

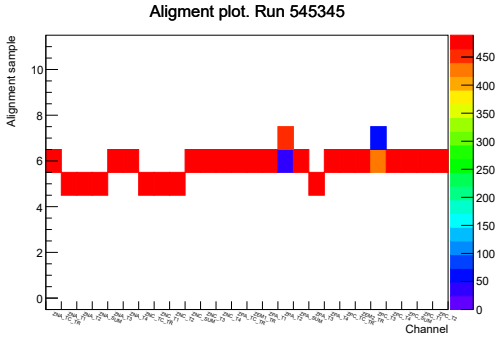


Fig. 3.45 Run 545345 alignment plot.

3.6.6 ZN trigger scheme

In figures 3.46 and 3.47 the trigger map output from the common PMT of the neutron calorimeters of sides A and C are shown. Run number 545345 was performed during fill number 9317 with filling scheme 50ns_1080b_960_960_288_40bpi_27inj_PbPb. The 960 colliding bunches at P2 can be seen in these plots. In addition the two abort

gaps, a ≈ 3 μs long area in the circulating ring the has to be kept free of particles in order to avoid losses during the rise time of the LHC extraction kickers in the beam dump [99], can be spotted. The first one from BC 225 to BC 343, and the second one from BC 2898 to BC 3016.

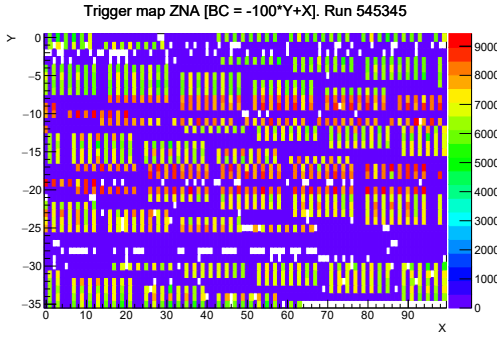


Fig. 3.46 Run 545345 ZNA TC trigger.

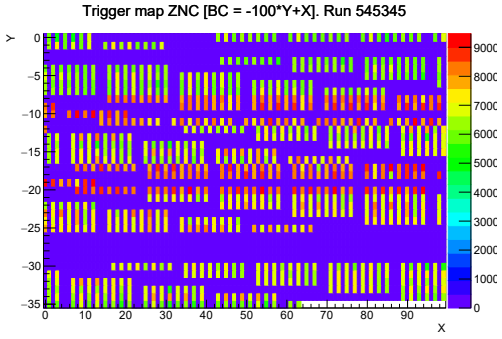


Fig. 3.47 Run 545345 ZNC TC trigger.

3.6.7 ZN common PMT signals

In figures 3.48 and 3.49 the digitized signal from ZNA TC and the zoom on the 1 neutron and 2 neutrons peaks are shown. Meanwhile, in figures 3.50 and 3.51 the

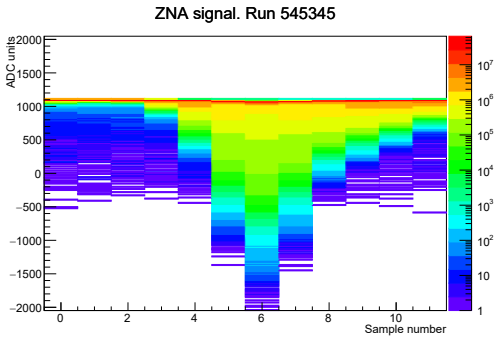


Fig. 3.48 Run 545345 ZNA TC signal.

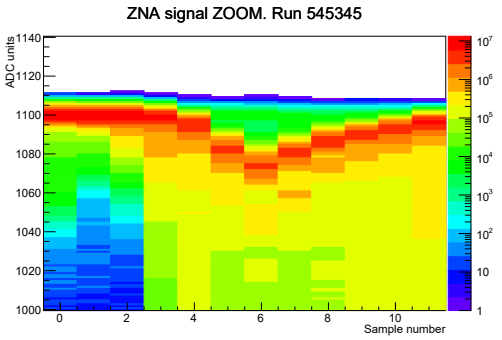


Fig. 3.49 Run 545345 ZNA TC signal zoom.

digitized signal from ZNC TC and the zoom on the 1 neutron and 2 neutrons peaks are displayed. The two sides have a comparable amplitude for both, 1 neutron and 2 neutrons signals, and maximum amplitude; in addition the number of entries is almost equal, aas expected.

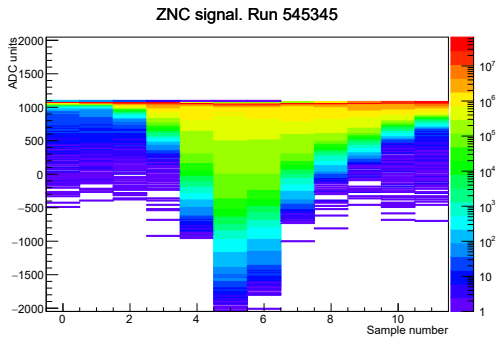


Fig. 3.50 Run 545345 ZNC TC signal.

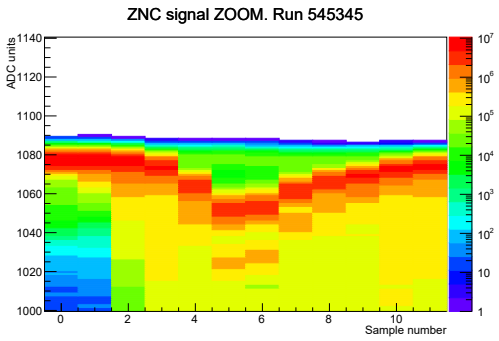


Fig. 3.51 Run 545345 ZNC TC signal zoom.

3.6.8 ZN energy spectra

In figures 3.52 and 3.53 the raw spectra before energy calibration of ZNA TC and ZNC TC obtained by integrating each signal shape from sample 4 to sample 8. The peaks of the 1, 2, 3 and 4 neutrons signals are visible.

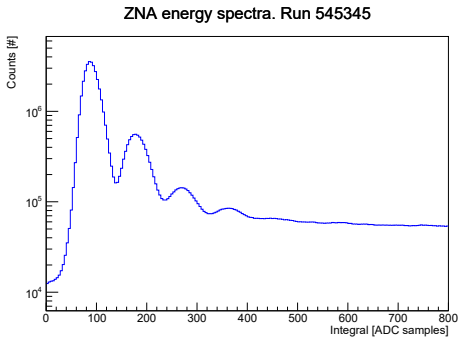


Fig. 3.52 Run 545345 ZNA TC energy spectra (log scale).

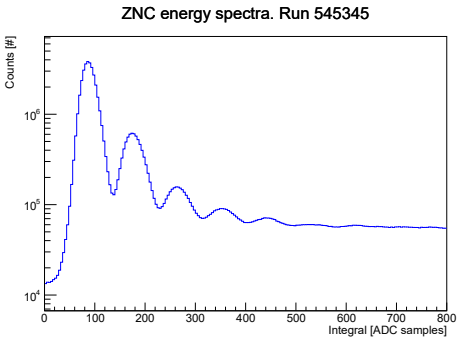


Fig. 3.53 Run 545345 ZNC TC energy spectra (log scale).

3.6.9 ZP common PMT signals

In figures 3.54 and 3.55 the digitized signal from ZPA TC and the zoom on the 1 proton and 2 protons peaks are reported. Meanwhile, in figures 3.56 and 3.57 the digitized signal from ZPC TC and the zoom on the 1 proton and 2 protons peaks are shown. Both size are comparable signal shapes, waveform amplitude and number of entries, as expected.

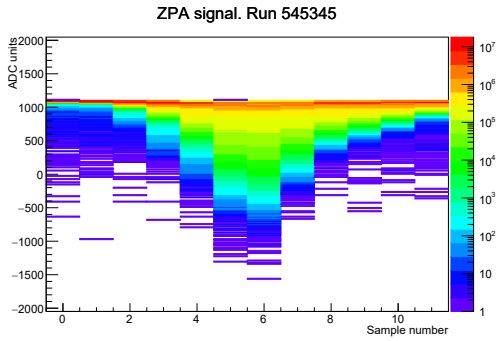


Fig. 3.54 Run 545345 ZPA TC signal.

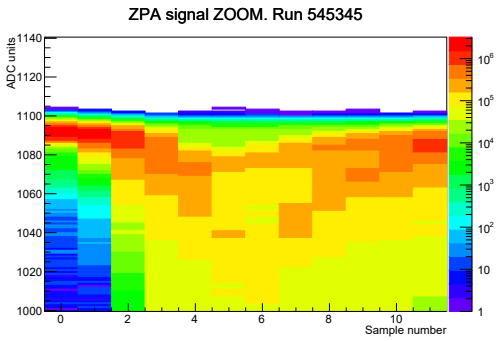


Fig. 3.55 Run 545345 ZPA TC signal zoom.

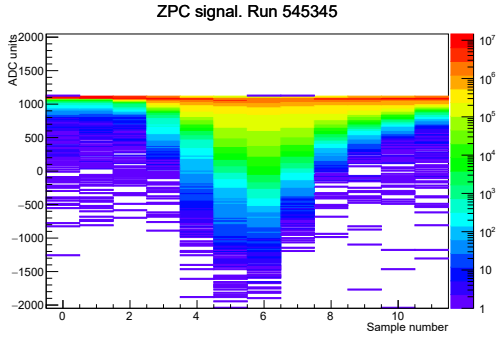


Fig. 3.56 Run 545345 ZPC TC signal.

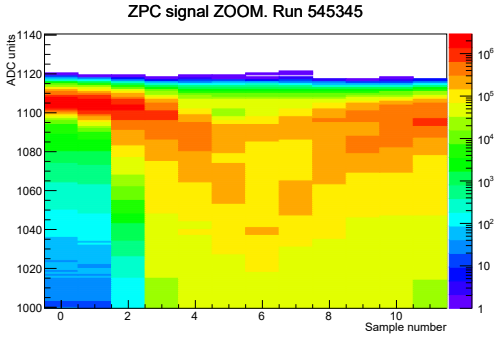


Fig. 3.57 Run 545345 ZPC TC signal zoom.

3.6.10 ZP energy spectra

In figures 3.58 and 3.59 the raw spectra, before energy calibration, of the common proton calorimeters of side A and side C obtained by integrating each waveform from sample 4 to sample 8. The 1 proton and 2 protons peaks are visible. On the X axis the integral in ADC samples is depicted, while one Y axis the number of entries is shown in logarithmic scale. To be noted that during the data taking of Run 1 and Run 2 the performance of the energy resolution of the previous readout system, combined with the lower interaction rate of the time, did not allow to properly discriminate the two protons peak. The new readout system can discriminate the two proton peak and possibly, thanks to the increased statistic of the longer runs taken by ALICE, the three proton peak.

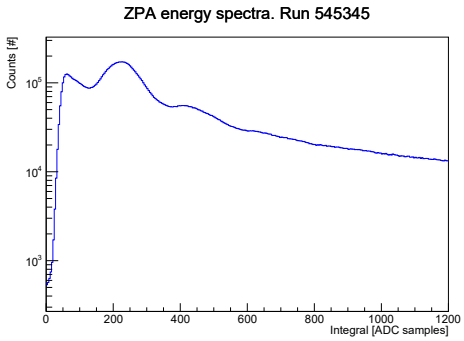


Fig. 3.58 Run 545345 ZPA TC energy spectra (log scale).

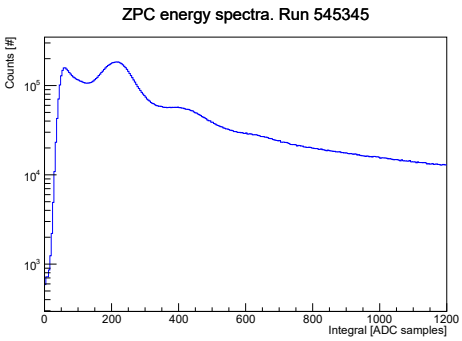


Fig. 3.59 Run 545345 ZPC TC energy spectra (log scale).

3.6.11 ZN centroids

In figures 3.60 and 3.61 the centroid plots for the neutron calorimeters of side A and C are shown. On side A the beam was centered with respect to the X axis, while on side C it was half a centimeter shifted towards right. With respect to the Y axis on both sides the beams seem pointing half a centimeter down, however, this can be corrected by changing the Y position of the two neutron calorimeter moving the platform where they are positioned. To be noted that these plots have been generated before the energy calibration of the four towers of the neutron calorimeters. The position of the average hit point may change after calibration.

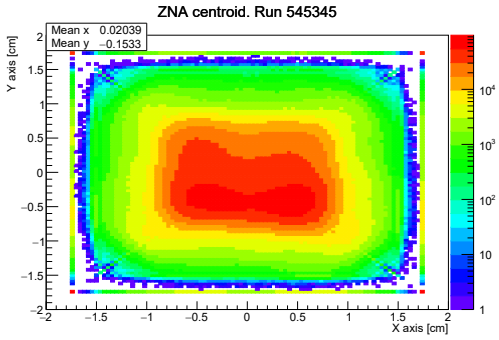


Fig. 3.60 Run 545345 ZNA centroid.

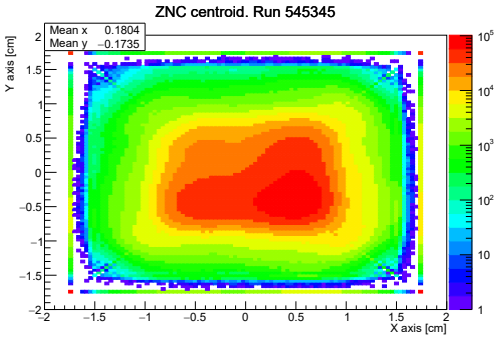


Fig. 3.61 Run 545345 ZNC centroid.

3.6.12 ZN nominal interactions

In figure 3.62 the timing information plot (sum vs difference) for run number 545345 is reported. With respect to the plot in figure 3.20 it can be noted that the much

higher event rate allowed the acquisition of a new type of parasitic interaction. On the two diagonals the main BCs are interacting with the displaced RF buckets, while on the vertical line two different displaced RF buckets are interacting with each other. This can be recorded thanks to the continuous readout system, which, acquiring all types of signals, can save event that occur in the correct position, but shifted in time by any value (in this case shifted by a multiple of 2.5 ns).

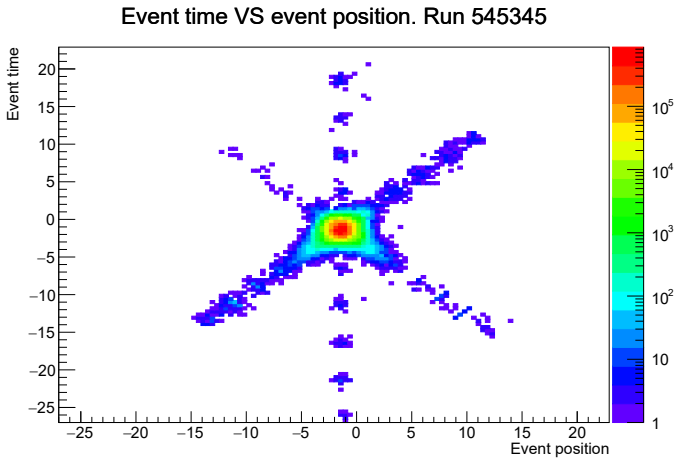


Fig. 3.62 Run 545345 nominal interactions.

3.6.13 The 2023 Pb–Pb Van der Meer scans

The evening of October 10^{th} 2023 the LHC injected fill number 9240 with filling scheme 100ns_432b_204_324_301_28bpi_27inj_VdMPbPb. The total beams duration was 3h 5m 36s. A single run (544434) was performed which lasted 3h 4m 57s, establishing a fill efficiency of 99,65 %. During this run the ALICE experiment performed its VdM scan. In figure 3.63 the interaction rate for ZNA TC during the whole fill can be seen. The different scan phases can be seen; when the beams are completely separated the rate goes to zero, and when the beams are head on the rate goes up to 300 kHz at the beginning of the fill. In figure 3.64 the same can be seen for side C. In figure 3.65 the ratio between the interaction rates of the two sides during the VdM scan. When the beams are completely separated the ratio goes below one, meaning that side C has higher noise. When the beams are colliding the ratio is stable at 1, meaning that the two side of the detector see the same amount of interactions, as expected. In figure 3.66 the data throughput out of the ZDC FLP

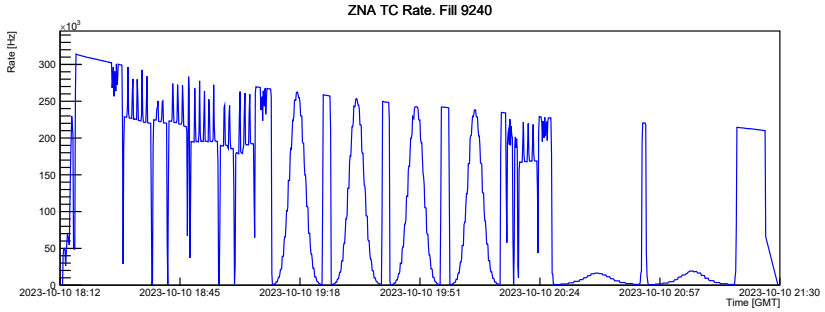


Fig. 3.63 ZNA TC interaction rate during the ALICE VdM scan of fill 9240.

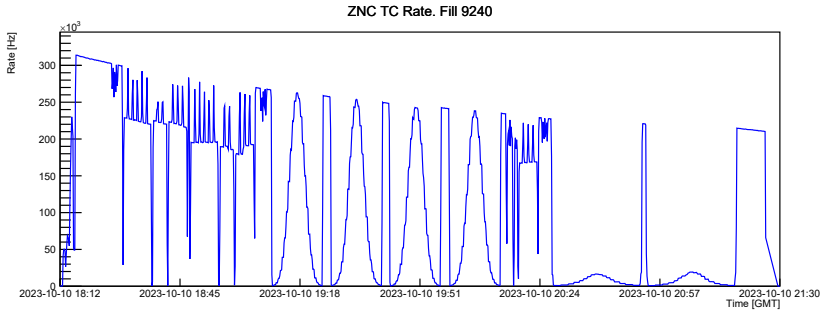


Fig. 3.64 ZNC TC interaction rate during the ALICE VdM scan of fill 9240.

during fill 9240 is displayed. Given the data format requirements of the continuous readout, even at an interaction rate of 0 Hz the data throughput is $\approx$ 50 MBps. This is due to the fact that every orbit the first and last BC are transmitted along with the RDH headers and footers. From the plot it can be seen that during the VdM fill the data throughput varied between 50 MBps and 390 MBps at 300 kHz interaction rate. This is in accordance with the calculations performed in section 3.1.4.

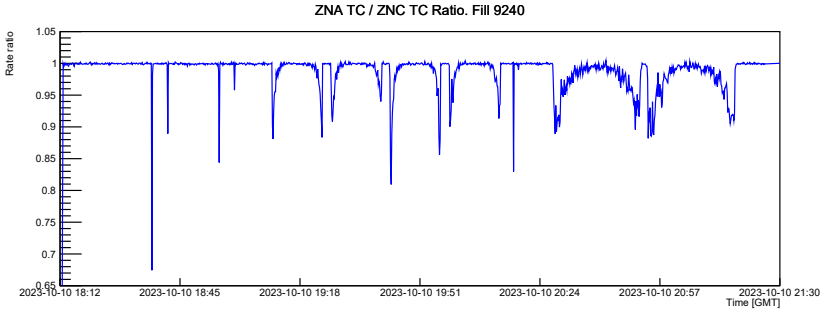


Fig. 3.65 ZNA TC / ZNC TC interaction rate ratio during the ALICE VdM scan of fill 9240.

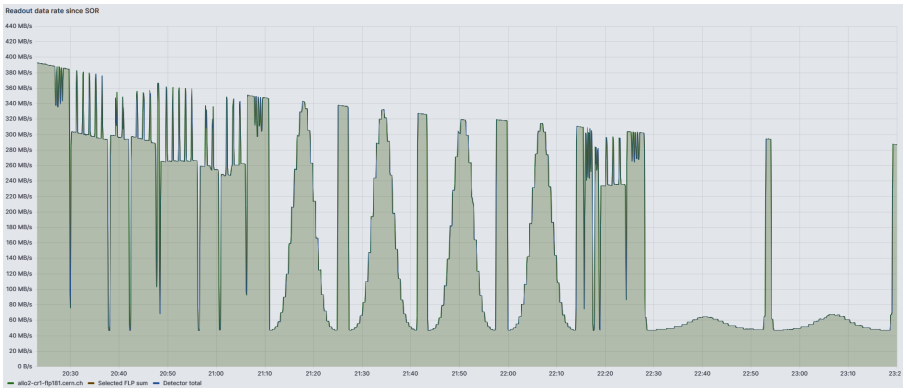


Fig. 3.66 ZDC data throughput during the ALICE VdM scan of fill 9240.

