Development of DAQ system for theacquisition of neutrino bursts from corecollapse supernovae in a DUNE far detector

Quentin David

N°d'ordre NNT : 2022LYSE1054

THESE de DOCTORAT DE L'UNIVERSITE DE LYON

opérée au sein de
l'Université Claude Bernard Lyon 1

Ecole Doctorale N° 52
Physique et Astrophysique de Lyon

Discipline : Physique

Soutenue publiquement le 05/05/2022, par :
Quentin DAVID

Development of DAQ system for theacquisition of neutrino bursts from corecollapse supernovae in a DUNE far detector

---

Devant le jury composé de :

Lu, Guo-Neng	Professeur des Universités	Université Lyon 1	Président
Lehmann Miotto, Giovanna	Physicienne	CERN, Genève	Rapporteure
Russo, Stefano	Ingénieur de Recherche-HDR	CNRS Paris	Rapporteur
Gallin-Martel, Marie-Laure	Chargée de Recherche-HDR	CNRS Grenoble	Examinatrice
Autiero, Dario	Directeur de Recherche	CNRS Lyon	Directeur de thèse
Galymov, Vyacheslav	Chargé de Recherche	CNRS Lyon	Co-directeur de thèse

Development of DAQ system for the acquisition of neutrino bursts from core collapse supernovae in a DUNE far detector

Author:
Quentin David

Supervisors:

**** Ph.D. Ecole Doctorale PHAST*

February 8, 2022

Acknowledgements

Table of Contents

Glossary

ACK : Acknowledgement

ADC : Analog to Digital Converter

ALM : Adaptive Logic Module

AMC : Advanced Mezzanine Card

AMM : Avalon Memory Mapped

APA : Anode Plane Assemblies

ARP : Address Resolution Protocol

ASIC : Application-Specific Integrated Circuit

AST : Avalon STreaming

AT : Above Threshold

ArgoNeuT : Argon Neutrino Teststand

BDE : Bottom Drift Electronics

CASTOR : CERN Advanced STORage

CC : Charged Current

CERN : European Organization for Nuclear Research

CLB : Configurable Logic Block

CMM : Context Memory Management

CMOS : Complementary Metal Oxide Semiconductor

CP : Charge Parity

CPA : Cathode Plane Assemblies

CRO : Charge ReadOut

CRP : Charge Readout Plane

DAQ : Data AcQuisition

DDR : Double Data Rate

DIO : Digital Input/Output

DMA : Direct Memory Access

DONuT : Direct Observation of Nu Tau

DP : Dual Phase

DPRAM : Dual Port Random Access Memory

DUNE : Deep Underground Neutrino Experiment

EHN1 : Experimentation Hall North 1

EMI : ElectroMagnetic Interference

EOP : End Of Packet

EOS : EOS Open Storage

ES : Elastic Scattering

ESD : Electro-Static Discharge

FD : Far Detector

FE : Front-End

FIFO : First-In First-Out

FNAL : Fermi National Accelerator Laboratory

FPGA : Field-Programmable Gate Array

Fermilab : Fermi national accelerator laboratory

GKVM : Gava-Kneller-Volpe-McLaughlin

GPS : Global Positioning System

GPU : Graphics Processing Unit

GbE : Gigabit Ethernet

HDL : Hardware Description Language

HV : High Voltage

ICARUS : Imaging Cosmic And Rare Underground Signals

IO : Inverse Ordering

IP : Internet Protocol

IP (FPGA) : Intellectual Property

IP2I Lyon : Institut de Physique des 2 Infinis Lyon

IT : Information Technology

JTAG : Joint Test Action Group

K2K : KEK to Kamioka

LARZIC : Liquid ARgon Zygote Integrated Circuit

LAr : Liquid Argon

LAr TPC : Liquid Argon Time Projection Chamber

LArIAT : Liquid Argon In A Testbeam

LBNF : Long Baseline Neutrino Facility

LEM : Large Electron Multiplier

LSB : Less Significant Bit

LUT : Look-Up Table

MAC : Media Access Controler

MARLEY : Model of Argon Reaction Low Energy Yields

MCH : MicroTCA Carrier Hub

MINOS : Main Injector Neutrino Oscillation Search

MM : Memory Mapped

MSB : Most Significant Bit

MSW : Mikheev - Smirnov - Wolfenstein

MicroBoone : Micro Booster Neutrino Experiment

NAT : Gesellschaft für Netzwerk- und Automatisierungs-Technologie mit beschränkter Haftung

NC : Neutral Current

NO : Normal Ordering

NOvA : NuMI Off-Axis ν_eAppearance

NP02 : Neutrino Plateform 02

NuMI : Neutrinos at the Main Injector

OSI : Open Systems Interconnection

PCB : Printed Circuit Board

PCIe : Peripheral Component Interconnect Express

PD : Photon detector

PD System : Photon Detection System

PEP : Positron-Electron Project

PFM : Primitive Finder Module

PFM10G : Primitive Finder Module 10G

PHY : Physical

PIP II : Proton Improvement Plan II

PMNS : Pontecorvo-Maki-Nakagawa-Sakata

PMT : Photo-Multiplier Tubes

POE : Primitive Offload Engine

PPS : Pulse Per Second

PRM : Prim Reconstruct Module

ProtoDUNE : Prototype DUNE

PTP : Universal Time-Sharing System

R&D : Research and Development

RAM : Random Access Memory

RISC : Reduced instruction Set Computer

RMS : Root Mean Square

ROI : Region Of Interest

SFT : Signal FeedThrough

SI : International System of units

SM : State Machine

SNB : Supernova Neutrino Burst

SNO : Sudbury Neutrino Observatory

SOP : Start Of Packet

SP : Single Phase

SRAM : Static Random Access Memory

SPI : Serial Peripheral Interface

ST : STreaming

SURF : Sanford Underground Research Facility

T2K : Tokai to Kamioka

TAI : International Atomic Time

TDE : Top Drift Electronics

TDR : Technical Design Report

TP : Trigger Primitives

TPC : Time Projection Chamber

UART : Universal Asynchronous Receiver Transmitter

UDP : User Datagram Protocol

UTS : Universal Time-Sharing System

VD : Vertical Drift

VHDCI : Very-High-Density Cable Interconnect

WR : White Rabbit

WRLEN : White Rabbit End-Node

linac : Linear accelerator

ppb : parts per billion

ppm : parts per million

μTCA: Micro Telecommunications Computing Architecture

Chapter 1

Introduction

This thesis work was performed within the framework of the charge readout electronics and DAQ developments for the DUNE dual-phase far detector module, as described in the DUNE dual-phase TDR volume. The charge readout electronics for the dual-phase (and since 2021 for the DUNE Vertical Drift second far detector module) is the outcome of a long R&D process carried on by the neutrino group at IP2I Lyon since 2006.

These developments included the cryogenic analog amplifiers, the digitization, timing and DAQ systems. A full deployment of the charge readout system developed at IP2I was performed on DUNE prototypes operating at CERN: first on the $3 \times 3 \times 1m^3$ prototype (2016-2017) and then on the $6 \times 6 \times 6m^3$ detector (NP02/protoDUNE dual-phase experiment, 2018-2020). The digitization system based on AMC cards operating in μTCA crates was designed in order to have different operation modes, based on external triggers for the prototypes, or in continuous data streaming in the case of the DUNE far detector module. The CERN prototypes are operating on surface, exposed to a large flux of cosmic rays. The main goal of their operation is to assess the overall detector performance and the one of the various sub-systems. These detectors can be externally triggered in different ways depending whether they are collecting cosmic ray events or charged particles beam events.

The DUNE far detector modules will operate underground at 1500 m depth with a cosmic ray flux of the order of a single muon every 11 seconds per detector module. Most of the recorded data volume will be related to noise or the intrinsic radioactivity of LAr or of the detector components. These detectors will have to be continuously sensitive, not only to the events related to the neutrino beam produced at 1300 km distance at Fermilab which happen with a very precise timing, but also to non-accelerator physics events such as atmospheric neutrinos, possible proton decay candidates and neutrino bursts from supernova core collapse.

This last physics item is particularly challenging from the point of view of the data acquisition system. The rate of supernova core collapse in the Milky Way is of the order of a few events per century. The first requirement for a detector like DUNE is to have a DAQ system with a very high livetime in order not to miss one of these rare events. The second challenge comes from the fact that the burst duration lasts several tens of seconds. Data must be continuously buffered during this long time in order to preserve them and be in the condition of making a decision of writing this very large amount of data persistently on disk. The other requirement is not to have zero skipping at all on the DUNE data but to record them in their complete integrity. The DAQ system must then collect continuously charge readout data from the front-end digitization units and analyse them on real time in order to understand if they are worth to be saved on disk since they contain a neutrino interaction from the beam, a cosmic ray, radioactivity event or a supernova neutrino burst.

These requirements make the DAQ back-end system very different than the ones used for the protoDUNEs at CERN. In some sense the continuous data streaming mode (at a sampling rate around 2.5 MHz) for all readout channels is a simplification with respect to the external trigger mode operated in the protoDUNEs. However when considering that in a μTCA crate data are produced by 10 AMC cards, each one with 10Gb/s connectivity to the μTCA crate backplane. For a total of 640 channels one quickly realizes that transmitting without losses these data would require a bandwidth of 20Gb/s per crate. The original connectivity of the μTCA crates is based on MCH modules with 10 Gb/s optical links. This is not a problem for the operation of the protoDUNE dual-phase, since it just limits the maximal trigger rate to 70Hz when transmitting drift windows of 4ms sampled at 2.5 MHz, however it is a clear bottleneck for the data volume needed when operating the DUNE far detector.

Several solutions were investigated in this respect such as: implementing on the AMC digitization cards an online lossless data compression allowing to limit the used bandwidth to less than 10Gb/s or staying with uncompressed data but augmenting the MCH connectivity to 40Gb/s, as it became recently possible and was successfully tested. In any case, the back-end nodes which collect the data from the optical links connected to the μTCA crates will have to perform an online analysis of these data in order to find out if the data contain any interesting energy deposition in the detector and transmit this essential information to a trigger coordination unit which will make the decision of storing these data, which meanwhile are buffered in the back-end DAQ nodes. In addition, if the data arrive compressed to the back-end node, they must be decompressed on real time in order to perform their analysis. This analysis task which is called "Trigger Primitives Search" is very challenging since it implies dealing with a very large data volume and on real time, without introducing any delay. This development was the main subject of this thesis work which evaluated a few possible solutions. Architectures based on FPGA based network cards which would constitute the interface of the back-end nodes with the μTCA crates and operate on real time the Trigger Primitives Search as well as possibly data decompression. Both these tasks were implemented by using the computing resources of the FPGAs on the network cards.

This document will then go further in the description of the problems relative to the detection of neutrinos, in particular to those from rare events like Supernova in the MilkyWay. Getting this valuable data will be a difficult challenge, that is to make the best of the small things that can be detected.

Chapter 2 will describe and explain the behavior of neutrinos through, in particular, a short history of discoveries on the subject. Moreover, since they will be a major preoccupation in the rest of this thesis, neutrinos coming from Supernova will be detailed.

Chapter 3 will then describe the future DUNE experiment. The different elements of the infrastructure such as the accelerator and the detection modules will be detailed. Information on the types and form of Supernova events expected in the experiment will be given. The operating principle and the means deployed in the Single Phase and Dual Phase detection technologies will be explained. Finally, the operation of the prototype detectors placed at CERN allowing the development and the validation of the whole experiment will be described.

Chapter 4 will give an overview of all the complex electronics and acquisition chain that must be put in place to store the interesting physics event data from the protoDUNE DP detector. For this, the different systems will be described such as the cryogenic front-end electronics, the back-end electronics and the data synchronization system. Finally, will be discussed the various evolutions of the system necessary to the good functioning in order to adapt it to the remote module of DUNE equipped with the DP technology which will be 40 times bigger than the prototype.

Chapter 5 will detail the first study carried out with the aim of reducing the data flow coming from the detector. This is intended to evaluate a lossless compression of the Huffman type adapted to the application. Its operation as well as the means implemented will be stated. Following this compression,

it was necessary to verify that the decompression of these data was feasible with the constraints posed by the experiment. The evaluation of a real time decompression is therefore described in this part. Finally, some results of compressed data transfers obtained in the framework of protoDUNE DP are discussed.

Chapter 6 will describe the second study realized during this thesis. It concerns a real time search of trigger primitives which aims to select if interesting events have occurred in the detector or not. After the study of the various constraints, it was chosen to carry out this study on FPGA boards, the operation of some basic design blocks are therefore detailed in a first step. Then, a first architecture was implemented by working with compressed data, its operation as well as its performances are exposed. Following these results, a second architecture has been realized, this time on a decompressed data stream whose behavior and results will also be discussed.

Chapter 2

Neutrino physics

2.1 Neutrinos

Neutrinos are elementary spin 1/2 particles. Along with the three charged leptons — $e^{\pm}$, $\mu^{\pm}$, and $\tau^{\pm}$ — they form the lepton family, the group of elementary particles that do not participate in the strong interactions. Since neutrinos are also electrically neutral, they can only interact with matter via weak force whose carriers are $W^{\pm}$ and Z^0 bosons. The interactions involving the exchange of $W^{\pm}$ are typically referred to as charge-current (CC), while those with Z^0 are called neutral-current (NC) reactions. The simplest diagrams representing the processes of a CC and an NC interaction of a neutrino with a nucleon are shown in Fig 2.1.

The key feature of the CC reactions is that the type of the outgoing lepton is given by the flavour of the incoming neutrino as illustrated by Fig 2.1. Thus identifying lepton in the final state allows to experimentally determine the flavour of the interacting neutrino.

Neutrinos (electron or anti-electron) are naturally produced in β decays of radioactive isotopes. In fact, the first hint of their existence came from the measurements of the beta electron energy spectrum. In 1914 James Chadwick using a magnetic spectrometer and a Geiger counter observed a continuous distribution of the β electron energies [1], while the conservation of energy applied to a two-body decay systems predicted a discreet energy for electrons. This result was completely unexpected and generated a considerable discussions at the time. The difficulty explaining this anomaly led Niels Bohr to voice a radical idea that the energy conservation principle could be violated for a given decay.

A solution to the β decay puzzle was offered by Wolfgang Pauli, who in 1930 proposed the existence of a new particle emitted along with the electron that he called "neutron" (English translation of his

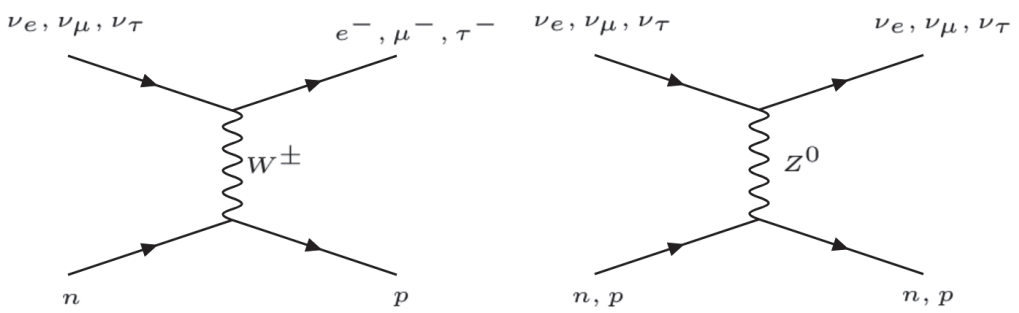


FIGURE 2.1: Charge Current (left) and Neutral Current (right) diagrams for neutrino scattering from a nucleon.

letter is given for example in [2]). The energy could thus be shared between the electron and "neutron" explaining the observed broadness of the energy distribution for the former. This "neutron" had to be an electrically neutral particle with a great penetrating power to account for the lack of any previous experimental detection. The actual neutron was discovered by Chadwick in 1932 [3]. Later on he and Maurice Goldhaber also succeeded to measure its mass to be close to that of the proton [4]. The neutron was much too heavy to be the invisible partner of the β decay electrons.