3.6.14 The photomultiplier aging effect

During Run 3, especially during the Pb–Pb data taking at full intensity, the most exposed calorimeters (common neutron PMTs) reached an event rate up to 1200 kHz. This caused a current in the photomultipliers higher than during Run 1 and Run 2. In figure 3.67 it is shown a trending plot of the 1 neutron and 1 proton peak position for both sides. On the X axis the run numbers of the Run 3 Pb–Pb data taking of 2023 are depicted. On the Y axis the position of the peaks in ADC samples are shown. It can be noted that slope of the neutron calorimeters is higher with respect to the proton calorimeters; this is due to the higher rate sustained by the former. In order to maintain the amplitude of the 1 neutron peak at the expected value, the voltages of the neutron photomultipliers were increased in order to increase the gain and therefore compensate the ageing. This can be seen in the plot of figure 3.67 around run number 545000, where the position of the one neutron peak gets a shift upwards.

3.7 The 2024 pp Van der Meer scans

After the 2023/2024 LHC winter shutdown the ZDC detector was turned on for the first pp VdM scan of 2024. This period lasted from the afternoon of May 16^{th} 2024 up to the morning of May 19^{th} 2024. The data taking was performed over the span of 2 main fills.

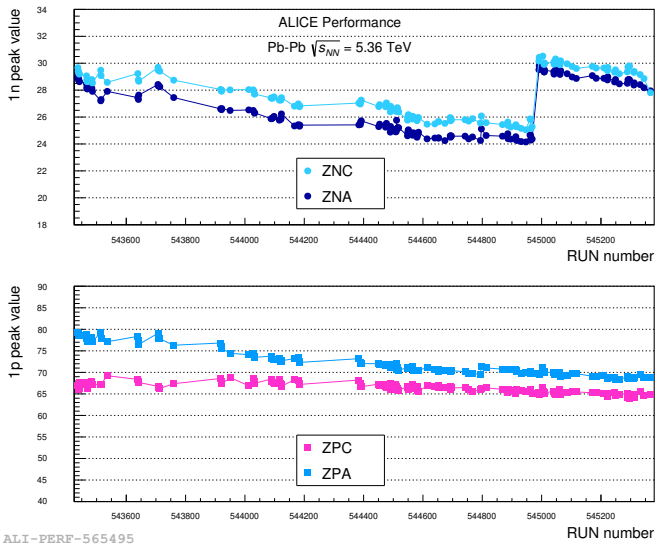


Fig. 3.67 Pb–Pb 1 neutron and 1 proton trending plot.

- fill number 9639 which lasted 34h 06m 35s; with filling scheme 525ns_138b_136_35_16_8bpi_19inj_800ns_bs525ns.
- fill number 9644 which lasted 12h 25m 59s; with filling scheme Multi_72b_60_20_22_4bpi_20inj_1000ns_bs1000ns.

In this section the plots from run number 551729 will be displayed. In figures 3.68 and 3.69 the signal from the two common PMTs of the neutron calorimeters during run number 551729 are shown. The shape and amplitude of the signal are in line with the previous pp data taking periods.

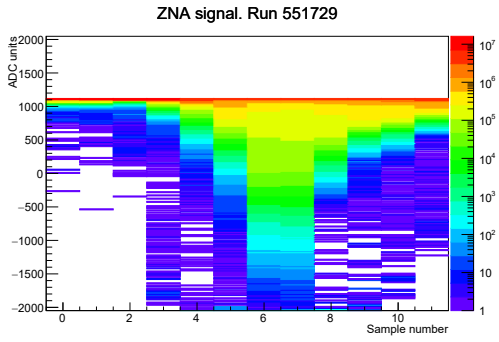


Fig. 3.68 Run 551729 waveform ZNA TC.

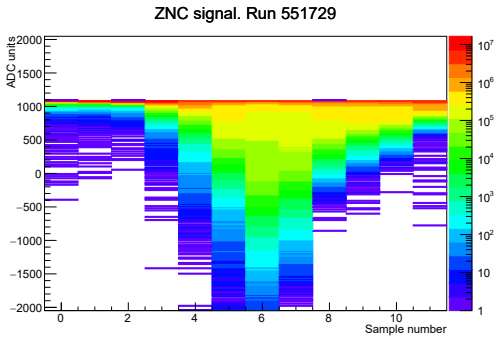


Fig. 3.69 Run 551729 waveform ZNC TC.

3.7.1 Alignment plot and interaction rate

In figure 3.70 the alignment plot of run number 551729 is displayed. Over the period of 8 hours the alignment was stable and centered around sample 6. In figure 3.71 the rate summary plot of run 551729 is shown. The most expose channels (ZNA TC, ZNC TC, ZPA TC, ZPC TC, ZEM 1 and ZEM 2) sustained an event rate of $\approx$ 8 kHz. This is in line with the other pp run with equivalent fillings scheme and configuration.

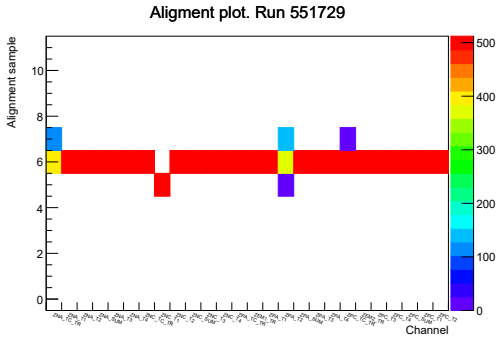


Fig. 3.70 Run 551729 alignment plot.

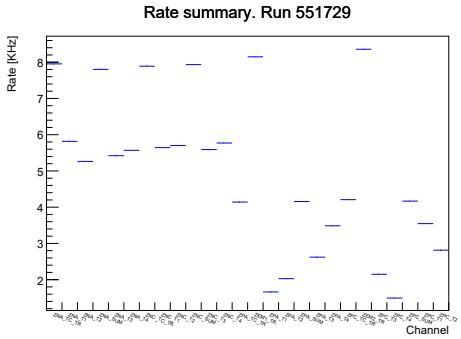


Fig. 3.71 Run 551729 Rate Summary.

3.8 Considerations and results

Over the span of three years, from 2022 to 2024, the ZDC detector took data in several periods, both in pp and Pb–Pb conditions. After every period problems and opportunities were identified, therefore improving the readout firmware and software. During the 2023 Run 3 Pb–Pb data taking the system acquired more data than Run 1 and Run 2 combined. Overall the system always performed properly and was able to take data with the other ALICE detector in global configuration.

Chapter 4

The MIZAR ASIC and readout firmware

This chapter will briefly introduce the experimental context of the newly designed MIZAR ASIC [100], of the TERZINA experiment, and PBR mission. In section 4.2 a detailed description of the Multi-channel Integrated Zone-sampling Analogue-memory based Read- out (MIZAR) Application-Specific Integrated Circuit (ASIC) will be provided and in section 4.3 an overview of the TERZINA experiment will be presented. In section 4.4 an exhaustive presentation of the TERZINA readout firmware architecture and design is reported.

4.1 Experimental context for the MIZAR ASIC

Ultra-High Energy Cosmic Rays (UHECRs) and tau leptons produced in the crust by Ultra-High Energy tau Neutrinos (UHENUs) create Extensive Air Showers (EASs) as they travel through the atmosphere [101]. These showers consist of relativistic particles that emit Cherenkov light, aligned with the EAS direction. This light can be used to trace back the direction and energy of the primary UHECRs or UHENUs. To study EAS parameters, a custom designed optical system integrated into a telescope with a focal plane of Silicon Photo-Multipliers (SiPMs) can be used. When placed at sub-orbital heights (approximately 30-40 km) or in Low Earth Orbit (around 500 km), the signal typically lasts for tens of nanoseconds. Thus, a time resolution of at least 100 MHz is required to properly process these types of signals. In addition,

full waveform sampling helps differentiate EAS-related events from others, such as interactions of low energy cosmic rays directly in the detector emulating a fast light signal. To meet these requirements, an ASIC known as Multi-channel Integrated Zone-sampling Analogue-memory based Read-out (MIZAR) is being developed using commercial 65 nm CMOS technology. This design is inspired by ongoing and upcoming missions in UHECR/UHENU observations, such as the Extreme Universe Space Observatory-Super Pressure Balloon2 (EUSO-SPB2) [102–104], which flew on May 13, 2023, on a stratospheric balloon platform, and the planned PBR (POEMMA Balloon with Radio) mission, along with the TERZINA [105, 101] and POEMMA [106, 107] space-based missions. MIZAR was created to manage a complete acquisition chain, from signal processing of the SiPMs to digital conversion and readout.

4.1.1 Brief overview of the NUSES space mission and the TERZINA experiment

The NeUtrino and Seismic Electromagnetic Signals (NUSES) mission aims to study cosmic radiation and the Sun-Earth environment using two advanced detectors. The first detector, TERZINA, focuses on UHECRs with energies exceeding 100 PeV [108, 109], while the second, ZIRE' [110], is dedicated to studying cosmic rays below 250 MeV. The mission is planned to last three years, with the satellite orbiting initially at 550 km of altitude and dropping eventually to around 525 km at the end of the experiment expected life. TERZINA is designed to detect Cherenkov light from EAS triggered by UHECRs in the Earth's atmosphere at energies above several hundred PeV. In addition, TERZINA will observe below the Earth's limb to monitor atmospheric and ground emissions, which will help in characterizing light intensity and validating the detection of Earth-skimming UHE neutrinos. The TERZINA experimental payload includes:

- the Optical Head Unit: a near-UV optical telescope.
- the Focal Plane Assembly (FPA): which is designed to detect photons from both above and below the limb.
- the Thermal Control System (TCS): to maintains the telescope's temperature.

- the external Harness and Electronic Units: in order to provide power to the components and communication with the earth.

The telescope features a dual mirror optical system to maximize the focal length within its space constraints. The system is shaped like a cut-cone with a diameter of 394 mm and a length of 350 mm, resulting in a focal length of approximately 925 mm with its optical axis pointing towards the Earth's limb. The FPA has a rectangular shape with a 2:5 aspect ratio and includes 10 arrays of Silicon PhotoMultipliers (SiPMs), each array having 8×8 pixels arranged in 2 rows of 5 arrays. The SiPMs use Near-Ultraviolet High-Density Low Crosstalk (NUV-HD-LowCT) technology from FBK [111]. The telescope's field of view is 7.2° horizontally and 2.5° vertically, with each pixel covering 0.18°. This setup allows it to observe a large atmospheric volume with a cross-sectional area at ground of 140×360 km^2.

4.1.2 Brief overview of the POEMMA-Balloon-Radio mission

The POEMMA-Balloon-Radio mission is currently in the conceptual study phase, drawing significant inspiration from the EUSO-SPB2 mission, which flew on a NASA Super Pressure Balloon launched on May 13, 2023, from Wanaka, New Zealand. The EUSO-SPB2 payload included a Fluorescence Telescope equipped with a Multi-Anode PhotoMultiplier Tube (MAPMT) camera aimed in order to capture fluorescence light from UHECR EAS with energies above 1 EeV. In addition it featured a Cherenkov telescope with a SiPM focal surface designed to observe Cherenkov emissions from EAS with energies above 1 PeV. The Cherenkov telescope device is a 1-meter diameter modified Schmidt telescope with a bifocal alignment of its four mirror segments, which focus the light into two distinct spots on the camera rather than one. The bifocal setup allows to differentiate between direct cosmic ray hits (resulting in one spot) and light from outside the telescope (resulting in two spots), thus reducing the background noise. The core of the Cherenkov telescope consists of a 512-pixel SiPM camera (S14521-6050AN-04 from Hamamatsu) with a 10 ns integration time and a readout depth of 512 samples centered around the trigger. This was designed in order to record very fast and bright signals, such as the Cherenkov emission from EASs, to meet the mission's scientific targets. The instrument has a field of view of 6.4° in zenith and 12.8° in azimuth and can be

adjusted during flight from horizontal to 10° below the Earth's limb, depending on the scientific requirements.

4.2 The MIZAR ASIC architecture

The MIZAR ASIC is a 64-channel chip designed in 65-nm CMOS technology for the readout of SiPMs [112]. For each channel the the analog signal produced by the SiPM gets processed by a common gate front-end amplifier and then stored in a 256 cells analog memory which works with 200 MHz sampling frequency. Each memory cell is made of a storage capacitor followed by a single-slope Wilkinson ADC with configurable resolution between 8 bits and 12 bits. This design allows the conversion of each channel in parallel, thus greatly reducing the overall digitization time. In addition, the memory can be configured to work in smaller sections (32 or 64 cells) to reduce the total data throughput and time of conversion. Given the tight power budget (5 mW/channel) and to minimize the dead time between conversions, the digitization process is enabled by an external acknowledge signal which is generated by the readout FPGA logic after analyzing a hitmap. The hitmap mechanism will be explained in detail in section 4.2.2. In figure 4.1 a simplified block diagram of the

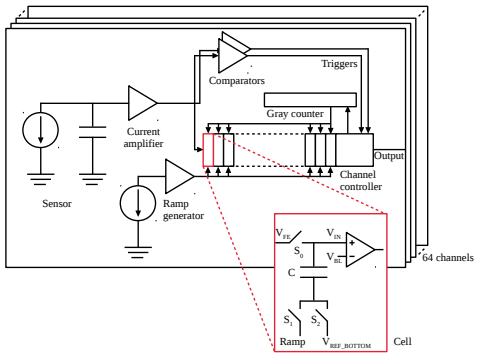


Fig. 4.1 TERZINA channel architecture diagram overview. Figure taken from [100]

MIZAR channel is shown, while in figure 4.2 the layout of a single memory cell is depicted.

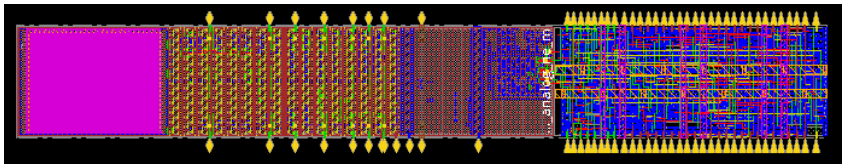


Fig. 4.2 Layout of a cell unit. On the leftmost side the pink area highlights the memory capacitor. The rightmost part represents the digital partition of the cell. Figure taken from [100]

4.2.1 Input/Output

The MIZAR ASIC has 64 analog inputs, one for each SiPM channel and 12 LVDS digital I/O divided into 4 outputs and 8 inputs. The digital logic of the chip works at 200 MHz; this frequency is obtained dividing by 2 the 400 MHz clock supplied by the readout FPGA. This approach was chosen in order to move the high frequency clock generation to the readout FPGA instead of the chip, thus reducing power consumption, cost, area and complexity of the ASIC. However, such a design sets a constraint on the maximum distance between chip and readout due to the transceivers limit and signal degradation over copper cables. From tests carried out in laboratory with two FPGA this limit seems to be around 1 m, which is enough for the expected use case of the MIZAR ASIC in the TERZINA and PBR3 experiments. This will be shown in the next chapter.

Name	Direction	Comment
CLK_400	in	400 MHz input clock for the digital domain
RESET	in	Reset for the digital domain
CMD_FPGA[1:0]	in	2 bit signal from FGPA; acknowledge signal
TP	in	Test Pulse input; used for debug
SPI_CLK	in	SPI clock; used for configuration
SPI_MOSI	in	SPI Master Out; used for configuration
SPI_EN	in	SPI enable; used for configuration
SPI_MISO	out	SPI MAster In; used for configuration
READ_RQ	out	Read Request signal from the ASIC
HITMAP	out	400 MHz DDR output for the hitmaps
DATA	out	400 MHz DDR output for the digitized data

Table 4.1 MIZAR digital I/O summary.

4.2.2 Hitmap description and generation

Figure 4.1 shows that each channel hosts a set of two configurable comparators. When a photon hits the sensor a current signal gets induced and processed by the front-end amplifier. The output of this stage gets then distributed to:

- a high threshold comparator;
- a low threshold comparator;
- the 256 memory cells.

The double threshold system is used as a level 0 trigger to activate the readout machinery only when a configurable minimum signal is detected. When the interesting condition is reached the READ_RQ output gets asserted high, to let know the FPGA that the ASIC detected a potential event. At this point the ASIC enables a configurable counter which is supposed to last $\approx$ 80 ns. If the FPGA does not reply to the hitmap request within this time window, or if it replies negatively, the event gets discarded. Otherwise a hitmap, a 64 bit string where each bit represents the comparator state of a channel, is generated and sent to the FPGA. The hitmap packet is made of 8 words of 14 bit, thus 112 bits in total. Each data stream is composed of:

- 14 bits of zero
- 8 bits of header for data alignment purposes
- 6 bits of packet info
- 16 bit of timestamp
- 12 spare bits
- 64 hitmap bits
- 6 bits of footer

In figure 4.3 a diagram with the hitmap packet structure is shown. To be noted that before each packet the ASIC always sends a word of zeros. The chip can be configured to operate in three modes:

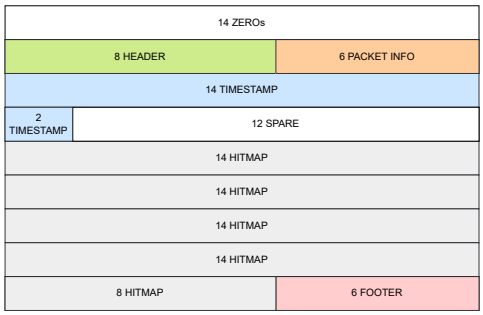


Fig. 4.3 MIZAR hitmap packet structure.

- TIME WINDOW:
  the ASIC produces a READ_RQ with a fix configurable period no matter what is the state of the two comparators.
- COINCIDENCE DRIVEN + EDGE DRIVEN + HIGH THRESHOLD:
  the ASIC produces a READ_RQ if:
  - at least two channels exceeded the low threshold. When a pixel surpasses the low threshold a configurable counter starts counting for ≈ 20 ns in order to check if the signal was generated by a physics event or some electronic noise. If within this time limit the low threshold is surpassed by only one pixel, then the event gets discarded.
  - at least one edge channel exceeded the low threshold. Each channel can be configured as edge. In a 8×8 matrix, for the TERZINA and PBR case, the edge pixels are the 8 in the first column and the 8 in the last column. This configuration was chosen in order to eliminate the blind trigger area between to adjacent SiPMs;
  - at least one channel achieved the high threshold;
- FPGA REQUEST:
  the ASIC produces a READ_RQ when an instruction from the FPGA is received.

Since each channel is equipped with two comparators, namely high and low threshold, the MIZAR ASIC can generate two different hitmaps. However, only a hitmap is sent each time an interesting event gets detected. The high threshold has always highest priority. This means that:

- If the low and high thresholds conditions are not met the hitmap is not generated.
- If only the low threshold condition is met the low threshold hitmap is generated.
- If the low and high thresholds conditions are met the high threshold hitmap is generated.
- (The case in which only the high threshold condition is met is not considered since the energy required for the high threshold will always be greater than the one required for the low threshold.)

After the READ_RQ assertion and the positive reply from the FPGA, the hitmap is generated. The resulting packet gets transmitted at 400 MHz in DDR. This means that, at 1.25 ns/bit, the whole transfer lasts 157.5 ns.

4.2.3 Data packet

As mentioned in the previous sections, the digitized data output can be configured in both, resolution (from 8 to 12 bits) and number of points (32, 64 or 256 samples). The sampling rate is fixed at 200 MHz. In figure 4.4 a data packet example in the configuration with 8 bit resolution and 32 samples is depicted. A total of 15 configurations are possible. Table 4.2 summarizes the data packet size for each configuration. The raw data generated by the the ASIC can vary between 20.6 kb and 200.7 kb; almost by a factor 10. Each packet is composed of:

- 14 bits of zero;
- 8 bits of header for data alignment purposes;
- 6 bits of packet info;
- 16 bit of timestamp;
- 9 address bits;
- 1 write status bit;
- 2 spare bits;

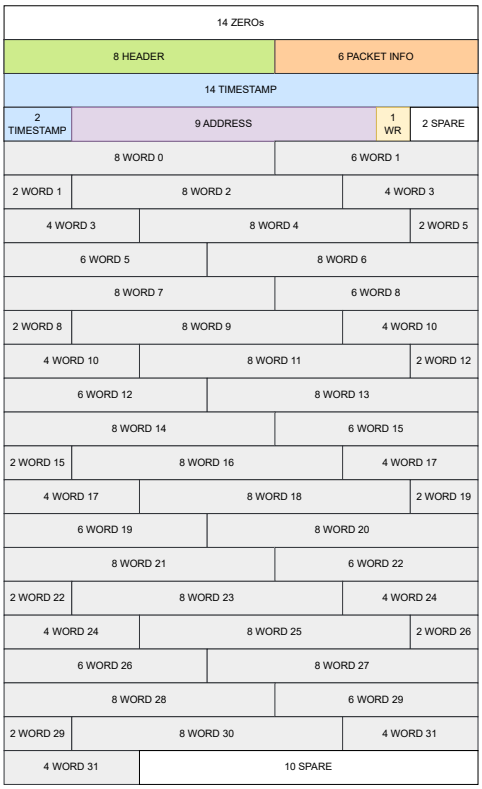


Fig. 4.4 MIZAR data packet structure; example with 8 bit resolution and 32 samples.

- ALL THE SAMPLES;
- spare bits until word completion.

4.3 The TERZINA camera and readout architecture

In the context of the TERZINA experiment the optical system was configured in a dual mirror setup to reduce aberrations on the photon detection plane (FPA). The FPA has a 5:2 rectangular aspect ratio and it is composed of 10 SiPMs, with 8x8 pixel geometry. The resulting telescope has a field of view of 7.2°horizontally and 2.5°vertically; this result in an observable atmosphere volume of 140x360 km^2 [101]. The sensors are provided by the FBK in Trento. Each SiPM is readout by one MIZAR ASIC. The camera and the electronic readout are arranged in a layered setup described in figure 4.5. The top layer hosts the 10 SiPMs which receive the photons from the telescope, while the middle layer accommodates the ASICs. Those two

Samples	Resolution	N bits	N words	ASIC kb
32	8	312	23	20.6
64	8	568	41	36.7
256	8	2104	151	135.3
32	9	344	25	22.4
64	9	632	46	41.2
256	9	2360	169	151.4
32	10	376	27	24.2
64	10	696	50	44.8
256	10	2616	187	167.6
32	11	408	30	26.9
64	11	760	55	49.3
256	11	2872	206	184.6
32	12	440	32	28.7
64	12	824	59	52.9
256	12	3128	224	200.7

Table 4.2 MIZAR data packet size for each configuration. In column "N bits" the number of bits, without padding, for the selected configuration for each channel. In column "N words" the total number of words, thus with padding, for each configuration per channel. In column "ASIC kb" the total packet size in kilobits for the hole ASIC plane (64 channels).

planes combined are designed to consume less than 5 W. The bottom plane integrates two FPGAs, each one controlling 5 ASICs. In this way, the worst-case scenario of an FPGA failure can be managed by tilting the telescope to exploit a low field of view configuration with just one FPGA and 5 ASICs. In figure 4.6 a diagram with a schematic diagram of the connections leading to the two readout FPGAs is shown. The data collected by the readout FPGAs is cleaned of all the non interesting events, then a bit-stream is sent to a data concentrator via a quad SPI protocol. The concentrator is managed by the satellite logic which stores all the events in RAM and uploads them to station on earth when it has a data connection. The maximum allowed data throughput, given the bandwidth constraints, is 45 Gb/day. Thus, the maximum theoretical trigger rate can be calculated as:

$$\frac{45\,Gb}{seconds\,in\,a\,day\ *\ event\ size} = \frac{45000000\,kb}{86400\,s\ *\ 20.6\,kb\ *\ 10} = 2.53\,Hz \qquad (4.1)$$

In equation 4.1 an example is given considering:

- 8 bit resolution;

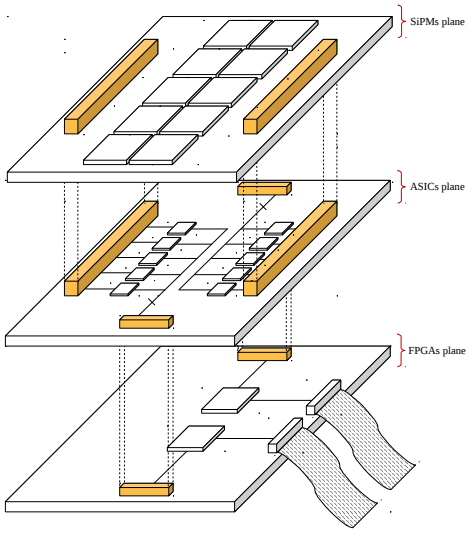


Fig. 4.5 TERZINA camera architecture diagram. Figure taken from [112]

- 32 samples;
- 64 channels per ASIC;
- 10 ASICs per camera.

The available average trigger rate could be greatly increased by reducing the number of channels sent. Instead of transmitting all the camera data at each trigger the readout can be configured for sending only the relevant chip or a portion of the pixels of the interesting ASIC. This approach avoids the transmission of data mainly related to the baseline of the pixels. Since the data transmission bottleneck concerns only the link from the satellite's DPCU (Data Processor Central Unit) to the earth, the firmware can always be configured for transmitting all the ASICs and the zero suppressing can be performed via software with the DPCU CPU. One of the proposals is to only send a 3x3 pixel matrix around the triggering area.

4.3.1 The expected trigger implementation

In figure 4.7 a diagram with the TERZINA camera plane with the pixels configured as EDGE highlighted in red is shown. In order to validate the event, and thus to start the digitization process, the FPGA looks for clusters in the hitmaps. In the case of the low threshold hitmap the accepted patterns are groups of two, three or four

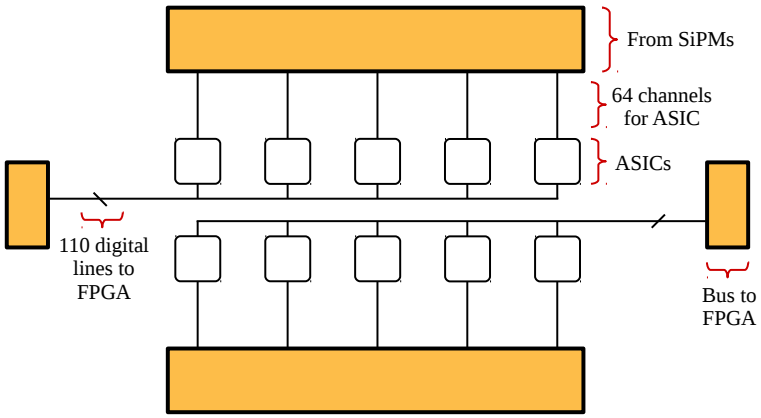


Fig. 4.6 Diagram showing the connections between SiPMs, ASICs and FPGAs.

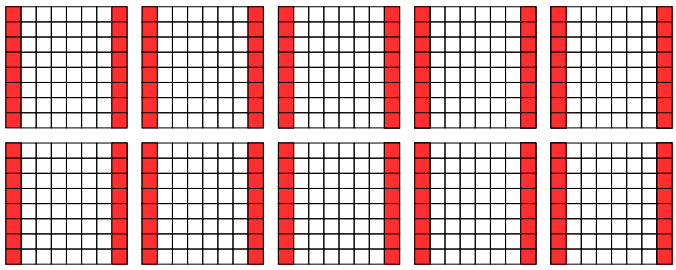


Fig. 4.7 TERZINA camera plane with pixels configured as EDGE highlighted in red. Figure taken from [101]

adjacent pixels. While, in the case of the high threshold hitmap, the accepted patterns are a single pixel or groups of two. In figure 4.8 a diagram with all the accepted patterns is depicted. Figure 4.9 shows an example of the TERZINA camera plane with some signals. In black there are the accepted patterns, while in grey there is only noise. To illustrate the triggering process Figure 4.10 shows a diagram with the trigger path. The main steps are:

- a photon hits the camera plane and generates an electrical signal. The comparators in the MIZAR ASIC detect the event. A read request is then sent to the FPGA;
- if the configuration allows it, the FPGA will reply positively to the ASIC which will generate the hitmap and send it to the FPGA;

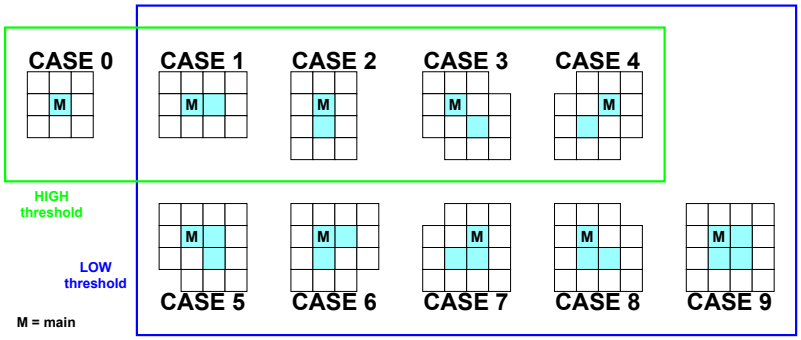


Fig. 4.8 Diagram with all the accepted trigger patterns for the high threshold (green) and the low threshold (blue).

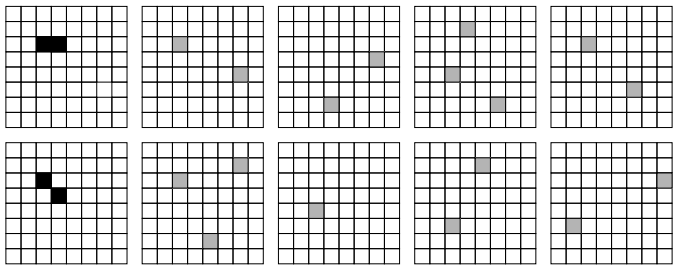


Fig. 4.9 Low threshold TERZINA camera plane example with two patterns and electronic noise. Figure taken from [101]

- a custom designed algorithm checks for specific patterns in the hitmap. If the fingerprints of the interesting events are found, a signal is sent to the ASIC which will start the digitization process and will send the data to the FPGA.

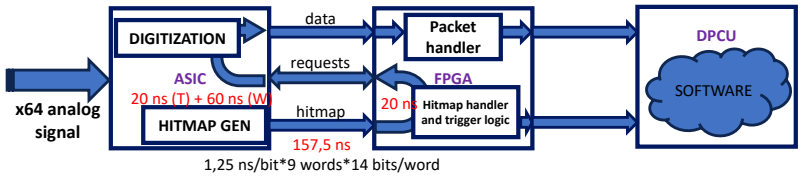


Fig. 4.10 Diagram showing the information exchanges during the triggering procedure.

At the end of the process the FPGA will send both, the hitmap packet and data packet to the DPCU which will store the events and manage the transmission to earth.

4.4 The TERZINA readout firmware

My contribution to the NUSES collaboration consists in the full design, implementation and test of the FPGA firmware used for testing the MIZAR ASIC and for readout during data taking in orbit. The TERZINA readout firmware is being developed using XILINX VIVADO 2021 in non project mode. Currently the MIZAR layout is being finalized and a fist MPW (Multy Project Wafer) will be submitted in late 2024. The firmware described in this section will be used to test the first chips in laboratory.

4.4.1 Firmware overview

The firmware was designed from scratch having as final goals:

- a reliable, fast, and consistent way to calibrate in time the data transmission between ASICs and FPGAs;
- a quick (less than 30 ns) handling time for the pattern seeking algorithm;
- an SPI system for the ASIC configuration slow control;
- a stable and reliable communication protocol to send the data to a PC (for testing purposes).

At the time being the firmware is still under development. In this section the working modules will be presented; the remaining parts to be completed will be discussed in section 4.5. In figure 4.11 it is presented a diagram with the main firmware blocks. From figure 4.11 it can be noted that:

- a PLL generates four synchronous clocks which are distributed to the different firmware modules:
  - 400 MHz $\rightarrow$ it is directly sent to the ASIC which will use it for the DDR communication and, after dividing it by two, to the internal logic and sampling.
  - 200 MHz $\rightarrow$ it is used for the firmware internal logic.

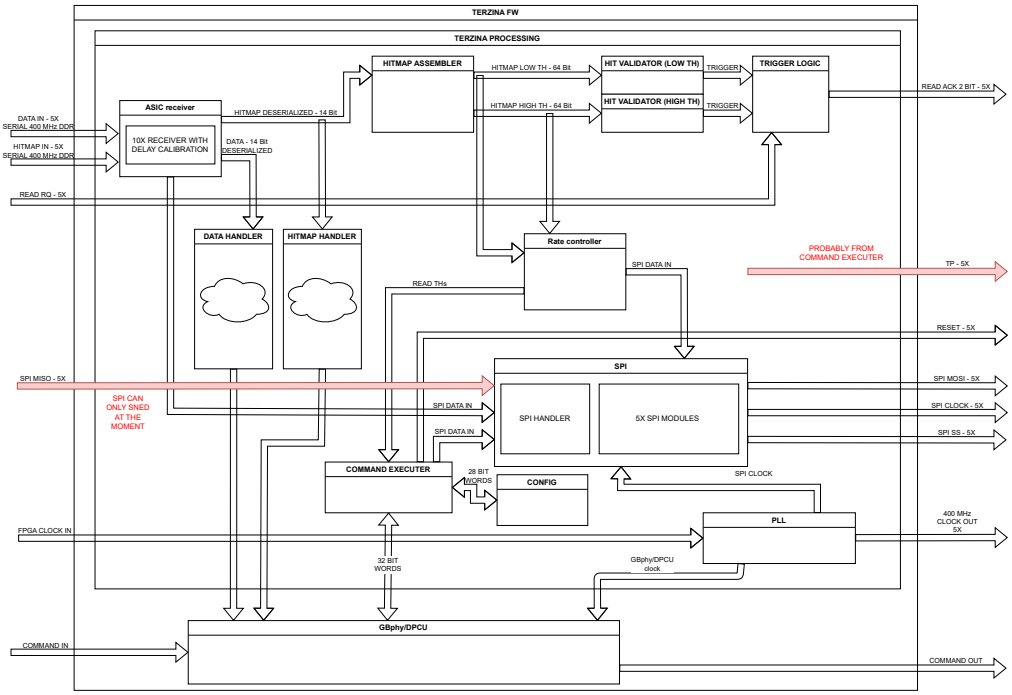


Fig. 4.11 TERZINA readout block diagram. In red the portions still to be developed.

  - 125 MHz → it is used for the bidirectional custom communication protocol with the PC.
  - 57.15 MHz → it is used for the de-serialization module.

- from the left side of the diagram the 10 ASIC differential serial communication lines arrive in a *ASIC receiver* module containing 10 *receiver* modules, 5 for the hitmaps and 5 for the pixel data. A custom procedure is performed regularly in order to compensate in time for the length of the cables and the differences in temperature. The output of the module are 10 registers of 14 bits each. The data transmission is performed at 400 MHz in DDR, thus 800 Mbps. The stream gets deserialized into 14 bits word at 57.15 MHz. The data is then put into a CDC (Clock Domain Crossing) FIFO with the output clocked at 200 MHz as the rest of the firmware logic.

- the 5 hitmap routes go to two different modules: to a *hitmap assembler* which is designed to have the least possible delay in order to feed the two *hit validator* modules for the pattern detection and to a *hitmap handler* which is designed to funnel evenly the data from the five ASICs into one stream of hitmaps ready for transmission to a PC.

- the hitmap assembler modules outputs 10 registers of 64 bits each. Two registers for each ASICs, one for the high threshold and one for the low threshold. The information gets processed by the hit validator modules.
- the trigger information is sent to a *trigger logic* module which also receives the read request signals from the chips. Using the information from the trigger signals, the read request signals and the readout configuration, this module dispatches the read acknowledge 2 bit register to each ASIC.
- a configuration module, which is connected to every other entity in the firmware, stores all the configuration parameters and allows for read-back.
- the assembled hitmaps are also sent to a module named *rate controller*, here a configurable logic counts the number of events over threshold in a time window for each pixel and, if certain conditions are met, can dispatch a command to increase/decrease the thresholds.
- a custom UDP module developed by the staff of INFN Turin is used for communication with the PC, transmitting and receiving 32 bits words. These words get decoded by the *command executor* module which can write/read the configuration registers, send reset signals, and send specific SPI commands to the ASICs.
- the *SPI* module handles the requests for the SPI lines using a priority encoder. The entity receives inputs from three sources:
  - the ASIC receiver module $\rightarrow$ SPI is used to inform the ASIC that the calibration procedure is ongoing and it has to start sending a deskewing or alignment pattern.
  - the rate controller module $\rightarrow$ if a threshold adjustment is needed the rate controller module generates the SPI command and sends it to the handler.
  - the command executor module $\rightarrow$ specific SPI commands can be sent for debugging or configuration purposes.

  The received commands are sorted by ASIC and stored in 5 different FIFOs, one for each chip. If certain conditions are met the commands are dispatched.
- the *data handler* module receives the de-serialized words from the ASIC receiver and, for each chips, it detects the start and the end of each data packet,

composes the message with a header, a footer and the information from the ASIC number, and then funnels the data from all the ASIC to a single output towards the PC.

4.4.2 ASIC receiver and receiver module

The ASIC receiver module is a wrapper containing the 10 receivers (5 for the data and 5 for the hitmaps), a handler for the SPI requests generated by the timing calibration procedure, and the CDC fifo for the SPI commands which are generated in the 57.15 MHz de-serialization time domain and need to be executed in the 200 MHz logic time domain. In figure 4.12 the receiver module input/output diagram is depicted; a total of nine inputs and four outputs can be seen. Below a brief description for each of them:

Fig. 4.12 Receiver module I/O diagram.

- CLK_400 is the input clock at 400 MHz used for de-serialization.
- CLK_57 is the input clock at 57.15 MHz used to handle the de-serialized words. To be noted that 57.15 = 800/14. The two clocks are generated from the same PLL and they are synchronous.
- reset is a synchronous reset input.
- receiver_enable is a one bit parameter which enables or disables the module. When the register is HIGH the receiver is enables, and when the register is LOW the input clocks get clock gated, thus disabling the de-serialization and reducing the power consumption.
- data_in_p and data_in_n are two the serial differential lines at 400 MHz in DDR coming from the ASIC.

- DoAlignSerialData is a signal which is used to start the timing calibration procedure. If the input is HIGH for at least one clock cycle the calibration machinery begins.
- AlignmentWord1 is a 14 bit register coming from the configuration module and is used during the de-skewing part of the timing calibration process. During normal operation is 0b01010101010101.
- AlignmentWord2 is a 14 bit register coming from the configuration module and is used during the shift alignment part of the timing calibration. During normal operation is 0b00000001111111.
- alignment_request_word1 is a 1 bit signal which gets converted into a SPI command in the ASIC receiver module. The command is used to make the MIZAR ASIC start sending the first alignment word.
- alignment_request_word2 is a 1 bit signal which gets converted into a SPI command in the ASIC receiver module as the previous one. The command is used to switch the chip output to the second alignment word.
- alignment_request_data is, as well, a 1 bit signal which gets converted into a SPI command in the ASIC receiver module, and is used to signal the end of the timing calibration process, thus the ASIC can restore normal operation.
- data_out is the 14 bit data output register clocked at 57.15 MHz.

Given the expected operation temperature variation due to the shade and solar exposure in space of the TERZINA experiment being higher than 40°C, a transmission line with fixed delays is not possible. This is especially true considering the high frequency and long run of the cabling. The Voltage Controlled Oscillators (VCOs) used for the clock generation of the modern FPGAs are not designed to maintain continuously the necessary stability needed for the experiment during the heating and cooling periods. For this reason a custom timing calibration procedure was designed in order to automatically test and select independently for each serial line the best possible delay parameter. In figure 4.13 a diagram of the calibration FSM is shown, meanwhile in figure 4.14 a screenshot obtained from an actual calibration test performed with hardware can be seen. The timing calibration procedure works as follows. During normal operation the FSM stays in IDLE mode, this means that no

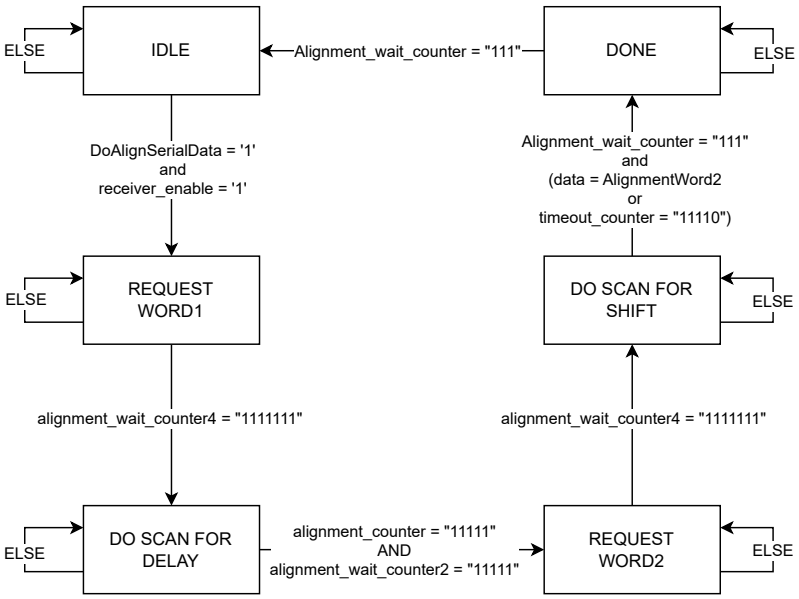


Fig. 4.13 TERZINA timing calibration procedure FSM.