Continuing with Pauli's hypothesis of the new particle, Enrico Fermi formulated the quantitative description of β decay with the invisible particle now named "neutrino" distinguishing it from Chadwick's neutron. Fermi's β decay theory, the precursor of the electroweak theory, successfully explained the continuous β decay spectrum by postulating a creation of an electron and (anti)-neutrino pair with the later assumed to be a massless neutral particle. The Fermi theory also predicted the existence of the processes of type:

$$\bar{\nu}_e + p \rightarrow n + e^+. \tag{2.1}$$

The reaction in Eq. 2.1, often referred to as inverse β decay, offers one possible experimental way for the neutrino detection. The first estimations of the neutrino capture rate cross section via this process was performed by H. Bethe and R. Peierls [5], who found it to be so small (the penetrating power of a 2.3 MeV neutrino was estimated to be "10^{16} km in solid matter") that they concluded that "there is no practically possible way of observing the neutrino". However, they were proven wrong. In 1956, C. Cowan and F. Reines announced the detection of electron anti-neutrinos [6] from the β decays of fission fragments in the Savannah River nuclear power plant. Using about 400 L of water with dissolved cadmium chloride, the duo were able to isolate neutrino events by detecting in delayed coincidence e^+ and n, the two final state products in Eq 2.1. The positrons annihilating with the surrounding material would give a prompt signal from two resultant photons, while neutrons would slow down until they are captured on the cadmium nuclei several microseconds late resulting in a subsequent delayed emission of a γ ray.

One should note that while now it is known neutrinos come in three flavours — ν_e, ν_μ, and ν_τ — and these flavours as well as their corresponding anti-partners all represent distinct particles, at the time of Cowan and Reines experiment this was far from the case. In fact, in 1954, Ray Davis attempted to observe the reactor $\bar{\nu}_e$ using the chlorine capture mechanism proposed by Bruno Pontecorvo [7]:

$$\nu_e + {}^{37}\mathrm{Cl} \rightarrow e^- + {}^{37}\mathrm{Ar}. \tag{2.2}$$

In this process the number of interacting neutrinos would be subsequently deduced by counting the number of ^{37}Ar isotopes. Davis completed the experiment at a nuclear reactor at Brookhaven, but did not detect any neutrino induced events [8] and concluded that neutrinos did not interact in the same way as anti-neutrinos.

Around 1960, the idea to use particle accelerators to generate high energy neutrino beams was proposed independently by Pontecorvo [9] and M. Schwartz [10]. The 1947 discovery of charged pion mesons and the fact that they decayed predominantly into a muon and a neutrino

$$\pi^+ \rightarrow \mu^+ + \nu_\mu, \tag{2.3}$$

suggested the possibility of generating an intense and energetic neutrino beam from pions that can be easily produced by colliding an accelerated proton beam with a stationary target. A few years after the proposal, Schwartz and a group of physicists at Brookhaven National Lab succeed in building the first accelerator neutrino beam experiment [11]. In the hypothesis that neutrinos produced in β decays and those emitted from $\pi^\pm$ disintegration were the same particle, one would have expected as many electron as muon tracks resulting from neutrinos undergoing CC interactions. Only muon tracks were observed leading to the conclusions there are at least two types or "flavours" of neutrino ν_e and ν_μ. The detection of the τ charged lepton by Martin Perl *et al.* at Stanford Linear Accelerator Center

in 1975 [12] suggested that there also exits a third neutrino flavour, ν_τ. In 2000 DONuT (Direct Observation of NU Tau) experiment at Fermilab announced the results of the first direct detection of ν_τ in the CC interactions.

Davis continued to pursue the neutrino detection using the chlorine reaction in Eq. 2.2 focusing his attention on neutrinos emitted by the Sun (solar neutrinos) [13]. In 1968, he and his collaborators published [14] a first measurement of the solar neutrino flux, which, however, was much lower than the theoretical expectations from modelling of the reactions within Sun [15]. This discrepancy, which became known as the "solar neutrino problem", turned out to be quite important.

Another intriguing hint about the behaviour of neutrinos came in 1998 by the scientists working at the Super-Kamiokande neutrino observatory in Kamioka, Japan [16]. By studying atmospheric neutrinos, which are produced in cosmic ray collisions with the Earth's atmosphere, they observed a large deficit of upward-going muon neutrinos compared to the down-ward going ones. As the neutrinos are weakly interacting particles they can travel through the Earth with very little absorption. It was, therefore, expected that there should not be much difference in the distribution of the detected neutrinos as a function of direction. As there was no excess observed for electron neutrinos, the best hypothesis that could explain the discrepancy is that the muon neutrinos travelling through Earth have converted or "oscillated" to the tau neutrinos. Since the latter could not be directly observed by Super-Kamiokande, the upward-going muons seemed to have "disappeared".

The Super-Kamiokande result was the first measurement of the neutrino oscillations, which was later tested by dedicated accelerator neutrino beam experiments such as K2K [17] and MINOS [18] and measured with high accuracy more recently by T2K [19] and NOvA [20].

The solar neutrino problem was resolved by SNO experiment [21], which was able to measure the total solar neutrino flux independent of flavour demonstrating that the deficit observed by Davis was the result of the neutrinos changing flavour. Measurements conducted using nuclear reactors as neutrino sources [22] also showed that neutrinos oscillate between flavours.

2.2 Neutrino oscillations

The quantum mechanical phenomenon of the neutrino oscillations arises from the fact that neutrino flavour states $\nu_{\alpha=e,\mu,\tau}$, which take part in the weak interactions do not possess a well-defined mass. Rather they can be written as a linear superposition of the mass (eigen)states $\nu_{i=1,2,3}$:

$$\nu_\alpha = \sum_{i=1}^{3} U^*_{\alpha i} \nu_i. \tag{2.4}$$

The unitary mixing matrix U relates mass states to flavour states. It is called the PMNS mixing matrix after Pontecorvo, Maki, Nakagawa, and Sakahata, the four physicists who worked on advancing the ideas of the neutrino mixing.

For a neutrino is produced in flavour state α at $t = 0$ and $x = 0$ and propagating in vacuum, its state at the position L from the source and time t is

$$\psi(L,t) = \sum_{i=1}^{3} U^*_{\alpha i} e^{-i(E_i t - p_i L)} \nu_i, \tag{2.5}$$

where E_i and p_i are energy and momentum of the i^{th} mass eigenstates.

To find the probability, $P_{\alpha\beta}$, of detecting a neutrino in a flavour state ν_β one has find the projection of the state ψ on the state ν_β (probability amplitude), whose modulus squared is then $P_{\alpha\beta}$:

$$P_{\alpha\beta} = |\langle\nu_\beta|\psi(L,t)\rangle|^2 = \sum_{i,j} U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j} e^{-i(E_i t - p_i L)} e^{i(E_j t - p_j L)}, \tag{2.6}$$

where one can use orthogonality of the mass states $\langle\nu_j|\nu_i\rangle = \delta_{ij}$.

Since in practice neutrinos are very relativistic $E_i \simeq E_j \equiv E$, the momentum of the i^{th} mass state can be expressed as

$$p_i = \sqrt{E^2 - m_i^2} = E - \frac{m_i^2}{2E}. \tag{2.7}$$

Combining Eq. 2.6 and Eq. 2.7 one can write $P_{\alpha\beta}$ as

$$P_{\alpha\beta} = \sum_{i,j} U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j} \exp\left(-i\frac{\Delta m^2_{ji} L}{2E}\right), \tag{2.8}$$

where Δm^2_{ji} is the *mass-squared difference* defined as

$$\Delta m^2_{ji} = m^2_j - m^2_i \tag{2.9}$$

To illustrate the key features of Eq. 2.8, one can take a simple model of two neutrino flavours ν_e and ν_μ. The mixing matrix is then just a 2×2 rotational matrix defined by a mixing angle θ:

$$U_2 = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{2.10}$$

There is also only one mass-squared difference to consider: $\Delta m^2 = m^2_2 - m^2_1$.