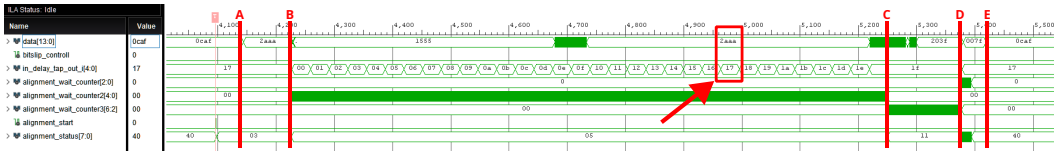


Fig. 4.14 Calibration example performed with hardware and recorded by an Internal Logic Analyzer (ILA). Given that the MIZAR ASIC is not ready yet, an ASIC emulator firmware was designed and uploaded into a second FPGA board; more on this will be presented in section 5.3.

modification is performed on the delay parameters and the logic is waiting a signal to start the procedure. The alignment_status register on the last row of figure 4.14 displays the different states of the calibration FSM:

- 0x40 → IDLE;
- 0x03 → READ REQUEST WORD 1;
- 0x05 → DO SCAN FOR DELAY;
- 0x11 → READ REQUEST WORD 2;
- 0x21 → DO SCAN FOR SHIFT;
- 0x08 → DONE.

If the receiver is enabled and the alignment command input goes to HIGH, then the procedure begins, in figure 4.14 this is depicted with point T (the trigger event of the ILA). The first action performed is to send a request to the ASIC to transmit the de-skewing pattern, at the same time a counter is running in order to wait for the message to be received by the chip. This can be noted at point A, in the first half of the READ REQUEST WORD 1 state, which highlights the moment where the data of the ASIC switches between an idle data message (in this case 0x0CAF) and the de-skewing pattern 0x(2AAA). The actual de-serialization process is performed by the LogiCORE IP SelectIO Interface Wizard provided by XILINX [113, 114] which has a 5 bit IODELAYE2 [115] tap in signal. In the DO SCAN FOR DELAY state two counters are used. The first one keeps track of the delay tap value (in_delay_tap_out_i in figure 4.14), meanwhile the second one (alignment_wait_counter_2 in figure 4.14) counts the clock cycles elapsed for each delay value. For each possible delay value the logic acquires 22 words and compares them with the known good de-skewing word and its inverse. At this point the logic performs the scan of all the delay values (from point B to point C in the example) and calculates the longest section with no errors (in this case from 0x10 to 0x1e). A XILINX divisor IP [116] is used to compute the middle position using equation 4.2.

$$Average\ position\ =\ \frac{\sum_{i=a}^{b} D_i}{b-a} \tag{4.2}$$

With a and b being respectively the starting and finishing positions of the section with no decoding errors and D_i being the value of the delay at the i^{th} position. In the case of the example the computed average position is 0x17. After the scan finishes the logic transitions to the next state, READ REQUEST WORD 2 (from point C to point D), in which a command is dispatched to the ASIC requesting the second calibration word and waiting for the command to be executed. In the DO SCAN FOR SHIFT state (point D) the logic asserts HIGH the bitslip control of the deserializer IP until the output matches with the expect word. In the provided example this happens after just one bit shift. The logic then goes into the DONE state sending a command to the ASIC to resume normal operations (point E in the example). To be noted that the calibration from figure 4.14 started from an already calibrated configuration. In figure 4.15 is presented another example which starts from a non calibrated configuration. To the left, in the red rectangle, the words prior to the timing calibration can be seen. In the center, yellow rectangle, the process is

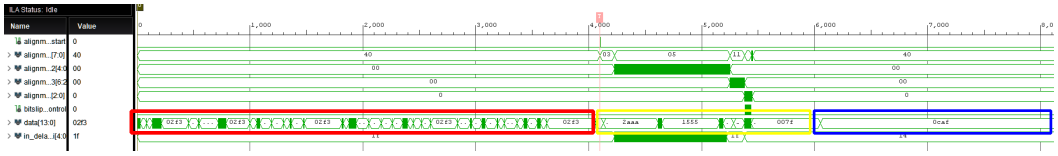


Fig. 4.15 Calibration example performed with hardware starting from a not calibrated configuration. The interesting areas, before, during, and after calibration are highlighted respectively in red, yellow and blue.

performed. To the right, blue rectangle, the words after the calibration. The complete procedure takes $\approx$ 1400 clock cycles at 57.15 MHz, thus 24.5 μs, and is expected to be performed in between every 5 to 30 minutes of activity.

4.4.3 HITMAP assembler

The HITMAP assembler module was designed in order to decode and dispatch in the shortest possible time two 64 bit registers with the hitmap information for high and low threshold for each ASIC. The system works by creating a 8 stages shift register with the input data and a check on the first two words. If the first word is all zeros and the second word start with the expected header, then a hitmap_ready flag is raised and, in the same clock cycle, the 64 bit hitmap is extracted and gets fed into a CDC FIFO in order to synchronize the stream with the rest of the logic at 200 MHz. The delay between the transmission from the ASIC of the last bit and the hitmap ready is in the order of 40 ns. The decoded hitmaps are immediately fed into the two hit validator modules (one for high and one for low threshold) in order to check for relevant patterns.

4.4.4 HIT validator

The HIT validator module contains the pattern finder logic used for triggering. The algorithm was designed specifically for the TERZINA use case trying to minimize as much as possible the processing time. This resulted in a sub 30 ns computation time. The module receives in input five hitmaps of 64 bits each, a signal highlighting the arrival of a new hitmap, and three configuration parameters (HitMapTime, enable_high_th_pattern, and enable_low_th_pattern) which will be described with more detail later in this section. The TERZINA firmware uses two of these modules,

one receiving the high threshold hitmaps and the other receiving the low threshold hitmaps. The two entities are identical except for the configuration parameters. When a new hitmap is received from the HITMAP assembler the information is stored into a register which will get erased by the logic after a configurable number of clock cycles defined by the HitMapTime parameter. This allows the algorithm to check for coincidences between two different ASICs only in a defined time window. All the valid hitmaps are constantly combined in a single register containing all the pixels plus a two pixel wide padding ring. The non valid hitmaps are combined as well in the register, but they are considered full of zeros. The resulting vector is 528 bits long and can be represented by the diagram shown in figure 4.16. Considering half

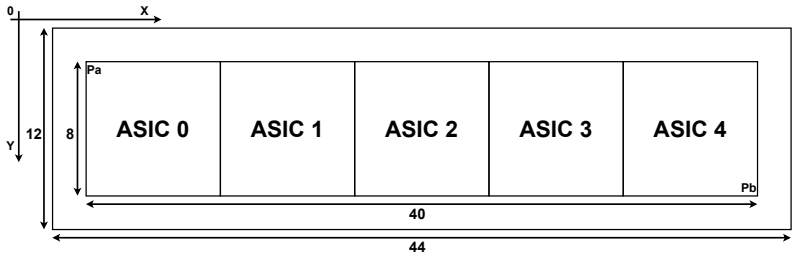


Fig. 4.16 HIT validator full hitmap with padding diagram. The X axis is depicted from left to right and the Y axis is depicted from top to bottom.

of the TERZINA plane, thus 5 sensors in a row each one with 64 pixels in a 8x8 configuration, lets imagine a 2 pixel wide frame around the sensors. At this point a 44x12 matrix is obtained of which only the center 40x8 rectangle is relevant for physics. Now lets take each row of the matrix and put it in series with each other in order to obtain a single array. In order to translate from the coordinates of the matrix to a position on the array, the formula in equation 4.3 can be used.

$$P(x,y) = x + 44 \cdot y \tag{4.3}$$

From the diagram in figure 4.16 the first pixel of ASIC 0 ($P(2,2)$), named P_a, and the last pixel of ASIC 4 ($P(41,9)$), named P_b, can be seen. These two pixels correspond then to bits number $P(2,2) = 90$ and $P(41,9) = 437$ of the register. This means that all bits before the 90^{th} and after the 437^{th} are always zero; the relevant portion of the register is from bit 437 downto bit 90. For each bit in the just mentioned section of the register a combinational check is performed. In parallel all the patterns from figure 4.8 are checked against the complete hitmap in such logic:

- the pixel marked with the capital M is considered the center of the pattern. It does not have any particular physics relevance, it just is the coordinate zero of the pattern, thus it can be said that $P(x_M, y_M) = k$.
- each other pixel of the pattern can be expressed in function of k. For example, the pixel TOP is k-44, the pixel DOWN is k+44, the pixel to the LEFT is k-1, the pixel to the RIGHT is k+1, and the pixel to the TOP LEFT is k-44-1.
- by instantiating a check for each pattern signature for each value of k between 437 and 90 with a VHDL *for loop* statement, all the possible cases can be validated in parallel. This approach is not particularly area or resource efficient; a synchronous scan over k would dramatically reduce the number of used registers, however, it would also increase the computation time which must be kept as low as possible.

An example of a valid trigger and two non valid triggers is given in figure 4.17. After

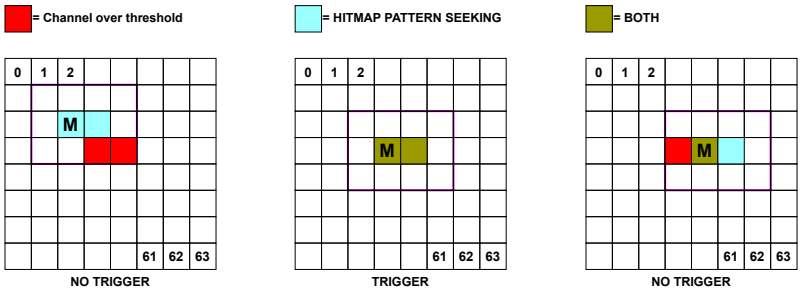


Fig. 4.17 Diagram showing an example of the algorithm looking for PATTERN 1 starting from three different positions. In the left case the Main pixel is not over threshold and the ring around the pattern which is supposed to be below threshold has two pixels over threshold. In the right case the Main is over threshold, however the pixel to its right is not and the pixel to its left it is. In the center case all the conditions are met and the trigger is fired.

receiving each new hitmap the logic performs the trigger routing. At completion the *checked* output is raised and, if a pattern was detected, the *trigger* output is also raised. This information is sent to the trigger logic module.

4.4.5 Trigger logic

The trigger logic module was designed to receive the Read Request signal from each ASIC and, depending on the configuration, to send the Read Acknowledge signal

back. This process is performed both, for the hitmap and the data, as described in section 4.3.1. There are three configuration parameters, read_request_config, ReadEnableTimeDelay and read_request_asic_*. The first one is used to select the operation mode of the module; in table 4.3 a list of the four operation modes is shown. The second one, ReadEnableTimeDelay, sets the number of clock cycles between

Parameter Value	TriggerLogic configuration (read_request_config)
0b00 (0x0)	READ NO ASIC
0b01 (0x1)	READ ONLY THE TRIGGERING ASIC
0b10 (0x2)	READ 3 ASICs, TRIGGERING + LEFT AND RIGHT
0b11 (0x3)	READ ALL 5 ASICs

Table 4.3 Trigger Logic configuration parameter.

the Read Request signal arrival and the ReadAcknowledge output. Meanwhile, the third one, read_request_asic_*, which is divided into 5 registers, one for each ASIC, is and enable parameter; when it is 0b1 the ASIC is enabled, thus the read requests will be accepted, and when it is 0b0 the ASIC is disabled, thus the read requests will be always declined or ignored. The system uses five FSMs, each one driving the Read Acknowledge signal of one ASIC, which work in parallel. A diagram of these FSMs can be seen in figure 4.18. When read_request_config is 0b00 and 0b01 the FSMs work independently and can have different states. When read_request_config is 0b10 and 0b11 the FSMs are linked to the same state. The output signal, Read Acknowledge, is a 2 bit register and can have 4 values which are shown in table 4.4.

Parameter Value	ReadAcknowledge
0b00 (0x0)	IDLE
0b01 (0x1)	READ ACCEPTED
0b10 (0x2)	READ DECLINED
0b11 (0x3)	FORCE READ

Table 4.4 Read Acknowledge possible output values.

To be noted that:

- the duration of the configurable delay can not exceed the timeout duration of the Read Request sent by the ASIC which is the time that the Read Request signal stays HIGH; this will be $\approx$ 80 ns.

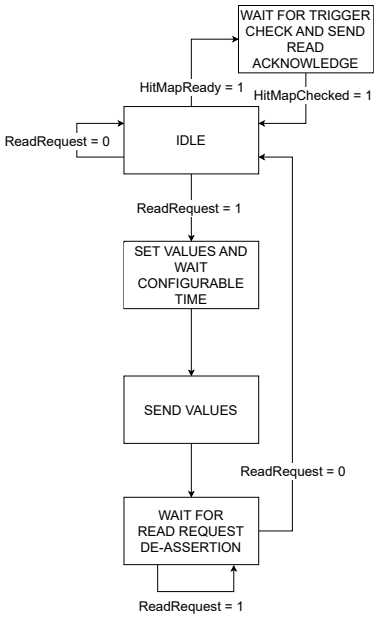


Fig. 4.18 Trigger logic finite state machine diagram. The same process manages both, hitmap and data requests. The system receives a hitmap ready signal from the hitmap assembler module; when it is HIGH the logic manages the upcoming Read Request for data, otherwise it is considered for hitmap.

- the output of the FSM will be asserted at the end of the delay period. In case of configuration 0b10 and 0b11 this will be calculated from the beginning of the first ASIC that sent a Read Request.
- when in configuration 0b10 and 0b11 the FSMs will not be triggered by disabled ASICs in order to reduce as much as possible the dead time.
- when in configuration 0b10 and 0b11 the FSMs will wait every ASIC to have Read Request = 0b0 before going back to IDLE (ready for a new trigger).

In figure 4.19 an example with Trigger logic configuration 0x1 is shown. ASIC 3 is disabled, all the others are enabled. The first five rows of the timing diagram depict the Read Request inputs, while the last five rows show the Read Acknowledge replies of the FPGA. The logic replies with a READ ACCEPTED (0x1) for every enabled ASIC before the timeout time (Read Request de-assertion) and with a READ DECLINED (0x2) to ASIC 3 which is disabled. In figure 4.20 another example is shown with the same ASIC settings, but with Trigger logic configuration 0x2 (read 3 ASICs, triggering + left + right). In this case the FSMs are all forced to have the same state. The read acknowledge output is given for all three ASICs at the end

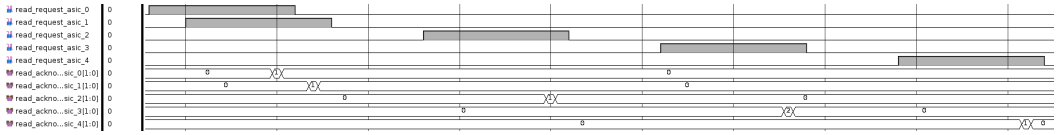


Fig. 4.19 Trigger logic simulation with configuration 0x1 and ASIC 3 disabled.

of the first counter. The disabled chip is ignored. The same principle is valid for Trigger logic configuration 0x3 (read 5 ASICs).

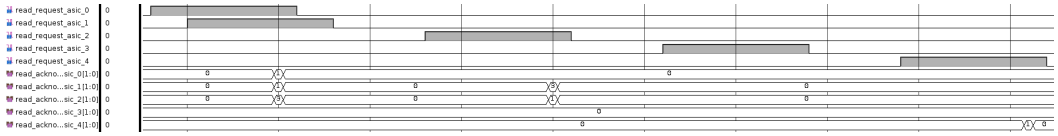


Fig. 4.20 Trigger logic simulation with configuration 0x2 and ASIC 3 disabled.

4.4.6 Configuration module

The configuration module stores and manages all the configurable parameter of the readout firmware. In tables 4.5 and 4.6 a complete list of the the parameter name, address and default value is shown. Given the 32 bit word limit of the data protocol each parameter is at maximum 16 bit long. For longer configuration settings the selected parameter is split into two or more words.
Below is shown a list with a brief description for each parameter:

- enable_data_or_and_hitmap
  bit 0 the firmware logic to send the hitmaps packets to the DPCU, while bit 1 does the same thing for the data packets. During normal operation both are enabled (0b11).
- enable_read_request_asic
  each bit of the parameter enables in the trigger logic module the readout of an ASIC. During normal operation all ASICs are enabled (0b11111).
- read_request_config
  trigger logic configuration parameter described in table 4.3.
- reset_from_config_module
  this parameter is currently disabled. During debugging it can be used to send a reset from the configuration module.

Parameter name	Address	Default value
enable_data_or_and_hitmap [01:0]	0x09	0b11
enable_read_request_asic [04:0]	0x0A	0b11111
read_request_config [01:0]	0x0B	0b01
reset_from_config_module [00:0]	0x0C	0b0
receiver_enable [04:0]	0x0D	0b11111
configurable_rate_low_1 [15:0]	0x0F	0x0000
configurable_rate_low_0 [15:0]	0x10	0x0001
configurable_rate_high_1 [15:0]	0x11	0x0000
configurable_rate_high_0 [15:0]	0x12	0x0001
rate_tolerance_low [15:0]	0x13	0x0001
rate_tolerance_high [15:0]	0x14	0x0001
configurable_time_delta_1 [15:0]	0x15	0x0000
configurable_time_delta_0 [15:0]	0x16	0x0FFF
low_threshold_step [07:0]	0x17	0x01
high_threshold_step [07:0]	0x18	0x01
HitMapTestpattern_low_3 [15:0]	0x1A	0x0000
HitMapTestpattern_low_2 [15:0]	0x1B	0x0000

Table 4.5 TERZINA configurable parameters part 1.

- receiver_enable
  each bit enables the receiver module for the respective ASIC. During normal operation all ASICs are enabled (0b11111).

- configurable_rate_low_1 and configurable_rate_low_0
  parameter used in the rate controller module in order to set the desired event rate for the low threshold comparator. Being it 32 bit long it is divided over two memory locations.

- configurable_rate_high_1 and configurable_rate_high_0
  parameter used in the rate controller module in order to set the desired event rate for the high threshold comparator. Being it 32 bit long it is divided over two memory locations.

- rate_tolerance_low
  parameter used in the rate controller module in order to set the allowed deviation from the configured event rate for the low threshold.

Parameter name	Address	Default value
HitMapTestpattern_low_1 [15:0]	0x1C	0x0000
HitMapTestpattern_low_0 [15:0]	0x1D	0x0030
HitMapTestpattern_high_3 [15:0]	0x1E	0x0000
HitMapTestpattern_high_2 [15:0]	0x1F	0x0000
HitMapTestpattern_high_1 [15:0]	0x20	0x0000
HitMapTestpattern_high_0 [15:0]	0x21	0x0030
HitMapTestpatternSel [07:0]	0x22	0x00
SampleBitSizeCfg [02:0]	0x23	0b000
CellNumberCfg [01:0]	0x24	0b00
ReadEnableTimeDelay [07:0]	0x26	0x0A
HitMapTime [07:0]	0x27	0x05
AddressOfLastPixel [08:0]	0x28	0b000111111
HeaderData [07:0]	0x29	0x55
AlignmentWord1 [13:0]	0x2A	0b01010101010101
AlignmentWord2 [13:0]	0x2B	0b00000001111111
enable_high_th_pattern [09:0]	0x2C	0b1111111111
enable_low_th_pattern [09:0]	0x2D	0b1111111111

Table 4.6 TERZINA configurable parameters part 2.

- rate_tolerance_high
  parameter used in the rate controller module in order to set the allowed deviation from the configured event rate for the high threshold.

- configurable_time_delta_1 and configurable_time_delta_0
  parameter used in the rate controller module in order to configure the integration time for the rate calculation.

- low_threshold_step
  parameter used in the rate controller module in order to configure the threshold adjustment step for the low threshold.

- high_threshold_step
  parameter used in the rate controller module in order to configure the threshold adjustment step for the high threshold.

- HitMapTestpattern_low_3 and HitMapTestpattern_low_2 and HitMapTestpattern_low_1 and HitMapTestpattern_low_0

64 bit test hitmap, divided into four 16 bit words, that can be injected into the hitmap assembler for debugging purposes.

- HitMapTestpattern_high_3 and HitMapTestpattern_high_2 and HitMapTestpattern_high_1 and HitMapTestpattern_high_0
  same principle as before, but for the high threshold hitmap.
- HitMapTestpatternSel
  parameter disabled used for debug.
- SampleBitSizeCfg
  parameter used in the DATA handler module used to define the bit resolution of the ADC sampling. It is used by the logic to calculate the expected number of words from each packet.
- CellNumberCfg
  parameter used in the DATA handler module used to set the number of cell readout for each event. It is needed in order to calculate the expected number of words of each data packet.
- ReadEnableTimeDelay
  this parameter sets the delay, in the Trigger logic module, between the received Read Request signal and the read Acknowledge output from the FPGA.
- HitMapTime
  in the Hit Validator module this parameter sets the validity time of the newly received hitmap.
- AddressOfLastPixel
  this parameter is temporally disabled. It was intended to be used in the Data Handler module in order to raise a flag when the FPGA received the last active pixel of an ASIC.
- HeaderData
  configurable header sent by the ASICs at the beginning of the stream of information. During normal operation it is 0x55.
- AlignmentWord1
  Alignment word used in the receiver module during the deskewing phase. During normal operation it is 0b01010101010101.

- AlignmentWord2
  Alignment word used in the receiver module during the shift phase. During normal operation it is 0b000000011111111.

- enable_high_th_pattern
  parameter used in the hit validator module used to set the valid patterns for the high threshold triggers.

- enable_low_th_pattern
  parameter used in the hit validator module used to set the valid patterns for the low threshold triggers.

4.4.7 Rate controller

The rate controller module receives the hitmaps from the hitmap assembler and generates SPI commands in order to adjust the threshold of each pixel in such a way to keep the event rate in a configurable expected range. A counter running a 200 MHz produces a time strobe and reset itself each time it reaches the configurable_time_delta parameter. This is the integration time of the rate calculation. For each pixel a counter is increased every time a hitmap with a respective HIGH bit is received. The system has a logic that ensures that each hitmap is counted only once. At each time strobe two condition are checked for both, high and low threshold. Below the example for the low threshold, the same applies for the high threshold. If the counter value of pixel k is greater than the sum of the desired rate and the tolerance:

$$hit\_counter\_low\_th(k) > (configurable\_rate\_low + rate\_tolerance\_low)$$

then the threshold configuration is increased by a step:

$$low\_th(k) <= low\_th(k) + low\_threshold\_step$$

On the other hand, if the counter value of pixel k is less than the difference of the desired rate and the tolerance:

$$hit\_counter\_low\_th(k) < (configurable\_rate\_low - rate\_tolerance\_low)$$

the threshold value is decreased by a step:

$$low\_th(k) <= low\_th(k) - low\_threshold\_step$$

For example, lets suppose that from simulations the expected trigger rate is in the order of (150 $\pm$ 15) Hz and the selected integration time is 10 s. The 200 MHz counter frequency leads to a period per unit of 5 ns. The configurable_time_delta value will be 0x77359400.

$$\frac{10s}{5ns} = 0d2000000000 = 0x77359400$$

In this amount of time the pixel has to count 1500 events, thus the configurable_rate can be set to 0x5DC. The tolerance in this example is 10%, thus 150 events. The rate_tolerance can be configured to 0x96. A safety mechanism does not allow the threshold to be greater or lower than a configurable value. Each threshold can be read back using a command contained in the command executor module that will be described in the next sections. After each strobe a scan of each new value is performed and, if there are updates, an SPI command is generated for each pixel and stored in a FIFO.

4.4.8 Communication protocol to PC

In order to send and receive data to and form a computer a custom UDP IP [117] developed at INFN Turin was implemented into the design of the TERZINA readout firmware. The protocol is able to receive and send 32 bit words at real world speeds of $\approx$ 10 MBps. A cyclic redundancy check (CRC) [118] is used to detect errors in the data transmission and a set of counters ensures that there are no missing words. This IP will be used only during the development and test phase. In the final deployed version of the firmware the UDP module will be replaced by a QUAD SPI handler in order to be compatible with the already existing DPCU. This thesis will not focus on the details and inner mechanisms of the UDP IP.

4.4.9 Command executor

The command executor module is used to sort the commands received by the computer, and stored in a FIFO, to the relevant firmware section in order to be performed. In addition, once the command has been executed, a reply gets composed and sent back to the PC. The 32 bits words are divided in: command_id (4 bits), command_address (12 bits), and command_data (16 bits). Below a list of all the currently available commands sorted by command_id with a brief descriptions:

- 0x0 → SetConfigParameter
  is used to write parameters into the configuration module. Of the 12 bits available for the address only 8 are in use.
- 0x1 → ReadConfigParameter
  is use to read back the parameters from the configuration module.
- 0x2 → ResetConfigParameter
  is used to reset the parameters from the configuration module to their default value.
- 0x3 → ReadThreshold
  is used to read the threshold value from the rate controller module. If the most significant bit of the address is high then the logic is reading a low threshold, otherwise a high threshold is sent. The address of the pixel can be computed using the formula in equation 4.4.

  $$address = 64 \cdot X + K \tag{4.4}$$

  With X being the ASIC number and K being the pixel number of the ASIC. This means that the first channel of ASIC 0 has address 0x0 and the last pixel of ASIC 4 has address 0x13F.
- 0x4 → SendAsicReset
  is used to send a reset signal to one of the ASICs depending on the address value. Address 0x000 being ASIC 0 and 0x004 being ASIC 4.
- 0x5 → SendTestHitMapCmd
  is used inject the test hitmap, stored in the configuration module, into the

hitmap assembler for debugging purposes. Each bit of the command data from 9 down to 0 controls one ASIC for low or high threshold. If the bit is high the hitmap is sent. The logic allows to inject the hitmaps to all the outputs of the hitmap assembler simultaneously.

- 0x6 → SendSPIDataCmd
  the 28 bit word composed of address + data is sent to the SPI handler in order to be processed.

- 0x7 → AlignSerialData
  this command starts the automatic time calibration procedure for the serial communication.

- 0x8 → ReadNumberOfDataWords
  this command is used to read the number of data words stored in the TERZINA FIFO which are ready to be sent to the PC.

- 0x9 → ReadDataWord
  if the TERZINA FIFO is not empty this command replies with the fist word available, otherwise an error message is sent.

4.4.10 HITMAP and DATA handlers

The HITMAP handler and DATA handler modules, given their similar operating logic, will be described in the same section. The scope of this two modules is to decode the beginning and the end of a packet coming from the ASIC, to frame the data with an header and a footer, and to funnel the different streams from each chip to a single FIFO without mixing the packets. Given that the hitmap packets have always the same length and are much shorter in size, the respective handler module is less complex than the counterpart for the data. A series of five FSMs manage the write operation of five FIFOs (namely stg1) each one being 18 bits wide (1 bit for the end of packet message, 3 bits for the ASIC_id and 14 bits for the words). If a word full of zeros is followed by a second word starting with the header message, then a counter gets initialized and the words begin to be written into the respective FIFO. When the counter reaches the expected number of words of the hitmap packet, the write operation is stopped and the FSM goes back to idle waiting for the next event. The most significant bit of the last word is set to HIGH signalling the end of the

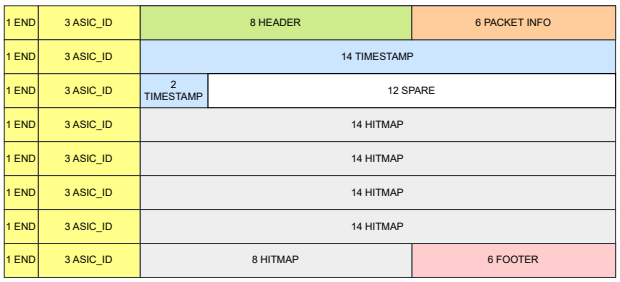


Fig. 4.21 Hitmap packet structure in the stg1 FIFO.

packet. The resulting data can be visualized in figure 4.21. A second FSM constantly

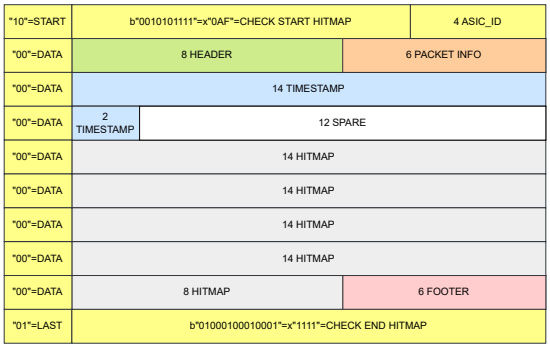


Fig. 4.22 Hitmap packet structure in the stg2 FIFO.

monitors the state of the first stage FIFOs. If one of them is not empty the packet begins to be moved to a second FIFO (namely stg2), common for all ASICs, 16 bits wide. The logic adds a start of packet and end of packet message. In figure 4.22 is depicted the diagram of the resulting packet. When finishing the write process of an hitmap packet, the logic performs a check of all the other stg1 FIFOs before coming back to the one just read. This is done in order to balance the occupancy evenly on all memories. In figure 4.23 a diagram with the different stages of FIFOs of the DATA and HITMAP handlers is shown. The DATA handler module uses the sample principle. The only notable difference is that the data packet has a different number of words which is function of the number of samples and digitization resolution. Using the CellNumberCfg and SampleBitSizeCfg configuration parameters the logic can calculate the expected number of words per packet. To be noted that the waveform of each pixel is managed as a different packet. This approach adds more complexity in the DPCU software which has to recombine the information of the different ASIC pixel using the timestamp and the respective addresses, however it allows a much more evenly data readout from the stg1 FIFOs. In figure 4.24 an example of a data

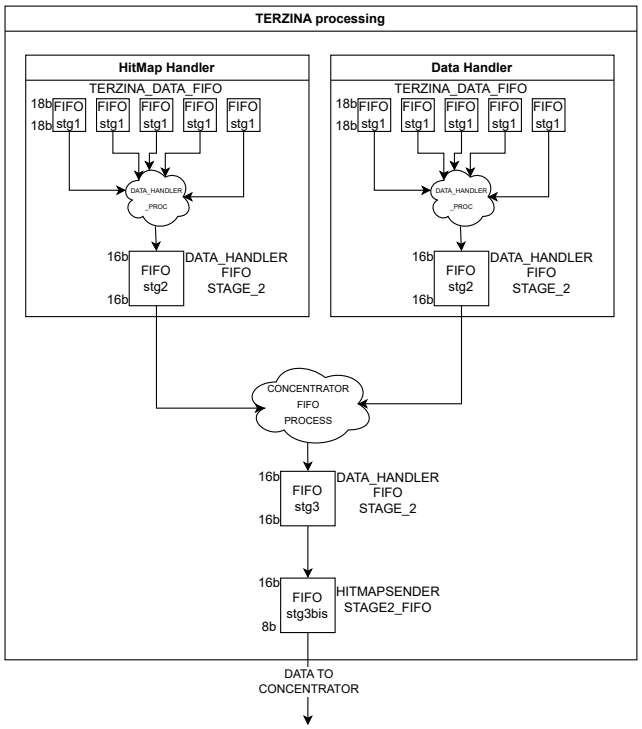


Fig. 4.23 HITMAP handler and DATA handler FIFO diagram structure.

packet with 32 samples and 8 bit resolution written on the stg1 FIFO is shown. While in figure 4.25 the same packet is shown, but moves to the stg2 FIFO. The start of packet and end of packet words can be seen at the top and bottom of the diagram. From the stg2 FIFOs of the two handlers the packets are funneled into a common stg3 FIFO. When moving the data the logic always gives priority to the data packets. The hitmaps packets are sent only if the data stg2 FIFO is empty. The last step is a CDC FIFO (namely stg3bis) with asymmetrical write and read ports which is used to synchronize the stream of data with the communication protocol.

4.4.11 SPI handler

The SPI handler module takes as input three streams of SPI commands data from three different FIFOs, one from the rate controller module, one from the receiver module, and one from the command executor module. Each command is 28 bits wide, the 4 most significant bits identify the ASIC targeted by the command, while the other 24 bits carry the payload. The SPI handler logic reads the commands from the three input FIFOs and, following the priority:

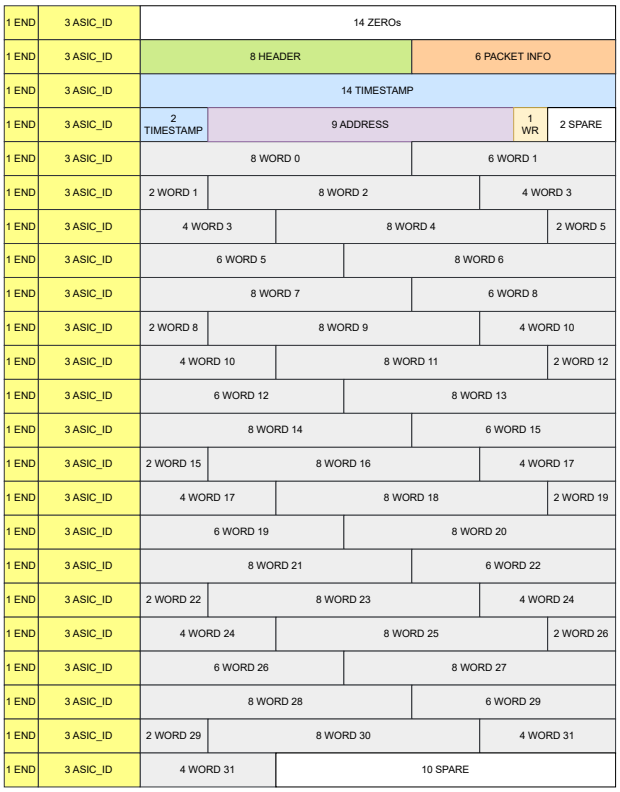


Fig. 4.24 Data packet structure in the stg1 FIFO.

1. rate controller;
2. command executor;
3. receiver module;

sorts each word to a set of five FIFOs (one for each FIFO). A SPI IP driver module is instantiated for each ASIC intended to be managed. Each driver is managed by an independent FSM which controls the read operations from the respective FIFO. The system does not use a common MOSI, MISO, and CLK output ports for all the chips for redundancy purposes. In the event of a failing trace or chip, having a separate output set for each ASIC would limit the loss to only the affected chip.

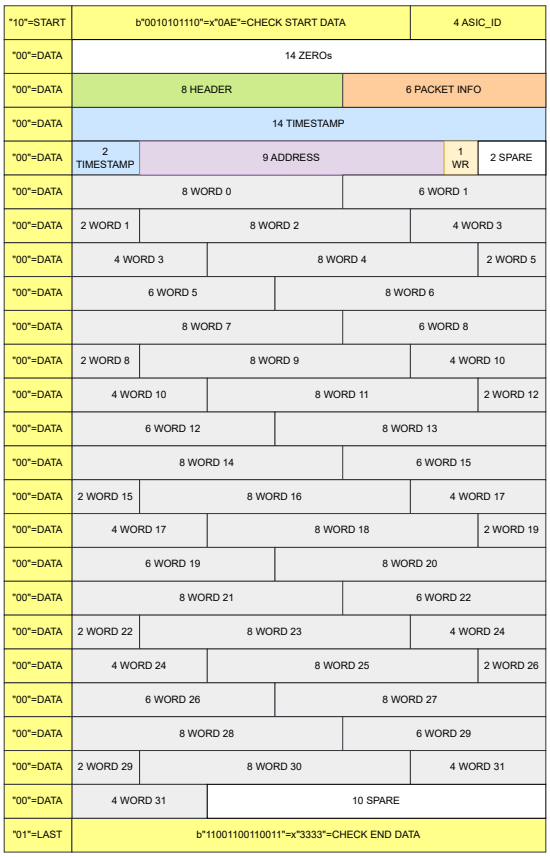


Fig. 4.25 Data packet structure in the stg2 FIFO.

4.5 Future work

At the time of writing this thesis the firmware is still under development. Below a list of the main features yet to be implemented in the future releases:

- SPI IN
  the current readout firmware is not able to receive SPI packets from the ASIC. This means that it is not possible to read back registers from the chip or to do a handshake test after sending a command.

- ASIC PARAMETER SET
  the configuration modules misses a set of parameters that will have to be passed to the ASIC for configuration. This section of code will be developed after the final MIZAR submission.

- TEST PULSE output
  For debugging purposes the MIZAR ASIC has a Test Pulse (TP) input. The firmware logic does not have a logic to control this signal yet. This will be developed after the ASIC submission.

- MANUAL THRESHOLD setting
  at the moment the logic does not allow to set manually a desired threshold pixel by pixel. This feature will be added to the firmware in the next version.

- QUAD SPI protocol
  currently the data is transmitted to the PC via the INFN UDP IP. One of the future additions to the logic will be the QUAD SPI communication protocol in addition to the already existing UDP. In this way with the same firmware it would be possible doing both, the tests in lab with a PC and the test with the DPCU.

- TRIPLE MODULAR REDUNDANCY
  given the expected use case of the TERZINA experiment the design of the ASIC and the FPGA readout firmware must take into consideration the expected ionizing dose absorbed by the electronics. In particular, in order to prevent the loss of physics data caused by single event upsets and single event latchups, portion of logic, as the hit validator or configuration module, should be designed with Error Correcting memory and triple modular redundant registers. Up until now the firmware development was focused more on the operational logic. The radiation resistant design solutions will be implemented into the design after the MIZAR submission, if allowed by the available FPGA resources and power budget.

Chapter 5

Results and considerations with the MIZAR FPGA readout firmware

This chapter will start with a few examples of the simulations performed during the development of the TERZINA readout firmware; this will be shown in section 5.1. Meanwhile, section 5.2 will delve into the description of the MIZAR emulator firmware. In sections 5.3 and 5.4 the laboratory setup assembled in Turin and the calibration tests performed will be described. In section 5.5 a brief overview on the software used for testing will be presented.

5.1 Simulation results

During the development of the TERZINA readout firmware the logic was extensively simulated using the VIVADO integrated XSIM simulator. In this section a few examples of the simulated results will be presented.

5.1.1 Data packet FIFOs

In figure 5.1 a simulation of the Stage 1 data FIFOs (one for each channel) is shown. Meanwhile, in figure 5.2, a simulation of the Stage 2 data FIFO is depicted. In this case 6 packets are written, configured with 32 cells and 8 bit resolution, and subsequently immediately read. To be noted how the word width is 18 bit for the

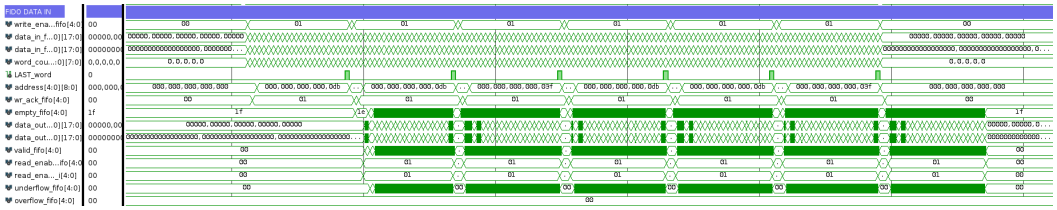


Fig. 5.1 Stage 1 data FIFO behaviour from simulation.

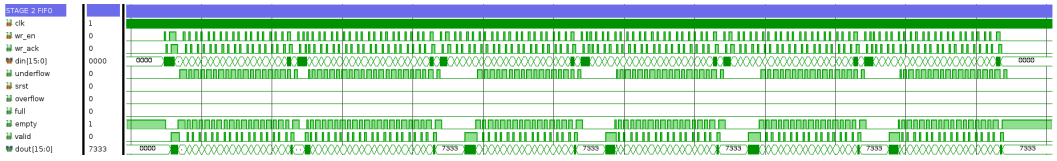


Fig. 5.2 Stage 2 data FIFO behaviour from simulation.

first stage and 16 bit for the second stage. Moreover, the difference between the write (57 MHz) and read (200 MHz) frequency results in the comb pattern seen in the stage 2 timing diagram.