The probability to detect ν_e appearance in a beam of ν_μ from Eq. 2.8 can then be reduced to

$$P_{\mu e} = \sin^2 2\theta \sin^2\left(\Delta m^2 \frac{L}{4E}\right), \tag{2.11}$$

which is periodic as a function of L/E with the oscillation frequency set by Δm^2 and the amplitude by the mixing angle θ.

So far, all the expressions have been written in the *natural* units, where the reduced Plank's constant $\hbar$ and the vacuum speed of light c are 1. By putting back their SI values one obtains:

$$\Delta m^2 \frac{L}{4E} \equiv 1.27 \Delta m^2(\mathrm{eV}^2) \frac{L(\mathrm{km})}{E(\mathrm{GeV})}. \tag{2.12}$$

The *appearance* probability $P_{\mu e}$ and the complimentary survival or *disappearance* probability $P_{\mu\mu}$ ($P_{\mu\mu} = 1 - P_{\mu e}$) are illustrated in Fig. 2.2 for a mono-energetic neutrino beam of 2 GeV, $\Delta m^2_{21} = 2.5 \times 10^{-3}$ eV^2, and $\theta = \pi/4$, which represents the maximal mixing between two flavour states in this model. As can be seen in the figure, at the distance of about 1000 km from the source — first oscillation maximum (or minimum) for the chosen energy where the expression in Eq. 2.12 is equal to $\pi/2$ — all ν_μ would have oscillated into ν_e and then converted back into ν_μ as they travel further on continuing the oscillatory pattern. While the source of the neutrino oscillations lies in the quantum mechanics of the interference between quantum states, it manifests itself on a remarkably macroscopic scale.

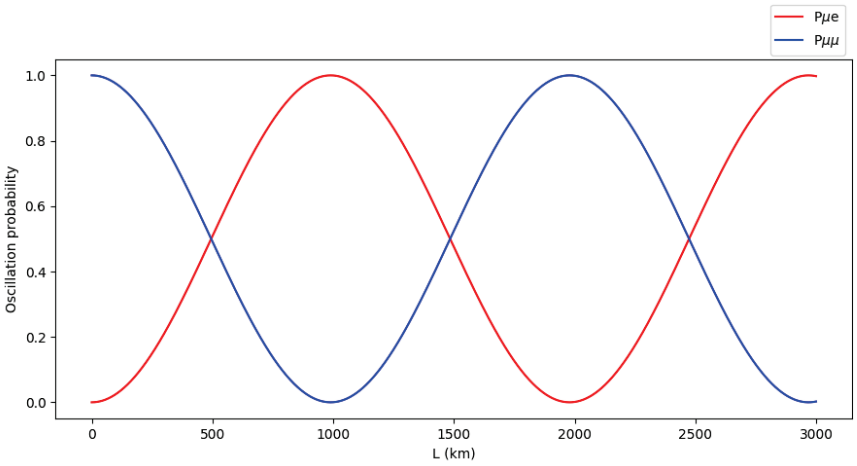


FIGURE 2.2: Illustration of the neutrino oscillation phenomenon for the simple model of two-neutrino mixing. The oscillation probabilities are computed for $\Delta m^2_{21} = 2.5 \times 10^{-3}$ eV^2, $\theta = \pi/4$, and 2 GeV neutrino energy.

One can write out the mixing formula in Eq. 2.4 in the matrix form:

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \tag{2.13}$$

The rows of the PMNS matrix show which linear combination of mass eigenstates constitute a given flavour ν_α. Similarly, the columns of give the flavour composition for a given mass state ν_i.

The PMNS matrix is parametrized in terms of three mixing angles θ_{12}, θ_{23}, and θ_{13} as well as complex phase δ_{CP}:

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta_{CP}} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.14}$$

where $c_{ij} \equiv \cos\theta_{ij}$ and $s_{ij} \equiv \sin\theta_{ij}$. These four parameters and two mass-squared differences Δm^2_{21} and Δm^2_{32} (or Δm^2_{31}) describe the oscillation patterns of the three neutrino flavours. In terms of the neutrino mass hierarchy, it is known that the ν_2 is heavier than ν_1, *i.e.*, $\Delta m^2_{21} > 0$. However, it is unknown whether ν_3 is the heaviest or the lightest neutrino. Fig. 2.3 illustrates the two possible neutrino mass hierarchies, referred to as Normal Ordering (NO) and Inverted Ordering (IO), for the three neutrino mass states. The approximate flavour composition of each ν_i state, given by $|U_{\alpha i}|^2$, is also shown. One should point out that the relative dimensions between the mass-squared differences shown in the figure are not to scale. The value Δm^2_{21} is about 7.4×10^{-5} eV^2, while that of Δm^2_{32} (or $|\Delta m^2_{31}|$ for IO) is more than 30 times larger and is around 2.5×10^{-3} eV^2 [23, 24]. This often allows to reduce the 3-flavour formalism to a simpler 2-flavour case if one of the Δm^2_{ij} determines the dominant frequency for the oscillations. Historically, the Δm^2_{21} mass-squared difference has been referred as solar (Δm^2_{sol}) due to its connection to the studies of solar neutrino oscillations, while the larger Δm^2_{32} as atmospheric (Δm^2_{atm}) as it was first probed by the Super-Kamiokande measurement of the oscillations in the flux of the atmospheric neutrinos.

While all three mixing angles having been measured now to a varying degree of precision, the value of δ_{CP} is yet unknown. If this phase is not equal to 0° or 180°, the oscillation probabilities for neutrino and anti-neutrinos can be different and thus the neutrino sector can be another source of the violation of the Charge Parity (CP) symmetry. The CP symmetry breaking is one of the key elements (Sakharov conditions) for generating the matter-dominant universe. While CP violation has been measured in the quark sector, it is too small to allow generating the required matter-antimatter asymmetry. The

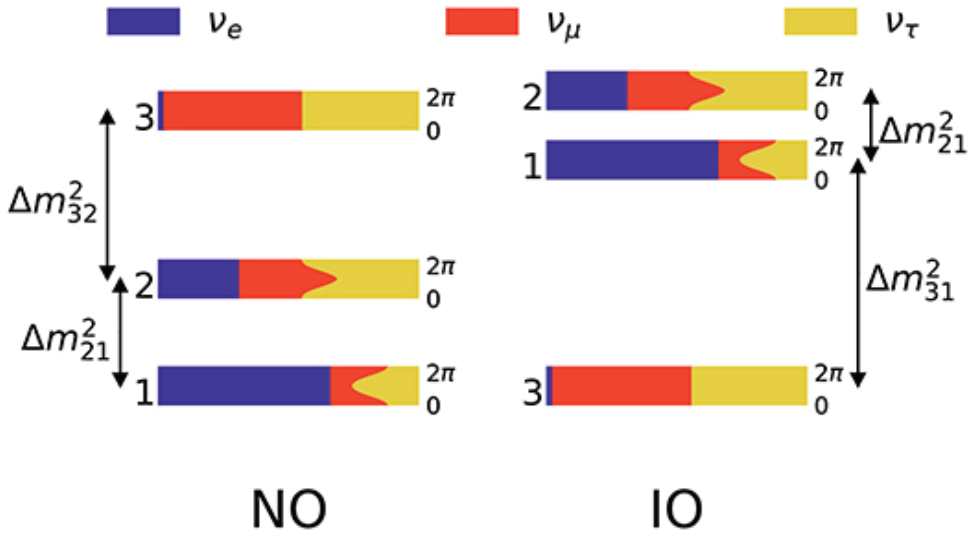


FIGURE 2.3: Neutrino mass hierarchies: Normal (Inverted) Ordering on the left (right) from [25]. The flavour composition of each neutrino mass state is illustrated in different colours. The unknown phase δ_{CP} is varied between 0 and 2π introducing variations in ν_μ and ν_τ relative contributions to ν_1 and ν_2 mass states.

neutrino sector could thus be the source of the CP symmetry-breaking on the scale needed to produce the imbalance between the matter and antimatter in the universe.

When travelling through matter the neutrino mixing behaviour is affected by the coherent scattering (the momentum of a scattered neutrino is unchanged) from the matter particles. All three neutrino flavours can interact with matter (neutrons, protons, electrons) via the neutral current interactions with the exchange of the Z^0 boson. However, in the case of electron neutrinos, a charge current coherent scattering from the matter electrons is also possible. Both NC and CC processes modify the Hamiltonian that describes the evolution of the neutrino quantum state. However, only CC interactions can lead to changes in the oscillation probabilities, since the NC processes affect all neutrino flavours and factor out. While the detailed discussion of the matter effects is beyond the scope of this work, one should mention that:

- in dense environments (such as a star interior) a resonant amplification of the flavour conversion could occur. This is called MSW effect after Mikheev, Smirnov, and Wolfenstein who first investigated the matter impact on the neutrino oscillations [26, 27];
- neutrinos propagating through the earth can also suffer substantial matter effects depending on the baseline, which can lead to different oscillation probabilities for neutrinos and anti-neutrinos potentially mimicking the CP violation.

The matter effects and the impact of δ_{CP} on the neutrino oscillation probability in the $P_{\mu e}$ channel are illustrated for neutrinos and anti-neutrinos in Fig. 2.4. The figure shows the oscillation probabilities as function of the neutrino energy at a fixed baseline (source to detector distance) of 1300 km for several δ_{CP} values and the normal mass ordering. The oscillation probabilities for anti-neutrinos are significantly suppressed compared to those of neutrinos by the matter effects, which allows determining the mass ordering unambiguously while measuring the δ_{CP} phase at the same time. This is one of the key goals of the physics program of the Deep Underground Neutrino Experiment (DUNE) described in Chapter 3.

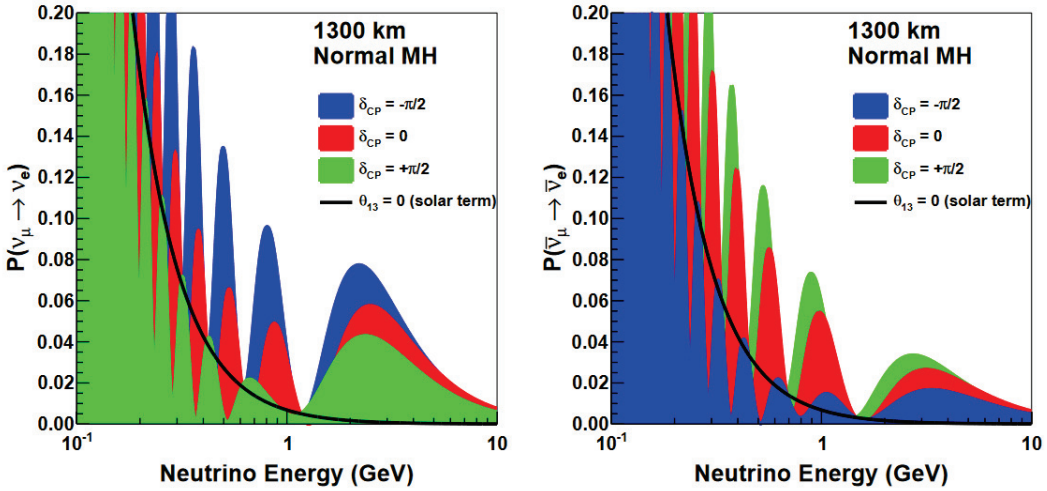


FIGURE 2.4: The appearance probability for neutrinos (left) and anti-neutrinos (right) at a baseline of 1300 km, as a function of neutrino energy, for different values of δ_{CP} and normal mass ordering from [28]. The black line indicates indicates the approximate contribution the through sub-leading Δm^2_{21} oscillation frequency computed with $\theta_{13} = 0$.

2.3 Core Collapse Supernova neutrinos

Supernovas are a great natural source of neutrino emission. Some neutrinos could be detected from the Core collapse Supernova 1987A in the Large Magellanic Cloud more than thirty years ago. These have provided valuable information on the behavior of Supernovas even if the number of detected events was limited by the detection technologies of the time. The work done in recent years on the detection of neutrinos should allow a more accurate observation of these neutrinos if a similar event were to occur again. Since the detection of the supernova neutrinos is the principal topic of this work, some description of the supernova formation and the properties of the emitted neutrino flux will be given in the rest of this chapter.

2.3.1 Core collapse mecanism

Supernova is a stellar explosion with large release of energy and light. There are two types of supernova formation mechanisms: thermonuclear supernovas (type Ia) and core-collapse supernovas (type II, Ib, Ic). Only the formation of the core-collapse supernova will be described here as these types of galactic events are the potential source for the detection by the underground neutrino experiments.

The core-collapse supernova occurs when a massive star (with a mass greater than 8 solar masses) reaches the end of its life cycle. Throughout its life a star burns through various elements in order to continuously generate energy to sustain itself from a gravitational collapse. In a first stage the hydrogen is fused into helium. As the hydrogen is burned up the star begins to contract raising its core temperature and pressure entering into the He burning stage, which, in turn, leads to a production of heavier carbon, oxygen, and neon. As helium is burned up the star contracts and core temperature is raised further allowing burning of carbon to form neon and magnesium. Subsequently neon, magnesium, and oxygen are burned to form silicon. The silicon burning is the final stage as it leads to a formation of iron, which has the highest binding energy and thus cannot be fused into any other elements to generate energy. The star's interior starts to appear as an onion-like structure illustrated in Fig. 2.5 with iron at its core and progressively lighter elements forming the layers.

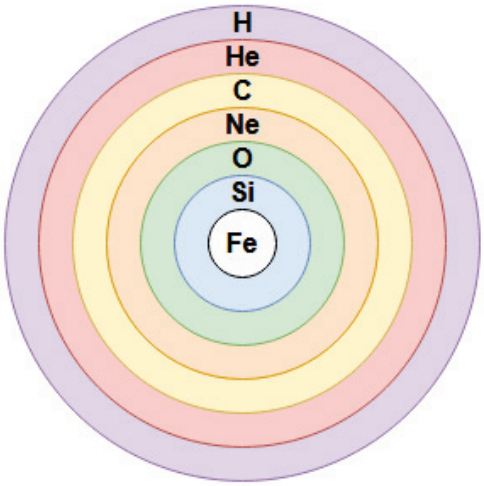


FIGURE 2.5: The different layers inside of a star which can end in a core collapse Supernova.

The density of the Fe core increases until the degeneracy pressure of the electrons is the force that counteracts the gravity and sustains the core from collapse. However, as the elements in the layers continue to fuse, the core mass continues to grow. An eventual core collapse can be triggered if the mass of the iron core exceeds the Chandrasekhar mass (about 1.4 solar masses) at which point the electron degeneracy pressure is not sufficient to prevent the collapse.

The process of the core collapse begins with the protons of the iron nuclei capturing the electrons to form neutrons and neutrinos. This is called the neutronization phase. Neutrinos escapes the core at the beginning. However, once the density of the mixture of iron and neutrons reaches about 10^{12} kg/m^3 the medium becomes opaque even for them and the core cannot be cooled anymore via the neutrino emission. As the protons and electrons are continued to be consumed a point is reached when the the electron degeneracy pressure support can no longer be available leading a sudden collapse of the central region with all the matter in a free fall moving with the speeds reaching about 25% of the speed of light. The free fall is abruptly halted when the density reaches the nuclear density values ($> 10^{14}$ kg/m^3). The in-falling matter collides with the dense inner region and bounces back creating a shock front that propagates outward blasting away the outer layers. The inner remnant can then either settle in to form a neutron star or a black hole.

As the shock front propagates outwards it is stalled by the in-falling matter. The exact mechanism for its revival is an active area of the current research where neutrinos play a prominent role [29]. The most promising scenario is via a neutrino heating: an absorption of a small part of the neutrino flux emitted from the core generates thermal pressure that reignites the shock propagation.

The energy released in a supernova is distributed as follows:

- 0.01% from the emitted luminosity;
- 0.99% from the kinetic energy of the released matter;
- 99% from neutrinos.