5.1.2 Timing calibration FSM

In figure 5.3 a simulation of the receiver timing calibration procedure is shown. This is the same process depicted in figures 4.14 and 4.15. With the simulation it is possible to observe how the calculation of the average position is performed (in the image the result is position 0x9). However, it is worth noting how the transition from 0b1010 and 0b0101 is perfectly smooth in simulation, meanwhile, in real life (figure 4.14) there is a region where the de-serialized data is completely unusable.

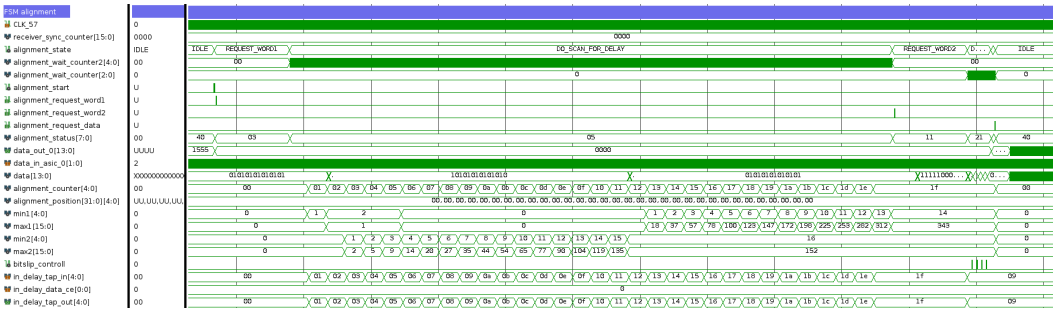


Fig. 5.3 Receiver timing calibration FSM logic from simulation.

5.1.3 SPI handler

In figure 5.4 a simulation of the ASIC 0 SPI module from simulation can be seen. In the timing diagram 5 words can be seen coming from the FGPA to the ASIC.

- 0x110000 → alignment request word1 hitmap (not present in simulation);
- 0x140000 → alignment request word1 data;
- 0x120000 → alignment request word2 hitmap;
- 0x150000 → alignment request word2 data;
- 0x130000 → alignment request data hitmap;
- 0x160000 → alignment request data data;

Those are the SPI messages sent by the readout module to the MIZAR chip in order to start the FSM and proceed with the next calibration step.

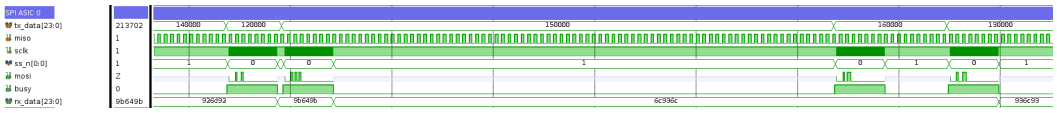


Fig. 5.4 SPI input output for ASIC 0 from simulation.

5.2 The MIZAR emulator FPGA firmware

In order to test on hardware the main features of the readout firmware, such as the timing calibration process and the data handler, before the final ASIC submission, I developed a second firmware to be uploaded on a Digilent Genesys 2 development board in order to emulate a simplified MIZAR. It is important to note that this emulator does not have any ADC feature, in fact it only sends configurable pre-determined hitmap and data packets in order to test the logic chain and readout software. In this section a brief overview of the emulator architecture and its main features will be presented. In figure 5.5 a diagram of the main blocks of the MIZAR emulator is shown. Six main components can be noted:

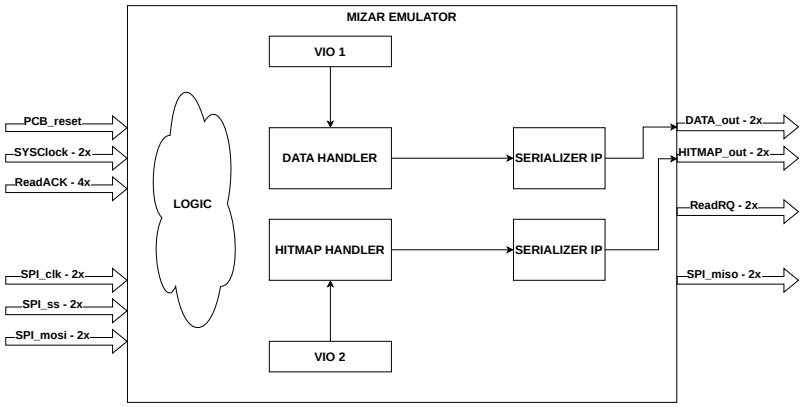


Fig. 5.5 MIZAR emulator firmware simplified block diagram.

- VIO 1 and VIO 2 which are used to set the configuration parameters and to control the operations;
- DATA HANDLER, which is used to generate the data packets;
- HITMAP HANDLER, which is used to generate the hitmap packets;
- two SERIALIZER IPs, which are used to convert the 14 bit words into a 400 MHz DDR bit stream

5.2.1 HITMAP handler

The HITMAP handler module developed for the MIZAR emulator is used to generate 14 bit words with the same data format and timing as the real chip. In figure 5.6 a diagram depicting the logic FSM can be seen. The system can be configured to send:

- a pseudo-random data stream;
- a configurable fixed word;
- the two patterns used for timing calibration;
- the 9 words of the hitmap packet;

in addition a configurable counter allows to send the hitmap packet periodically, thus simulating whatever trigger rate desired.

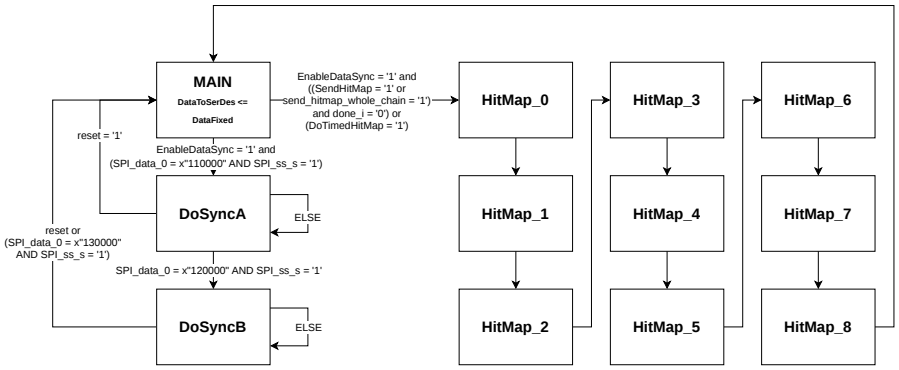


Fig. 5.6 MIZAR emulator HITMAP handler FSM block diagram.

5.2.2 DATA handler

The DATA handler module of the MIZAR emulator is based on the same principle of the HITMAP handler. It has the same capabilities with the addition that, given the number of pixel of the ASIC, the data packet can be repeated a configurable number of times in order to simulate the transmission of more than one channel. In figure 5.7 a diagram of the DATA handler FSM is shown.

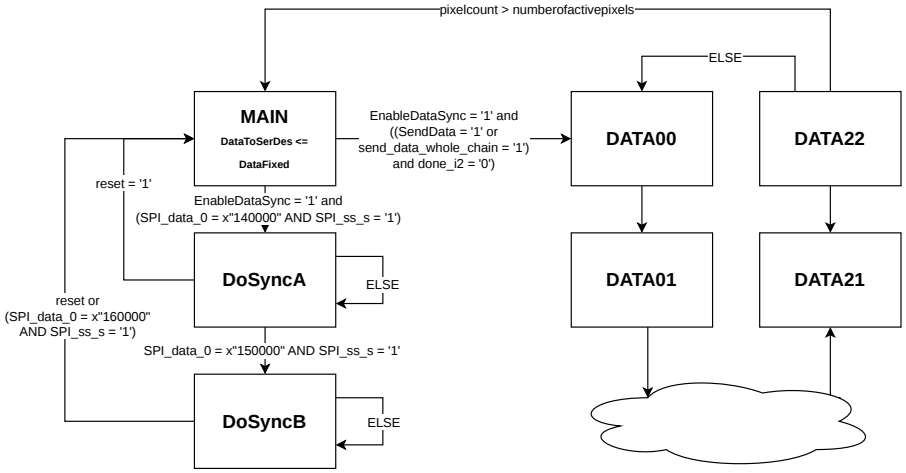


Fig. 5.7 MIZAR emulator DATA handler FSM block diagram. The cloud shape summarizes DATA states from 02 to 20.

5.2.3 Main logic

The main logic of the MIZAR emulator is used to manage simultaneously the work of HITMAP and DATA emulator and the read request operations. In figure 5.8 a diagram of used FSM is shown. Once the transmissions lines are calibrated, operation that up to this point has to be initialized manually by the user, the system is ready to send packets to the readout FPGA. From the IDLE state the logic waits for the assertion of the simulate_whole_chain signal. When this happens the ReadRQ gets asserted HIGH and the logic waits for a reply from the readout FPGA. If the received ReadACK signal is "10" (read declined) or the timeout limit is reached, the logic goes back to IDLE, otherwise it starts sending the hitmap. After the packet transmission is done the logic asserts again the ReadRQ signal, this time for the data transmission. If the readout FPGA confirms the detection of an interesting pattern and replies to the MIZAR emulator with a ReadACK signal asserted to "01" (read accepted) or "11" (read forced) then the data packets start to be sent.

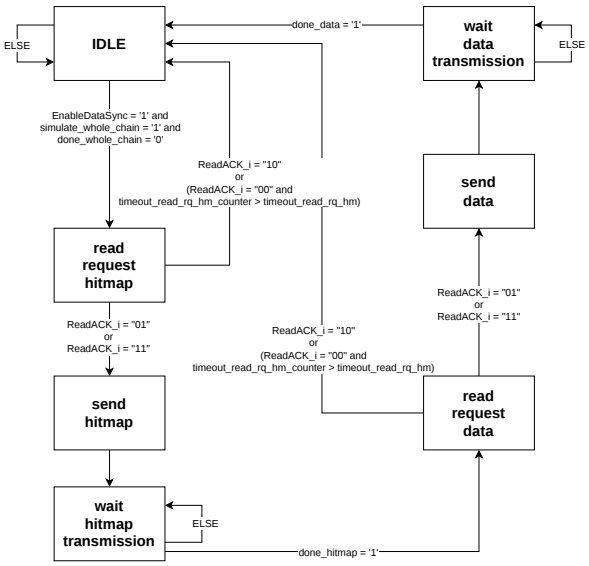


Fig. 5.8 MIZAR emulator main FSM diagram.

5.2.4 Emulator control panel

In figure 5.9 the MIZAR emulator VIO control panel is shown. Below a list of all the configurable parameters:

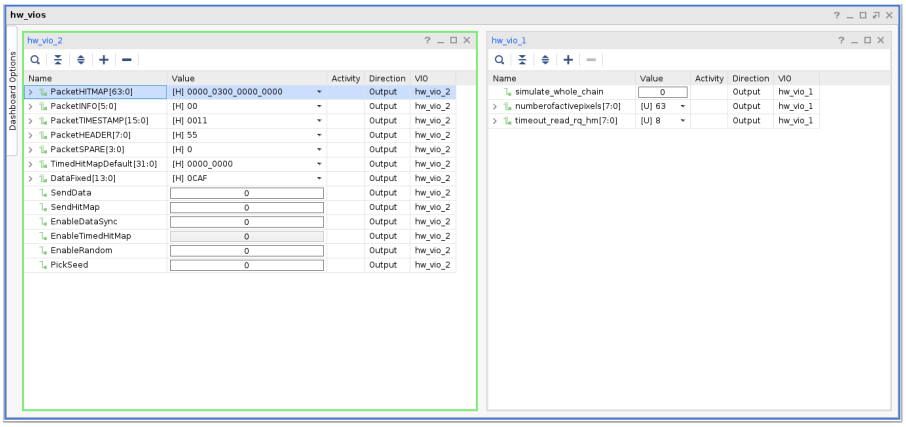


Fig. 5.9 MIZAR emulator VIO control panel using XILINX Vivado 2021.

- simulate hole chain → (Active HIGH button) When active this button start the whole data transmission process: read request, hitmap transmission, read request, and data transmission.
- number of active pixels → Sets the number of active pixels of the chip, thus the number of times that the data packet has to be repeated when sending data.
- timeout read rq hm → Sets the timeout time of the read request reply, counted in numbers of clock cycles at 200 MHz.
- Packet HITMAP → HITMAP used for the packet generation.
- Packet INFO → Additional INFO used for the packet generation.
- Packet TIMESTAMP → Timestamp used for the packet generation for both, HITMAP and data.
- Packet HEADER → Transmission header, during normal operation is 0x55.
- Packet SPARE → Spare bits.
- Timed HITMAP Default → If the timed HITMAPs are enables this parameter sets the period, expressed in number of clock cycles, between two packets. For example, in order to configure a rate of 100 Hz, thus a period of 10 ms, the parameter should be set at 0d2000000 = 0x001E8480.
- Data Fixed → Configurable default transmitted word.

- Send Data → (Active HIGH button) When active this button triggers the transmission of a data packet.
- Send HITMAP → (Active HIGH button) When active this button triggers the transmission of a HITMAP packet.
- Enable Data Sync → (Toggle button) This parameter enables the FSM logic. When in operation this must always be HIGH.
- Enable Timed HITMAP → (Toggle button) When HIGH the HITMAP packets are sent regularly with a period controlled by the Timed HITMAP Default parameter.
- Enable Random → (Toggle button) When HIGH pseudo-random data is transmitted.
- Pick Seed → (Active HIGH button) When active the internal timestamp counter gets frozen and used as seed for the pseudo-random data generator.

5.3 The laboratory setup

A test setup was assembled in laboratory at INFN Turin. A XILINX KC705 development board [119], shown in figure 5.10, and a Digilent Genesys 2 development board [120], shown in figure 5.11, are used. The first one is used for the readout firmware, while the second one is used for the MIZAR emulator. Both boards mount a KINTEX 7 XILINX FPGA chip [121]. In figure 5.12 a picture of the working environment is shown, meanwhile, in figure 5.13 a diagram of the connections of the test setup is shown. In figure 5.14 is depicted an ILA recorded event from a test in which the MIZAR emulator requested a HITMAP transmission, the readout FGPA accepted, the HITMAP packet was transmitted, and the interesting pattern in the HITMAP was recognized.

5.4 Calibration tests results

One of the most interesting tests performed in laboratory using the complete setup concerned the transmission timing calibration using different length of cables. In

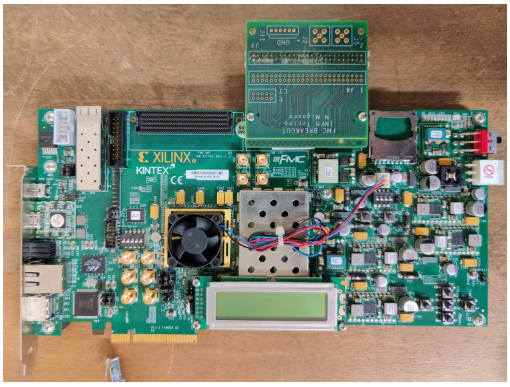


Fig. 5.10 XILINX KC705 development board.

Fig. 5.11 Digilent Genesys2 development board.

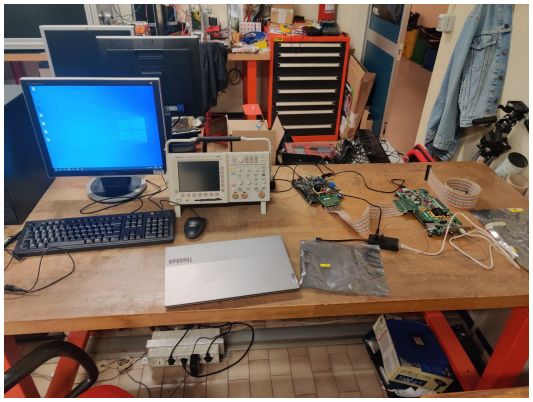


Fig. 5.12 Test setup in laboratory at INFN Turin.

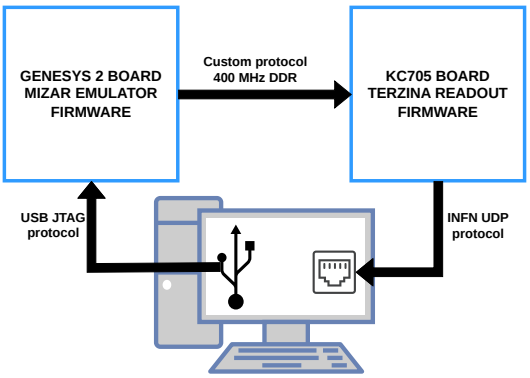


Fig. 5.13 Diagram of the laboratory setup configuration

particular three different cables were used, 0.5 m, 1 m, and 1.5 m. In figures 5.15 and 5.16 two cases can be seen, the first one, to the left, with a 1 m long cable, while the second one, to the right, with a 1.5 m cable. The system was able to calibrate and maintain a reliable connection up to the 1 m cable. The performance of the link was evaluated by transmitting a known pattern over a period of 30 min and counting the errors. If in the 30 min test period there were no errors the test was considered successful. However, it is important noting that this test is only relevant for validating the calibration FSM, since the hardware of the actual experiment will be completely different.

- The output driver of the MIZAR ASIC will have different specifications with respect to the KINTEX 7 FPGA.

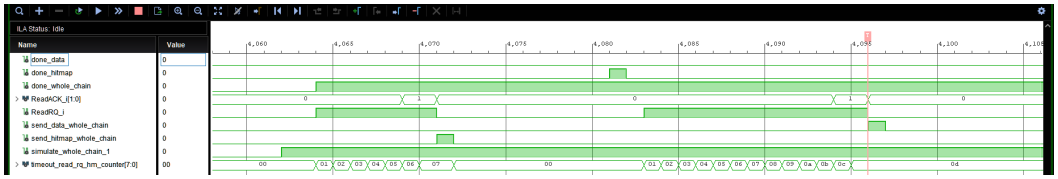


Fig. 5.14 Whole trigger chain event recorded by ILA.

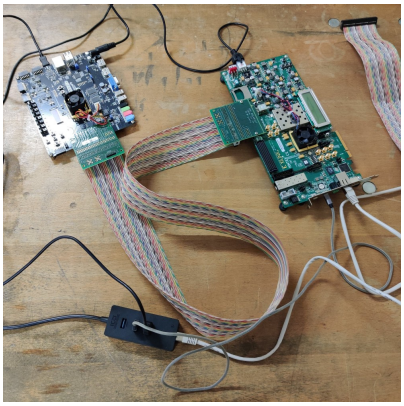


Fig. 5.15 Timing calibration test bench with 1 m cable.

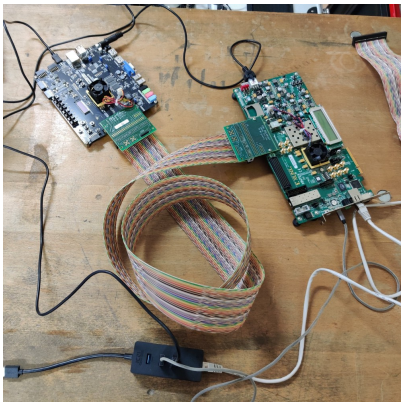


Fig. 5.16 Timing calibration test bench with 1.5 m cable.

- The TERZINA experiment will use shielded low loss high quality cabling with respect to the not shield flat twisted pairs wires used in laboratory.
- Given the physical constraints of the FPGA board IO (FMC connector), a INFN developed breakout PCB board which is visible in all figures was used as an adaptor. This increased the number of connections, thus lowering the signal integrity. The TERZINA experiment will use space tested shielded locking connections rated for high bandwidth signals.

In figure 5.17 an ILA recorded calibration process is shown.

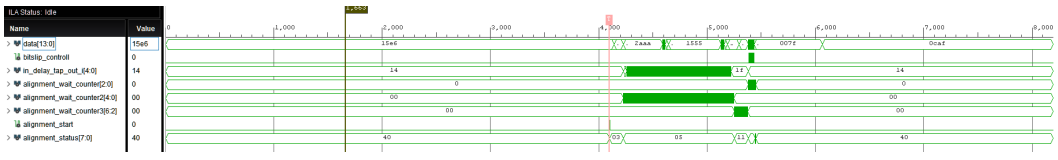


Fig. 5.17 Calibration process performed with 1 m cable recorded from ILA.

5.5 The readout software

In order to configure, control, and test the readout firmware it is necessary to send and receive UDP packets. In the first test the commands were sent manually one by one or with simple python scripts. The replies were recorded using the Wireshark [122] packet sniffer software. Subsequently a LabVIEW [123] readout software started to be developed. In the next sections a couple of packets recorded with Wireshark and the GUI of the LabVIEW readout software will be presented.

5.5.1 WireSHARK recorded packets

In figure 5.18 a screenshot showing an UDP packet coming from the readout FPGA to the PC containing both, the hitmap and data packets for more than one pixel, can be seen. To be noted that UDP IP used for the project has a limit on the number of bytes per packet which is lower than the total size of the TERZINA full data payload. This means that the complete data transmission has to be slip into more than one packet. In figure 5.19 this behaviour is shown. In light blue two TERZINA payload

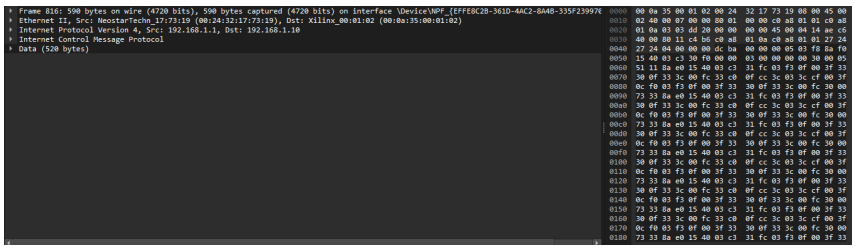


Fig. 5.18 UDP event payload (HITMAP + data) recorded by Wireshark.

(HITMAP packet + 64 pixel packet configured with 32 cells and 8 bit resolution) are shown. Each payload is split into 4 packets, the first three at full size (1058 bytes) and the last one at $\approx$ 100 bytes. The UDP protocol has a configurable IP, thus more than one readout board can be controlled with a PC. Ideally the whole camera plane, 10 MIZAR ASICs divided between two readout modules, could be read using a PC.

5.5.2 LabVIEW panels

The microelectronics group at INFN Turin has a great expertise in developing and using LabVIEW for ASIC debugging, testing and readout applications. In this

No.	Time	Source	Destination	Protocol	Length	Destination Por	Source Port
714	236.918700	192.168.1.1	192.168.1.10	ICMP	98	10020	10020
725	243.871218	192.168.1.10	192.168.1.1	UDP	1058	10020	10020
726	243.871218	192.168.1.10	192.168.1.1	UDP	1058	10020	10020
727	243.871218	192.168.1.10	192.168.1.1	UDP	1058	10020	10020
728	243.871218	192.168.1.10	192.168.1.1	UDP	98	10020	10020
729	243.871745	192.168.1.1	192.168.1.10	ICMP	590	10020	10020
730	243.871816	192.168.1.1	192.168.1.10	ICMP	590	10020	10020
731	243.871864	192.168.1.1	192.168.1.10	ICMP	590	10020	10020
732	243.871899	192.168.1.1	192.168.1.10	ICMP	126	10020	10020
812	302.412617	192.168.1.10	192.168.1.1	UDP	1058	10020	10020
813	302.412617	192.168.1.10	192.168.1.1	UDP	1058	10020	10020
814	302.412617	192.168.1.10	192.168.1.1	UDP	1058	10020	10020
815	302.412617	192.168.1.10	192.168.1.1	UDP	118	10020	10020
816	302.412702	192.168.1.1	192.168.1.10	ICMP	590	10020	10020
817	302.412755	192.168.1.1	192.168.1.10	ICMP	590	10020	10020
818	302.412784	192.168.1.1	192.168.1.10	ICMP	590	10020	10020
819	302.412809	192.168.1.1	192.168.1.10	ICMP	146	10020	10020

Fig. 5.19 Overview of two complete events divided into 4 UDP packets each.

context I decided to use the same philosophy for the MIZAR readout and test software. In this section a few examples of the LabVIEW panel will be presented. In figure 5.20 the main front panel is shown. From here the user can send custom commands and receive the respective reply. In addition, knowing the address, each parameter can be configured, read and reset to the default value. The timing calibration procedure (alignment) can be manually started. From the Rate Controller

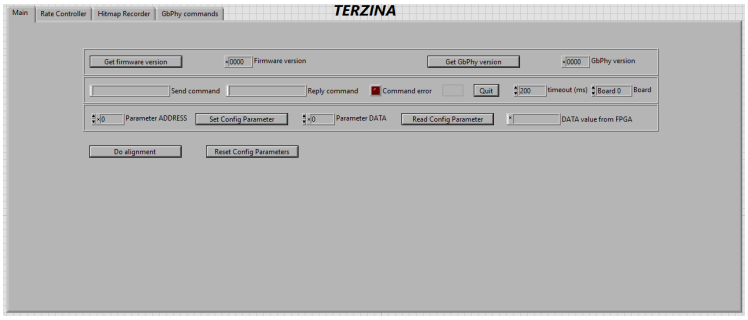


Fig. 5.20 TERZINA LabVIEW readout software main panel.

panel, shown in figure 5.21, the user can monitor in time the value of the threshold of a pixel. The plot shown to the left displays on the Y axis the threshold value and on the X axis the time expressed in number of samples from the beginning of the acquisition. This panel, used in accordance with the timed HITMAP feature of the MIZAR emulator, can be used to check the automatic threshold configuration logic of the readout firmware. In figure 5.22 the TERZINA HITMAP decoder panel is shown, while in figure 5.23 an example of a decoded HITMAP is displayed. All the data is not only displayed on screen but also saved on a TDMS [124] file. Lastly, in figure 5.24 the configuration panel for all the UDP IP settings is shown. Here the user can set parameters such as the timeout time between events, the maximum length of a packet, the UDP port value for the events packets and the relative MAC

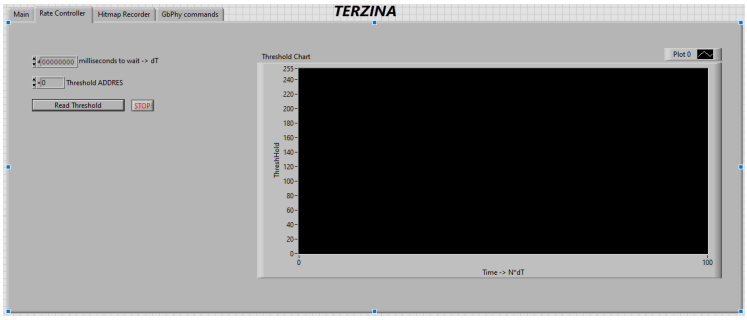


Fig. 5.21 TERZINA LabVIEW readout software Rate Controller panel.

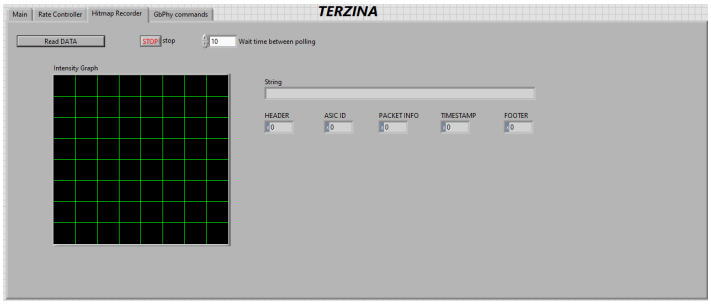


Fig. 5.22 TERZINA LabVIEW readout software HITMAP Decoder panel.

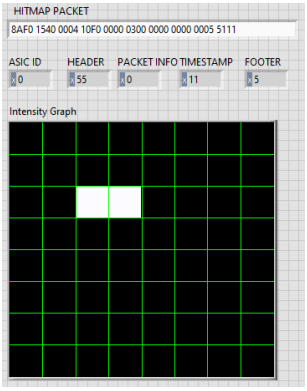


Fig. 5.23 HITMAP Decoder with example packet

address and IP. Generally those commands are intended only for debugging, during normal operation the user is expected to use the default settings.

5.6 Considerations and future work

In this section some considerations on the performed test and future work to be done will be presented.

5.6.1 Considerations on performed tests

At the time of writing this thesis the TERZINA readout firmware is in a complete preliminary phase. The main features needed for the communication between ASIC and PC have been implemented and tested with simulations. Most of the feature have been also tested in hardware using the MIZAR emulator. This includes:

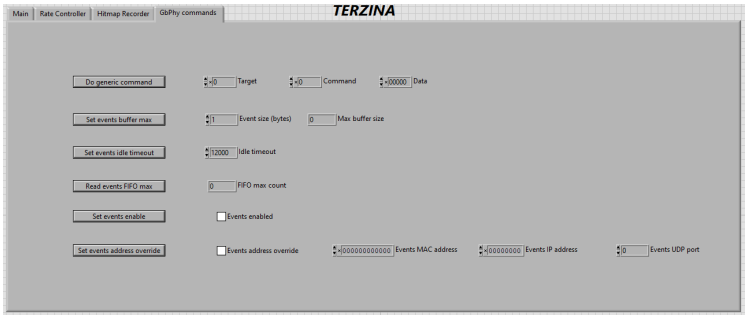


Fig. 5.24 TERZINA LabVIEW readout software Configuration panel.

- the configuration of the parameters;
- the calibration procedure;
- the HITMAP validation algorithm;
- the HITMAP and data packet handler;
- the automatic threshold adjustment logic.

Until now all the tests have given positive results.

5.6.2 Future work

The first version of the readout firmware can be considered almost complete. Until the submission of the MIZAR ASIC the future development effort will be focused towards the LabVIEW readout software which has still limited functionalities. In particular, before handling a real chip, the system should be capable of:

- saving configuration files and automatically load them into the ASIC at each start-up;
- visualize and save full waveform data;
- perform preliminary data analysis during acquisition in order to monitor data quality in real time;
- control and manage data acquisition for more than one readout board at the same time.

Chapter 6

Conclusions

This thesis presented two different readout systems, one for the ZDC detector of the ALICE experiment at CERN and one for the TERZINA experiment of the NUSES collaboration. Both readout designs have in common the use of an FPGA system for real time trigger elaboration, data handling and configuration management. This chapter will elaborate on the results obtained with both collaborations, on the differences between the two projects and on the points that they have in common.

The ZDC detector of the ALICE experiment at CERN upgraded its entire readout system in order to cope with the event rate provided at P2 by LHC in Run 3. The new system was build using 8 IOxOS IFC_1211 carrier modules, each one hosting a 4 channel FMC ADC_3112 digitizer, a Xilinx Kintex Ultrascale xcku040-1ffva1156 user programmable FPGA, and two 4.8 Gbps SFP+ optical fiber transceivers which are used to transmit and receive data through the CERN developed GBT protocol. The electrical signal produced by the ZDC photomultipliers are digitized by the ADC; at the same time from the fiber the detector receives timing information from the CTP. The FPGA readout firmware combines digitized data and timing information, creates data packets, selects the relevant portions of the waveform and transmits them to the CRU. In particular the new system is capable to: preserve the time and charge resolution performance of Run 2 ($\approx$ 20% energy resolution for the single neutron peak and $\approx$ 0.35 ns time resolution); acquire data in self-triggered mode without dead at an average event rate in Pb–Pb collisions of 1.4 MHz. triggering efficiently in presence of a large signal dynamics; flagging physics events with a bunch spacing of 50 ns which is lower than the length of the signal of 60 ns; evaluate

for each orbit the average of the baseline in the bunches where no collisions take place and to monitor and compute in real time the collision rate for each channel independently. The FPGA firmware was designed in order to receive configuration commands from the optical fiber through SWT messages, to automatically calibrate the the shift between timing and amplitude information, and to monitor in real time the status of the GBT link. The system was successfully deployed at P2 in 2022; it acquired Pb–Pb data during the low intensity run in 2022 and during the full Run 3 heavy-ion campaign in 2023 at a center-of-mass energy of 5.36 TeV per pair of nucleons. The system was always able to sustain the data acquisition at the event rate provided by LHC, in addition the energy resolution for the 1 neutron peak improved from $\approx 20\%$ in Run 1 and Run 2 to $\approx 16\%$. This readout apparatus is expected to take data until the end of Run 3 in 2025.

The MIZAR ASIC, designed by the INFN microelectronics group in Turin, is a 64-channels chip for SiPMs readout built in 65-nm CMOS technology. The IC is expected to be used as a technology demonstrator for the use of SiPMs in space. The novelty of the ASIC consists in the elaborate trigger system which comprehends a configurable double threshold for each pixel and a generation of the map of all the hits (called hitmap). Before the beginning of the digitization phase the data is stored in a 256 cells wide analog memory. The ASIC can be configure to digitize the samples with a resolution ranging from 8 up to 12 bit and each event can be configured to use 32, 64 or 256 memory cells. The FPGA firmware designed for its readout is capable to: manage the complete readout of up to 5 MIZAR ASICs; calibrate automatically the data transmission delay lines in order to compensate for variations in temperature; deserialize hitmaps packets and evaluate in less than 30 ns the presence of an relevant pattern; manage the handshake operations with the ASIC in order to start the digitization process; receive, process, store the packets (data and hitmap), and send them to the data processor central unit for further analysis; monitor in real time the event rate for each channel and to apply changes to the both thresholds in a configurable allowed range. The system was tested with simulations and in hardware using a FPGA ASIC emulator. The performance of the readout system does not generate any bottleneck for the ASICs operation. Tests with the actual IC are expected for 2025.

Both readout systems process an analog signal, digitize it and evaluate the presence

of relevant physics information through a custom algorithm. However, on the one hand the ZDC readout has broad constraints on power consumption and data throughput. It is not exposed to ionizing radiation and is located in a VME crate on a water cooled rack. During the development of the firmware no time and effort was spent into over-optimizing the power consumption of the logic or into enabling error detecting and correcting logic other than the FEC algorithm that is used in the upwards and downwards optical links. On the other hand the MIZAR readout has constraint in both, data throughput and power. In addition is expected to be exposed to ionizing radiation during operation. This meant that the design process had to make use of more elaborated techniques. In order to save power the logic runs at the lowest possible clock frequency for each module, in addition clock gating was implemented for de-activated modules. ECC memory was used for the main FIFOs and TMR is being considered for the data handling and configuration FMSs. In order to reduce data throughput the readout can be configured to send a limited number of ASICs. The ZDC signal processing done within the FPGA is somehow simpler. Due to the less stringent bandwidth constraints much of the processing is done in the reconstructions programs that run on a linux farm. On the other hand, due to the fast data rate, the FPGA is run at higher clock frequency and care has to be taken in dealing with different clock domains when interfacing with different systems of the DAQ.

These two projects showcase the flexibility of FPGA chips and firmware in adapting to different constraints of the usage scenario, either emphasizing speed or frugality on resource consumption. This allows to implement flexible and efficient autotrigger algorithms that allow to reach the goal of acquisition of the full data without dead time.

References

[1] J. J. Thomson. Xl. cathode rays. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 44(269):293–316, 1897.

[2] E. Rutherford. Lxxix. the scattering of α and β particles by matter and the structure of the atom. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 21(125):669–688, 1911.

[3] N. Bohr. I. on the constitution of atoms and molecules. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 26(151):1–25, 1913.

[4] Sean F. Johnston. Peter galison, image and logic: A material culture of microphysics. *Science and Public Policy*, 26:75–76, 1999.

[5] Ernest Rutherford and H Geiger. An electrical method of counting the number of α-particles from radio-active substances. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 81(546):141–161, 1908.

[6] Georges Charpak, Roger Bouclier, Tullio Bressani, Jean Favier, and Crtomir Zupancic. The use of multiwire proportional counters to select and localize charged particles. *Nucl. Instrum. Methods*, 62:262–268, 1968.

[7] Steffen Georg Weber and Anton Andronic. ALICE event display of a Pb-Pb collision at 2.76A TeV. General Photo, 2015.

[8] Robert Hofstadter. The detection of gamma-rays with thallium-activated sodium iodide crystals. *Phys. Rev.*, 75:796–810, Mar 1949.

[9] B. Dolgoshein, V. Balagura, P. Buzhan, M. Danilov, L. Filatov, E. Garutti, M. Groll, A. Ilyin, V. Kantserov, V. Kaplin, A. Karakash, F. Kayumov, S. Klemin, V. Korbel, H. Meyer, R. Mizuk, V. Morgunov, E. Novikov, P. Pakhlov, E. Popova, V. Rusinov, F. Sefkow, E. Tarkovsky, and I. Tikhomirov. Status report on silicon photomultiplier development and its applications. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 563(2):368–376, 2006. TRDs for the Third Millenium.

[10] R. Santonico and R. Cardarelli. Development of resistive plate counters. *Nuclear Instruments and Methods in Physics Research*, 187(2):377–380, 1981.

[11] R. Arnaldi, A. Baldit, N. Bastid, G. Blanchard, E. Chiavassa, P. Cortese, G. Dellacasa, N. De Marco, P. Dupieux, B. Espagnon, J. Fargeix, M. Gallio, L. Luquin, F. Manso, V. Metivier, A. Musso, A. Piccotti, A. Rahmani, V. Ramillen, O. Roig, L. Royer, E. Scomparin, C. Soave, and E. Vercellin. Study of resistive plate chambers for the alice dimuon spectrometer. *Nuclear Physics B - Proceedings Supplements*, 78(1):84–89, 1999. Advanced Technology and Particle Physics.

[12] Michael Deveaux. Development of CMOS Monolithic Active Pixel Sensors for the ALICE-ITS Outer Barrel and for the CBM-MVD. Development of CMOS Monolithic Active Pixel Sensors for the ALICE-ITS Outer Barrel and for the CBM-MVD. *PoS*, VERTEX2015:045, 2015.

[13] Gianluca Aglieri Rinella. The ALPIDE pixel sensor chip for the upgrade of the ALICE Inner Tracking System. *Nucl. Instrum. Methods Phys. Res., A*, 845:583–587, 2017.

[14] A Szczepankiewicz. Readout of the upgraded ALICE-ITS. *Nucl. Instrum. Methods Phys. Res., A*, 824:465–469, 2016.

[15] ALICE TPC Collaboration. The upgrade of the ALICE TPC with GEMs and continuous readout. *JINST*, 16(03):P03022, 2021. 88 pages, 60 figures.

[16] G Darbo, P Farthouat, and A Grillo. Readout architecture of the ATLAS upgraded tracker. *CERN*, 2008.

[17] Caen 2730 digitizer family product page. https://www.caen.it/subfamilies/2730-digitizer-family/.

[18] Teledyne sp devices adq214-dcln product page. https://www.spdevices.com/en-us/Products_/Pages/ADQ214-DCLN.aspx.

[19] Marta Bagatin. *Ionizing Radiation Effects in Electronics: From Memories to Imagers*. CRC Press, December 2015.

[20] The see problem; sponsored by nasa headquarters code qw february 15 1996, section 1. kenneth a. label. https://radhome.gsfc.nasa.gov/radhome/papers/seeca1.htm.

[21] 20th annual single event effetcs (see) symposium. san diego, marriot la jolla, april 12-14 2011. https://radhome.gsfc.nasa.gov/radhome/see/2011/seesummary.cfm.

[22] The see problem; sponsored by nasa headquarters code qw february 15 1996, section 4. kenneth a. label. https://radhome.gsfc.nasa.gov/radhome/papers/seeca4.htm.

[23] The see problem; sponsored by nasa headquarters code qw february 15 1996, section 6. kenneth a. label. https://radhome.gsfc.nasa.gov/radhome/papers/seeca6.htm.

[24] Qiang Huang and Jin Jiang. An overview of radiation effects on electronic devices under severe accident conditions in npps, rad-hardened design techniques and simulation tools. *Progress in Nuclear Energy*, 114:105–120, 2019.

[25] W Cary Huffman and Vera Pless. *Fundamentals of error-correcting codes*. Cambridge Univ. Press, Cambridge, 2003.

[26] Triple modular redundancy generator. https://tmrg.web.cern.ch/tmrg/.

[27] Ayman Ahmed. New fpga blind scrubbing technique. In *2016 IEEE Aerospace Conference*, pages 1–9, 2016.

[28] M Fras, H Kroha, O Reimann, B Weber, and R Richter. Use of triple modular redundancy (tmr) technology in fpgas for the reduction of faults due to radiation in the readout of the atlas monitored drift tube (mdt) chambers. *Journal of Instrumentation*, 5(11):C11009, nov 2010.

[29] Bridgford Brendan, Carmichael Carl, and Wei Tseng Chen. Single-Event Upset Mitigation Selection Guide. Technical report, XILINX, March 2008.