The signal emitted by neutrinos during a supernova is therefore a valuable information to understand and study the mechanism of the core collapse.

2.3.2 Neutrino flux

The first component of the neutrino flux produced in a supernova event is from the "neutronization" phase due to electronic captures in iron: $e^- + p \rightarrow \nu_e + n$. The ν_e flux slowly increases until the density becomes too opaque for neutrinos to escape causing a drop in the neutrino luminosity. The neutrino emission increases rapidly when the core rebounds as the propagating shock releases the trapped neutrinos. This intense burst happens on the order of few tens of miliseconds. It is followed by an "accretion" phase lasting a few hundred milliseconds when the shock is stalled by the in-falling matter. It is then followed by a "cooling" phase that lasts about ten seconds where a newly born proto-neutron star sheds its remaining energy.

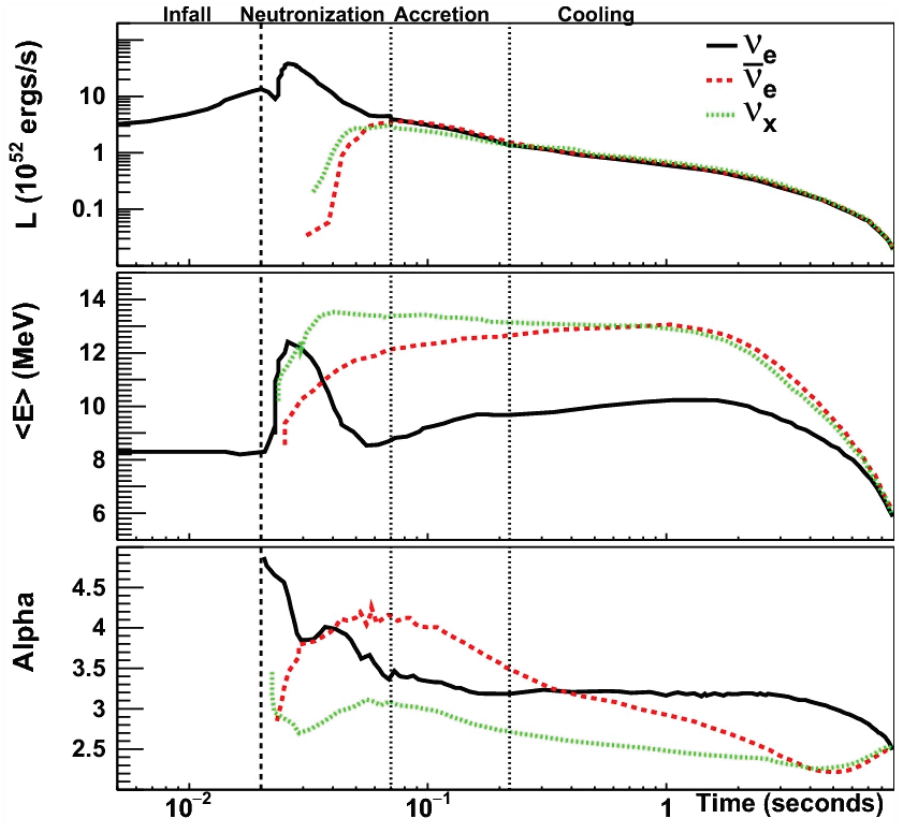


FIGURE 2.6: Illustration of the supernova time-dependant flux parameters for different neutrino flavour from [30]. The top plot shows the luminosity as a function of time, the middle plot shows average neutrino energy and the third plot shows the pinching parameter α. The vertical dashed lines delimit different regions in the supernova evolution. The line at 0.02 s indicates the time of the core bounce and the following 50 ms corresponds to the neutronization burst, which is made primarily of ν_e flavour.

Predicting neutrino the energy spectrum, luminosity, and flavour content of the neutrino flux generated in a supernova explosion is highly non-trivial due to an extreme environment within the collapsing core. Transport calculations using Monte Carlo methods [31] appear to give results that can be parametrized as:

$$\Phi(E_\nu) = N\left(\frac{E_\nu}{\langle E_\nu \rangle}\right)^{\alpha} \exp\left[-(\alpha+1)\frac{E_\nu}{\langle E_\nu \rangle}\right], \tag{2.15}$$

where:

- E_ν is the neutrino energy
- $\langle E_\nu \rangle$ is the mean neutrino energy

- α is a "pinching parameter"
- N is a normalization constant depending of the total luminosity

The flavors ν_e, $\overline{\nu_e}$ and ν_x (where $x = \mu$, $\overline{\mu}$, τ, $\overline{\tau}$) should have different average energies and pinching parameters.

The neutrino flux can be described as a time dependent function by specifying three α's for the three successive time windows: neutronization, accretion, and cooling. Fig 2.6 shows the evolution of the neutrino flux parameters as a function of time in a supernova core collapse. The top panel shows the expected neutrino luminosity, the middle the average neutrino energy, and the bottom panel the pinching parameter alpha. No flavor change due to neutrino oscillations is taken into account in the simulations, which could greatly change the results.

Fig 2.7 shows the spectra for three neutrino flavour categories obtained using the parameters from the model in the Fig 2.6. In general the supernova neutrinos are emitted with energies ranging from a few MeV to tens of MeV. In order to constraint an underlining model data have to be collected continuously at least within a 10 s time window covering all phases of the supernova evolution.