[30] Chip hall of fame: Xilinx xc2064 fpga, hardware that can transform itself on command has proven incredibly useful. https://spectrum.ieee.org/chip-hall-of-fame-xilinx-xc2064-fpga.

[31] megachips fpga to asic conversion. https://www.megachips.co.jp/english/product/asics/solution/fpga/.

[32] Rtg4 radiation-tolerant fpgas. https://www.microchip.com/en-us/products/fpgas-and-plds/radiation-tolerant-fpgas/rtg4-radiation-tolerant-fpgas.

[33] Space-grade amd kintex ultrascale xqr fpga family. https://www.amd.com/en/products/adaptive-socs-and-fpgas/fpga/kintex-ultrascale-xqr.html.

[34] Lew Kowarski. *An account of the origin and beginnings of CERN*. CERN Yellow Reports: Monographs. CERN, Geneva, 1961.

[35] Vittorio Frigo. LHC map in 3D. Schéma du LHC en 3D. AC Collection. Legacy of AC. Pictures from 1992 to 2002., 1997.

[36] Ewa Lopienska. The CERN accelerator complex, layout in 2022. Complexe des accélérateurs du CERN en janvier 2022. *CERN*, 2022. General Photo.

[37] CERN. *Proceedings of the ECFA-CERN Workshop : Large Hadron Collider in the LEP tunnel: Lausanne, Switzerland 21 - 27 Mar 1984. ECFA-CERN Workshop on Large Hadron Collider in the LEP tunnel*, Geneva, 1984. CERN.

[38] ATLAS collaboration. ATLAS: letter of intent for a general-purpose pp experiment at the large hadron collider at CERN. *CERN*, 1992.

[39] M. Della Negra. CMS: letter of intent by the CMS Collaboration for a general purpose detector at LHC. Technical report, CERN, Geneva, 1992.

[40] ALICE Collaboration. Letter of Intent for A Large Ion Collider Experiment [ALICE]. Technical report, CERN, Geneva, 1993.

[41] H Dijkstra, Hans Jürgen Hilke, Tatsuya Nakada, and Thomas Ypsilantis. LHCb Letter of Intent, LHCb Collaboration. Technical report, CERN, Geneva, 1995.

[42] S Baird. Accelerators for pedestrians; rev. version. Technical report, CERN, Geneva, 2007.

[43] ALICE Collaboration. Alice upgrades during the lhc long shutdown 2, 2023.

[44] Bikash Sinha Sourav Sarkar, Helmut Satz. *The Physics of the Quark-Gluon Plasma*. Springer Berlin, Heidelberg, 2010.

[45] The ALICE Collaboration. The alice experiment at the cern lhc. *Journal of Instrumentation*, 3(08):S08002, aug 2008.

[46] Arturo Tauro. ALICE Schematics. General Photo, 2017.

[47] ALICE collaboration. Technical Design Report for the Upgrade of the ALICE Inner Tracking System. Technical report, CERN, 2014.

[48] J D Berst, G Claus, C Colledani, G Deptuch, W Dulinski, U Goerlach, Yu A Gornushkin, D Husson, Y Hu, J P Le Normand, G Orazi, J L Riester, R Turchetta, and M Winter. Monolithic active pixel sensors for high resolution vertex detectors. *CERN*, 2000.

[49] ALICE collaboration. Upgrade of the ALICE Time Projection Chamber. Technical report, CERN, 2013.

[50] ALICE collaboration. The ALICE Transition Radiation Detector: construction, operation, and performance. Technical report, CERN, 2018.

[51] ALICE collaboration. *ALICE Time-Of-Flight system (TOF): Technical Design Report*. Technical design report. ALICE. CERN, Geneva, 2000.

[52] Yi Wang and Yancheng Yu. Multigap resistive plate chambers for time of flight applications. *Applied Sciences*, 11(1), 2021.

[53] P. Cortese, G. Dellacasa, and the ALICE collaboration. ALICE Electromagnetic Calorimeter Technical Design Report. Technical report, CERN, 2008.

[54] V I Man'ko, W Klempt, L Leistam, J De Groot, and Jürgen Schükraft. *ALICE Photon Spectrometer (PHOS): Technical Design Report*. Technical design report. ALICE. CERN, Geneva, 1999.

[55] M Yu Bogolyubsky, A Di Mauro, S V Erine, C Gregory, F Hahn, S Haider, M S Ippolitov, Yu V Kharlov, A Klovning, O A Maeland, P Martinengo, N G Minaev, O H Odland, François Piuz, R Rongved, B Skaali, S A Sadovsky, V D Samoylenko, Yu Sibiriak, V I Suzdalev, V V Tikhonov, M Volkov, J Van Beelen, D Williams, A Zviagine, and M A Blick. Charged Particle Veto Detector for the PHOS Spectrometer. Technical report, CERN, Geneva, 1999.

[56] ALICE collaboration. Technical Design Report of the High Momentum Particle Identification Detector. Technical report, CERN, 1998.

[57] ALICE collaboration. Technical Design Report for the Muon Forward Tracker. Technical report, CERN, 2015.

[58] ALICE Collaboration. *ALICE dimuon forward spectrometer: addendum to the Technical Design Report*. Technical design report. ALICE. CERN, Geneva, 2000.

[59] The new alice fast interaction trigger. https://ep-news.web.cern.ch/content/new-alice-fast-interaction-trigger.

[60] D. Finogeev et al. Readout system of the ALICE Fast Interaction Trigger. Technical report, CERN, 2020.

[61] M Gallio, W Klempt, L Leistam, J De Groot, and Jürgen Schükraft. *ALICE Zero-Degree Calorimeter (ZDC): Technical Design Report*. Technical design report. ALICE. CERN, Geneva, 1999.

[62] S.C. Zugravel, P. Cortese, G. Cotto, N. De Marco, P. De Remigis, L. Lombardo, G. Mazza, C. Puggioni, E. Puleo, L. Quaglia, and M. Sitta. Alice zero degree calorimeters. the new readout system in lhc run 3. *Journal of Instrumentation*, 18(02):C02009, feb 2023.

[63] The ALICE Collaboration. Performance of the alice experiment at the cern lhc. *Journal of Instrumentation*, sep 2014.

[64] Lhc long term schedule. https://cds.cern.ch/record/2770529/plots.

[65] Daniele Mirarchi. Vol. 31 - Crystal Collimation for LHC. Crystal collimation for LHC. Technical report, CERN, 2015. Presented 18 Jun 2015.

[66] P Antonioli, A Kluge, and W Riegler. Upgrade of the ALICE Readout and amp; Trigger System. Technical report, CERN, 2013.

[67] Fabio Sauli. The gas electron multiplier (gem): Operating principles and applications. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 805:2–24, 2016. Special Issue in memory of Glenn F. Knoll.

[68] W J Willis. Applications of wire chambers in high-energy physics. *Nucl. Instrum. Methods*, 176:61–65, 1980.

[69] ALICE collaboration. The new alice computing farm for run 3. https://cerncourier.com/a/alice-ups-its-game-for-sustainable-computing/, August 2023.

[70] Fabienne Landua. ALICE detector LS2 upgrade in four languages. *CERN*, 2022. General Photo.

[71] Zdc collaboration. https://alice-collaboration.web.cern.ch/menu_proj_items/ZDC.

[72] Werner Herr. Features and implications of different LHC crossing schemes. Technical report, CERN, Geneva, 2003.

[73] 00 nim-camac & 01 coaxial connectors. https://web.lemo.com/img/resources/catalog/ROW/UK_English/00and01Coaxial_en.pdf.

[74] Model 740 quad linear fan-in/fan-out. https://groups.nscl.msu.edu/nscl_library/manuals/phillips/740.pdf.

[75] Caen. https://www.caen.it/download/.

[76] Ioxos website. https://www.ioxos.ch/.

[77] Vme bus technology. https://www.vita.com/VMEbus-FAQ.

[78] ifc1211 ioxos. https://www.ioxos.ch/produit/ifc_1211/.

[79] XILINX AMD. *UltraScale FPGA Product Tables and Product Selection Guide (XMP102)*, 02 2016. Rev. 1.7 English.

[80] Ti ads5409 adc. https://www.ti.com/product/ADS5409.

[81] Ssmb, ssmc, and scms specifications. https://coaxicom.com/specifications/ssmb-ssmc-and-scms-specifications/.

[82] Fm-s14 quad sfp/sfp+ transceiver fmc. https://www.fastertechnology.com/products/fmc/fm-s14.html.

[83] O. Bourrion, J. Bouvier, F. Costa, E. Dávid, J. Imrek, T.M. Nguyen, and S. Mukherjee. Versatile firmware for the common readout unit (cru) of the alice experiment at the lhc. *Journal of Instrumentation*, 16(05):P05019, may 2021.

[84] P. Chochula, L. Jirden, A. Augustinus, G. De Cataldo, C. Torcato, P. Rosinsky, L. Wallet, M. Boccioli, and L. Cardoso. The ALICE detector control system. *IEEE Trans. Nucl. Sci.*, 57:472–478, 2010.

[85] Mateusz Lechman, A Augustinus, P Bond, P Chochula, A Kurepin, O Pinazza, and P Rosinsky. Managing operational documentation in the ALICE Detector Control System. *J. Phys.: Conf. Ser.*, 396:052047, 2012.

[86] M. Barros Marin, S. Baron, S.S. Feger, P. Leitao, E.S. Lupu, C. Soos, P. Vichoudis, and K. Wyllie. The gbt-fpga core: features and challenges. *Journal of Instrumentation*, 10(03):C03021, mar 2015.

[87] CERN. *Proceedings of the Topical Workshop on Electronics for Particle Physics: Paris, France 21 - 25 Sep 2009. Topical Workshop on Electronics for Particle Physics*, Geneva, 2009. CERN.

[88] Jakub Kvapil, Anju Bhasin, Marek Bombara, David Evans, Anton Jusko, Alexander Kluge, Marian Krivda, Ivan Kralik, Roman Lietava, Sanket Kumar Nayak, Simone Ragoni, and Orlando Villalobos Baillie. Alice central trigger system for lhc run 3. *EPJ Web of Conferences*, 251:04022, 2021.

[89] O Bourrion, D Evans, J Imrek, A Jusko, A Kluge, M Krivda, J Kvapil, R Lietava, L A Pérez Moreno, O Villalobos-Baillie, and E Willsher. Interface between CTS-CRU and CTS-Detector Front Ends Trigger Notes for Developers. Technical report, CERN, Geneva, 2020.

[90] J Bovier, I Garcia, P Cristin, and J F Gillot. TOSCA 2B AP Specific ZDC Readout UG. Technical report, IOxOS, Gland, 2021.

[91] S Baron, J P Cachemiche, F Marin, P Moreira, and C Soos. Implementing the GBT data transmission protocol in FPGAs. *Proceedings of the Topical Workshop on Electronics for Particle Physics*, 2009.

[92] Vladislav Balagura. Van der meer scan luminosity measurement and beam–beam correction. *The European Physical Journal C*, 81(1):26, Jan 2021.

[93] D Jacquet, T Argyropoulos, E Bravin, A Calia, M Hostettler, E Métral, D Mirarchi, D Nisbet, T Persson, S Redaelli, M Solfaroli, G Trad, and J Wenninger. The LHC run 2022. *JACoW IPAC*, 2023:MOPL020, 2023.

[94] Edmond Ciapala, L Arnaudon, P Baudrenghien, O Brunner, A Butterworth, T Linnecar, P Maesen, J Molendijk, E Montesinos, D Valuch, and F Weierud. Commissioning of the 400 MHz LHC RF System. —, 2008.

[95] Alice photo repository. https://cds.cern.ch/record/2841494?ln=it.

[96] Run 3 pbpb alice integrated luminosity. https://home.cern/news/news/experiments/alice-bags-about-twelve-billion-heavy-ion-collisions.

[97] R et Al Bruce. 2023 Pb-Pb background observations and mitigations, and plans for future studies. —, —:9, 2023.

[98] Backgroun studies zdc news. https://home.cern/news/news/accelerators/accelerator-report-optimisation-greater-success-and-new-challenges.

[99] C Wiesner, W Bartmann, C Bracco, E Carlier, M Fraser, B Goddard, D Jacquet, N Magnin, and J Wenninger. Abort Gap Keeper and Abort Gap Protection. *CERN*, pages 141–145, 2019.

[100] A. Di Salvo, S. Garbolino, M. Mignone, S. C. Zugravel, A. Rivetti, A. Palmieri, and M. E. Bertaina. A 64-channel asic for full waveform sampling with 200 ms/s for space-based cosmic-particles applications. *Nuovo cimento della Società italiana di fisica. C. Geophysics and space physics*, 2024.

[101] Burmistrov, Leonid. Terzina on board nuses: A pathfinder for eas cherenkov light detection from space. *EPJ Web Conf.*, 283:06006, 2023.

[102] Johannes Eser, Angela V. Olinto, and Lawrence Wiencke. Overview and First Results of EUSO-SPB2. *PoS*, ICRC2023:397, 2023.

[103] George Filippatos. EUSO-SPB2 Fluorescence Telescope in-flight performance and preliminary results. *PoS*, ICRC2023:251, 2023.

[104] Tobias Heibges. Overview of the EUSO-SPB2 Target of Opportunity program using the Cherenkov Telescope. *PoS*, ICRC2023:1134, 2023.

[105] I. De Mitri, M. Di Santo, and on behalf of the NUSES collaboration. The nuses space mission. *Journal of Physics: Conference Series*, 2429(1):012007, feb 2023.

[106] POEMMA collaboration. The poemma (probe of extreme multi-messenger astrophysics) observatory. *Journal of Cosmology and Astroparticle Physics*, 2021(06):007, jun 2021.

[107] Angela V. Olinto. Poemma (probe of extreme multi-messenger astrophysics) roadmap update, 2023.

[108] A. L. Cummings, R. Aloisio, and J. F. Krizmanic. Modeling of the tau and muon neutrino-induced optical cherenkov signals from upward-moving extensive air showers. *Physical Review D*, 103(4), February 2021.

[109] A. L. Cummings, R. Aloisio, J. Eser, and J. F. Krizmanic. Modeling the optical cherenkov signals by cosmic ray extensive air showers directly observed from suborbital and orbital altitudes. *Physical Review D*, 104(6), September 2021.

[110] NUSES collaboraiton. The Zire experiment on board the NUSES space mission. *PoS*, ICRC2023:139, 2023.

[111] Alberto Gola, Fabio Acerbi, Massimo Capasso, Marco Marcante, Alberto Mazzi, Giovanni Paternoster, Claudio Piemonte, Veronica Regazzoni, and Nicola Zorzi. Nuv-sensitive silicon photomultiplier technologies developed at fondazione bruno kessler. *Sensors*, 19(2), 2019.

[112] Mario E. Bertaina, Pietro Antonio Palmieri, Andrea Di Salvo, Sara Garbolino, Raffaele Aaron Giampaolo, Marco Mignone, Hiroko Miyamoto, Angelo Rivetti, Silvia Tedesco, and Stefan Zugravel. A new front end electronics for the detection of the optical Cherenkov signals by Extensive Air Showers directly observed from sub-orbital and orbital altitudes. *PoS*, ICRC2023:311, 2023.

[113] XILINX AMD. *LogiCORE IP SelectIO Interface Wizard v3.1 Data Sheet (DS746)*, 02 2011. Rev. 2.0.

[114] XILINX AMD. *SelectIO Interface Wizard Product Guide (PG070)*, 10 2016. Rev. 5.1 English.

[115] XILINX AMD. *Vivado Design Suite 7 Series FPGA and Zynq 7000 SoC Libraries Guide (UG953)*, 05 2024. Rev. 2024.1 English.

[116] XILINX AMD. *Divider Generator v5.1 Product Guide (PG151)*, 02 2021. Rev. 5.1 English.

[117] H. T. Johansson, A. Furufors, and P. Klenze. Fakernet – small and fast fpga-based tcp and udp communication, 2020.

[118] Waylon Jepsen. Cyclic redundancy checks and error detection, 2022.

[119] XILINX AMD. *KC705 Evaluation Board for the Kintex-7 FPGA User Guide (UG810)*, 02 2019. Rev. 1.9 English.

[120] DIGILENT. *Genesys 2 FPGA Board Reference Manual (410-300)*, 08 2017. Rev. Genesys 2 F.

[121] XILINX AMD. *7 Series Product Tables and Product Selection Guide (XMP101)*, 04 2021. Rev. 1.8 English.

[122] Angela Orebaugh, Gilbert Ramirez, Jay Beale, and Joshua Wright. *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress Publishing, 2007.

[123] Rick Bitter, Taqi Mohiuddin, and Matt Nawrocki. *LabVIEW: Advanced programming techniques*. Crc Press, 2006.

[124] Automate saving and reporting data with labview. https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x1k3CAA&l=en-GB.

Chapter 7

ZDC parameter configuration

BOARD N ->	0	1	2	3	4	5	6	7
TH LVL CH0	0xa	0xa	0xa	0xa	0xb	0xb	0xb	0xb
TH LVL CH1	0xc	0xc	0xc	0xc	0xb	0xd	0xb	0xd
TH LVL CH2	0xa	0xa	0xa	0xa	0xb	0xb	0xb	0xb
TH LVL CH3	0xa	0xa	0xa	0xa	0xb	0xb	0xb	0xb
DIF DL CH0	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF DL CH1	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF DL CH2	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF DL CH3	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF MS CH0	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
DIF MS CH1	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
DIF MS CH2	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
DIF MS CH3	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
ALICE MSK	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10
AUTO T MSK	0xc	0xc	0xc	0xc	0xc	0xc	0xc	0xc
READ MSK	0xb	0x7	0xb	0x7	0xf	0x7	0xf	0x7
BASELI DIV	0xa	0xa	0xa	0xa	0xa	0xa	0xa	0xa
TRIGG COND	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
TRIGG SMPL	0x6	0x6	0x6	0x6	0x6	0x6	0x6	0x6
DL CS DFLT	0xb	0xb	0xb	0xb	0xb	0xb	0xb	0xb
FRST BC CL	0x129	0x129	0x129	0x129	0x129	0x129	0x129	0x129
AUTO RES S	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
AVG CL MSk	0xf	0xf	0xf	0xf	0xf	0xf	0xf	0xf
AVG CL CFG	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
EN DS FINE	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
DBG SG CFG	0x6	0x6	0x6	0x6	0x6	0x6	0x6	0x6
EN CAL TRG	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1

Table 7.1 Parameter configuration for Pb-Pb data taking. DL SMP and DL CRS are omitted since they change at each LHC fill. BASELI DIV is automatically written by the DCS in relation to the filling scheme in the machine.

BOARD N ->	0	1	2	3	4	5	6	7
TH LVL CH0	0xb	0xb	0xb	0xb	0xb	0xb	0xb	0xb
TH LVL CH1	0xd	0xd	0xd	0xd	0xb	0xd	0xb	0xd
TH LVL CH2	0xb	0xb	0xb	0xb	0xb	0xb	0xb	0xb
TH LVL CH3	0xb	0xb	0xb	0xb	0xb	0xb	0xb	0xb
DIF DL CH0	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF DL CH1	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF DL CH2	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF DL CH3	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
DIF MS CH0	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
DIF MS CH1	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
DIF MS CH2	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
DIF MS CH3	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x1fff
ALICE MSK	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10
AUTO T MSK	0xc	0xc	0xc	0xc	0xc	0xc	0xc	0xc
READ MSK	0xb	0x7	0xb	0x7	0xf	0x7	0xf	0x7
BASELI DIV	0xa	0xa	0xa	0xa	0xa	0xa	0xa	0xa
TRIGG COND	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
TRIGG SMPL	0x6	0x6	0x6	0x6	0x6	0x6	0x6	0x6
DL CS DFLT	0xc	0xc	0xc	0xc	0xc	0xc	0xc	0xc
FRST BC CL	0x129	0x129	0x129	0x129	0x129	0x129	0x129	0x129
AUTO RES S	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
AVG CL MSk	0xf	0xf	0xf	0xf	0xf	0xf	0xf	0xf
AVG CL CFG	0x4	0x4	0x4	0x4	0x4	0x4	0x4	0x4
EN DS FINE	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
DBG SG CFG	0x6	0x6	0x6	0x6	0x6	0x6	0x6	0x6
EN CAL TRG	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1

Table 7.2 Parameter configuration for pp data taking. DL SMP and DL CRS are omitted since they change at each LHC fill. BASELI DIV is automatically written by the DCS in relation to the filling scheme in the machine.