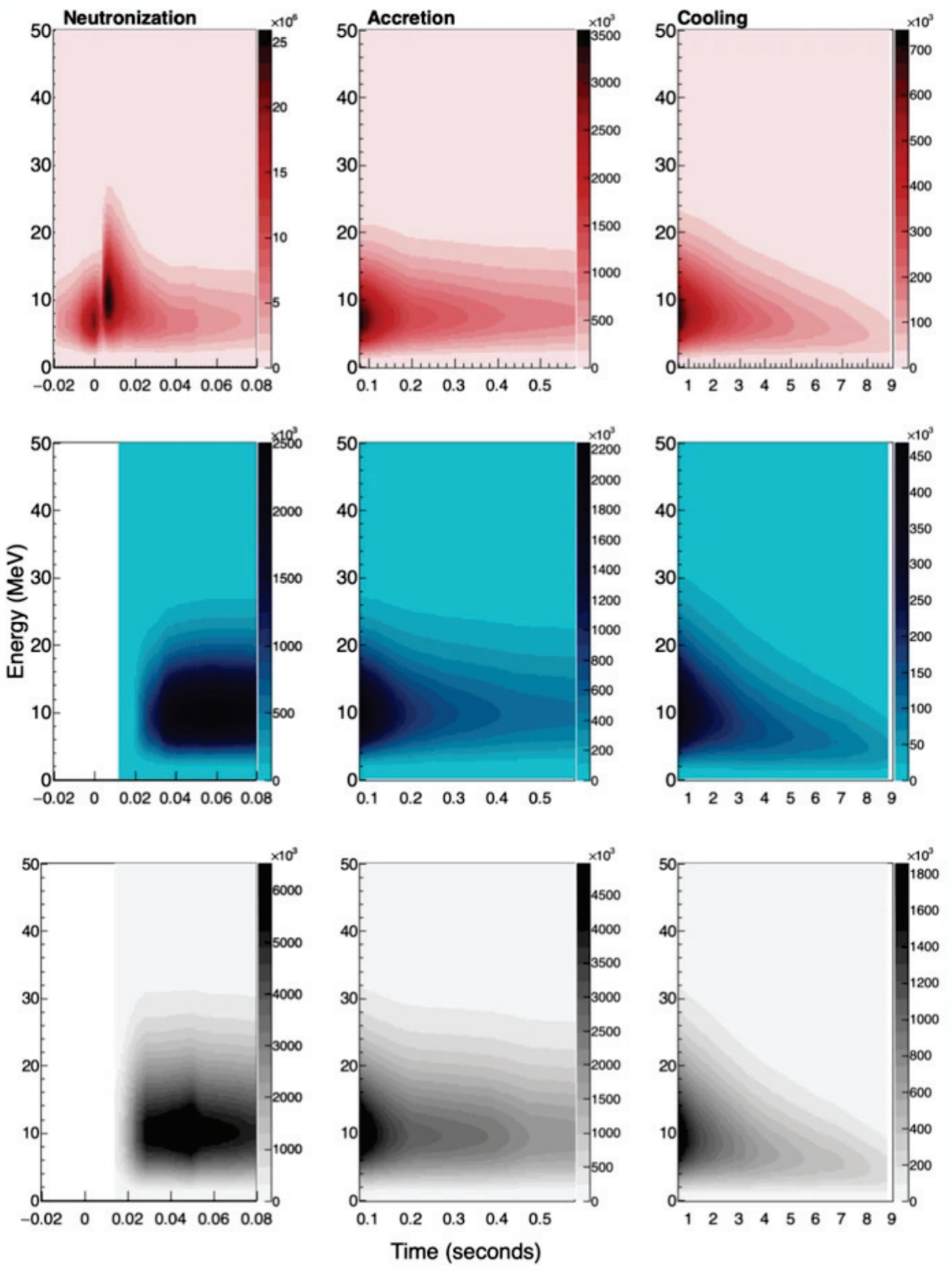


FIGURE 2.7: Time-dependent neutrino spectra shown on three different timescales for different phases in the supernova evolution from [30]. Top: ν_e. Center:$\bar{\nu}_e$. Bottom: ν_x.

Chapter 3

The Deep Underground Neutrino Experiment (DUNE)

The work of this thesis has been carried out in the framework of DUNE, the Deep Underground Neutrino Experiment. This new generation experiment will try to determine the properties of neutrinos with a unique sensitivity, never reached before. The results from DUNE will try to answer several fundamental questions such as :

- why anti-matter seems to have disappeared in the universe during the big-bang;
- to detect Supernova Neutrino Bursts (SNBs) and study the evolution of the Supernova core collapse;
- to find out whether protons are stable particles or they can decay with a very long lifetime.

In this section, a description of the experiment will be provided as well as details on the scientific objectives of the project. In particular, the detection of neutrinos from supernova events will be discussed as their detection will be one of the biggest challenges for the acquisition system which is of interest in this work.

3.1 DUNE experimental setup

In order to answer the fundamental questions listed above, the DUNE international collaboration designed and launched the construction of the experiment [32], illustrated Fig 3.1, composed of :

- a very high intensity (of the order of 1 MW) proton beam capable of generating a powerful flux of neutrinos;
- a near detector installed just after the neutrino source to measure the initial neutrino flux before neutrino oscillations;
- a far detector located 1300 km from the beam source and 1.5 km underground to measure the oscillated neutrino flux.

The experiment is currently under construction in the United States. The neutrino source is being built at the Fermi national accelerator Laboratory (Fermilab) in Illinois. The far detector will be

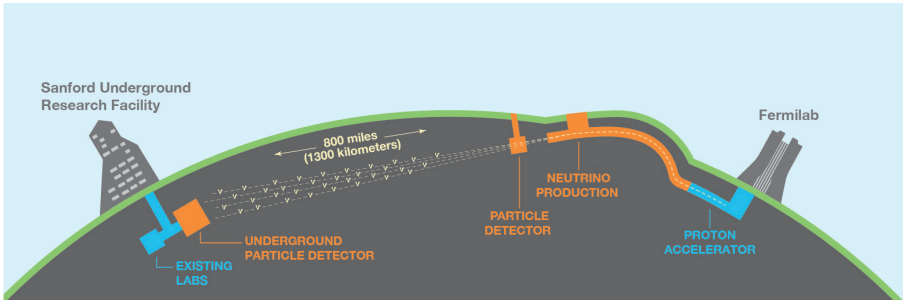


FIGURE 3.1: Overview of DUNE experiment

placed 1300 km away from Fermilab, 1.5 km underground at the Sanford Underground Research Facility (SURF) in South Dakota.

The Long Baseline Neutrino Facility (LBNF) includes at Fermilab the neutrino beam for DUNE to perform a precision study of the neutrino oscillation properties. The neutrino beamline will operate a 1.2 MW proton beam delivered by the upgraded accelerator complex at Fermilab. As part of the Proton Improvement Plan II (PIP II), the laboratory plans to build a new superconducting linear accelerator (linac) capable of delivering more than a MW beam power, upgradable to multi-MW level.

The production principle of a conventional neutrino beam is illustrated schematically in Fig 3.2. A beam of accelerated protons impinges on a target. This results in a large flux of secondary particles, created in hadronic interactions of the protons of the beam with the target nuclei. These hadronic particles predominantly consist of the lightest π mesons with some addition of K mesons and heavier hadrons. A system of magnetic lenses (horns) is then used to focus the charged pions coming out from the target (and other charged particles) into a nearly collimated beam. The horns can operate either in positive or negative polarity with the resulting magnetic field allowing to focus (and defocus for the opposite particle charges) either positive or negative particles, respectively. For a positively selected beam, the beam of secondary particles, after focalization, passes through a decay tunnel (decay pipe) where most of the pions decay to produce anti-muons and neutrinos. The negative beam polarity can be selected in case one wants to produce an anti-neutrino beam. A beam absorber is placed at the exit of the decay pipe to absorb any remaining particles apart from neutrinos.

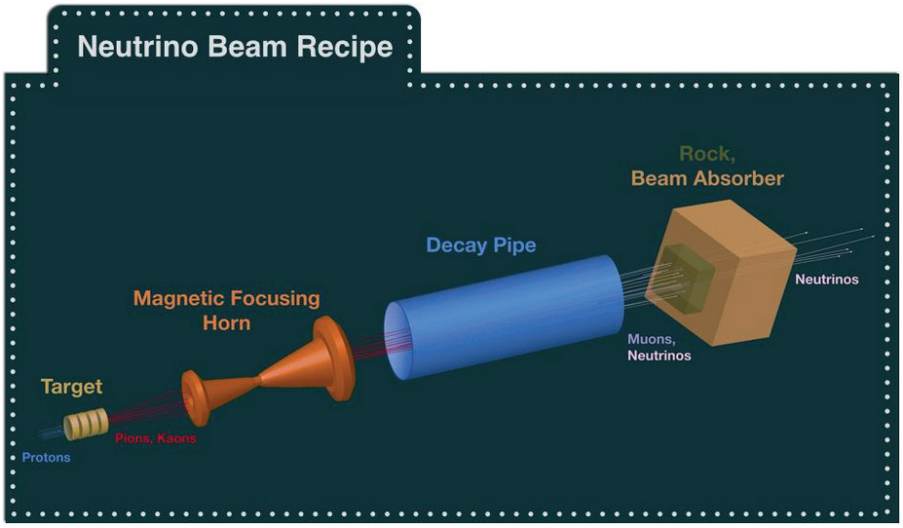


FIGURE 3.2: Neutrino Beam Recipe

In DUNE, the protons are delivered to the target after passing through a complex of particle accelerators. First, they are accelerated by the superconducting linac, then they pass through the Booster

and finally leave the Main Injector. At this point, the protons travel at a speed very close to the speed of light and have an energy of 120 GeV.

The protons are extracted from the Main Injector into the DUNE proton beamline and guided to the target hall complex that houses the target and three focusing horns (Fig. 3.3) The focused secondaries then enter into a 194m long decay pipe that ends at the hall housing the absorber.

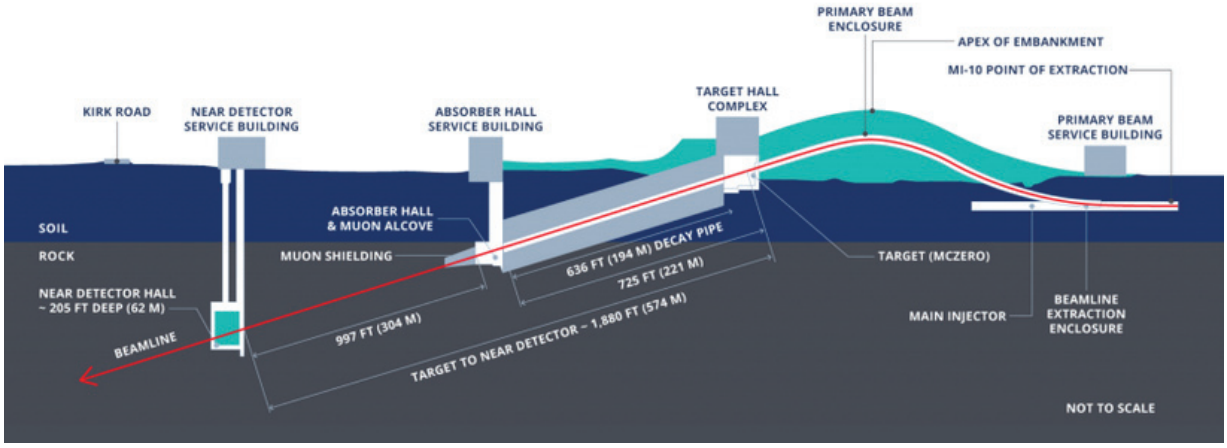


FIGURE 3.3: DUNE neutrino beamline and near detector complex.

While the neutrino flux consists predominantly of muon neutrinos (or muon anti-neutrinos depending on the polarity of the horns), it also contains $\bar{\nu}_\mu$ as well as electron neutrinos (electron anti-neutrinos for opposite focusing polarity). The latter in particular are an irreducible background for the $\nu_\mu \to \nu_e$ oscillation signal. The neutrino flux properties, such the energy spectrum and flavour composition, are monitored by a suit of near detectors located a few hundred meters downstream of the absorber hall. These would be used to constrain observations in the far detector and allow precision measurements of the neutrino mixing parameters.

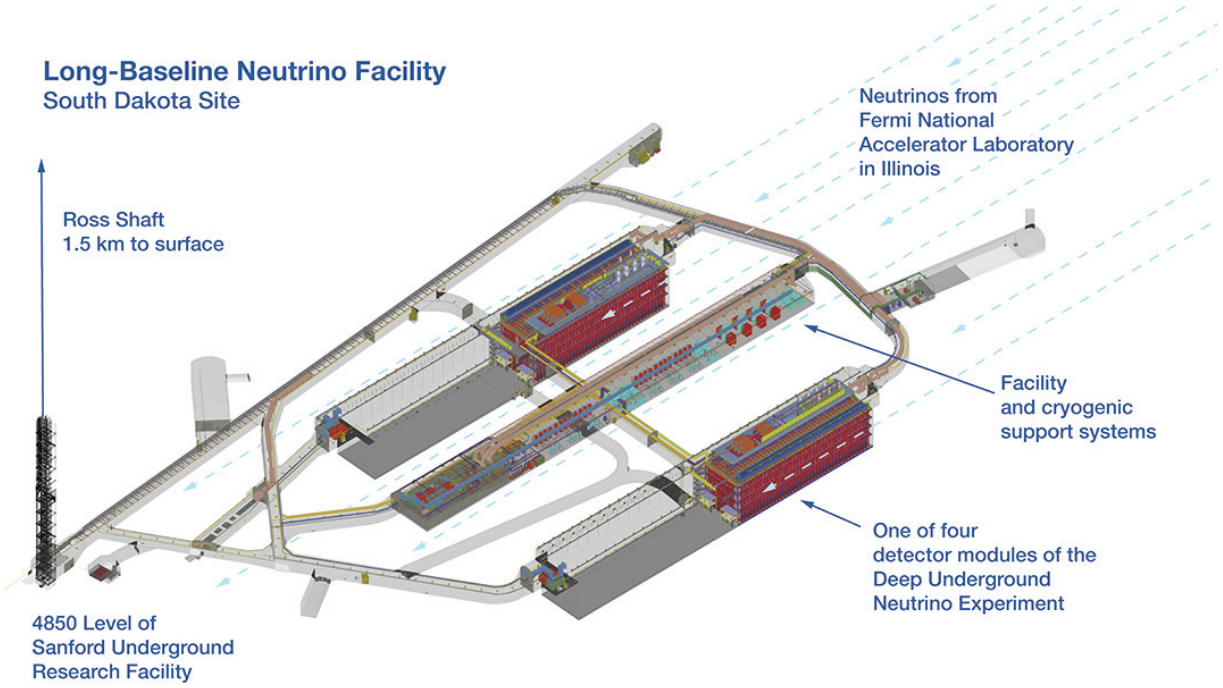


FIGURE 3.4: DUNE far detector and LBNF facility caverns at SURF.

The DUNE far detector is composed of four distinct liquid Argon time projection chamber (LAr TPC) modules. Each module contains approximately 10 kton of liquid Argon in the active volume. As illustrated in Fig. 3.4 the detector modules are hosted in two large underground caverns.

3.2 The DUNE Far Detector

3.2.1 The liquid Argon Time Projection Chamber

In the 70's, Nygren introduced the concept of the Time Projection Chamber (TPC) [33]. This particle detection technique was developed to answer the problem of detecting whole events produced by PEP (Positron-Electron Project).

The principle of a TPC (see Fig. 3.5) is to exploit a volume filled with a detection medium, a gas or a liquid. When charged particles arrive, they ionize the medium on their path and leave a track made by a cloud of electrons all around the trajectory of the particle. Moreover, the medium is immersed in a uniform electrostatic field defined by an anode and a cathode surface placed at the opposite ends of the volume. The electric field separates the electrons from the positive ions at the moment of the ionization and allows drifting the electrons of the track towards the anode. It is very important that the medium should not reabsorb the electrons during their drift path to the anode.

At the level of the anode, a segmented charge collection system is installed in order to capture the electrons and reconstruct their arrival position. The detection of the track can thus be carried out in 2 dimensions by using for instance a anode detection plane instrumented with two orthogonal views. This is the projection of the track on the anode. Finally, by knowing the constant drift speed of the electrons in the medium (related to the value of the electrostatic field) and by measuring the drift time of the electrons, it is possible to reconstruct also the 3rd space coordinate of each point of the track and to perform a 3D reconstruction.

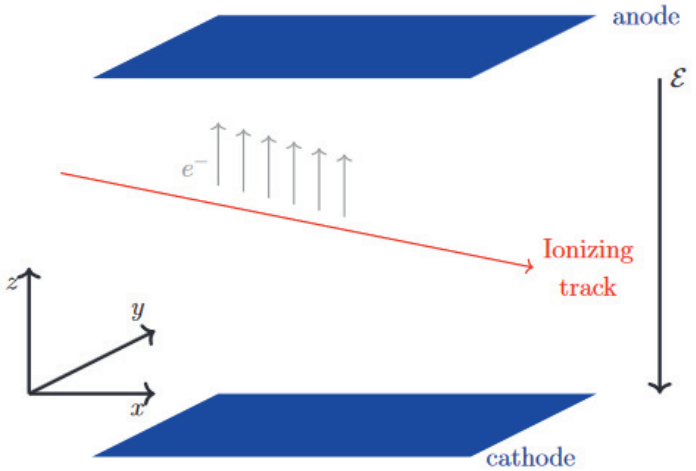


FIGURE 3.5: The Time Projection Chamber working principle from [34]

In the case of DUNE, the medium used as target for the interaction of neutrinos and the detection of the secondary charged particles coming out from their interactions is liquid Argon. The idea to develop a LAr TPC detector was launched by C. Rubbia in 1977 [35]. Liquid Argon has indeed many qualities:

- it is a dense medium, which increases the probability of neutrinos to interact.

- it has a high electron mobility and is neutral, which allows the electrons not to recombine.
- it is present in large quantities on earth (1% in the air in its gaseous form) and is inexpensive.

In order to answer the questions DUNE is addressing - such as the search for the violation of the CP symmetry; the search for proton decay to get closer to Grand Unification theories or the observation of neutrinos coming from supernovae - a detector of unprecedented dimensions is needed. It requires a detector capable at the same time of being a very massive target in order to have enough neutrino interactions, and of ensuring a good energy and space resolution, below a centimeter, to image the details of these events.

The DUNE far detector, designed to meet these requirements, will constitute a total fiducial mass of 40 kton. More precisely, this detector will be separated into four sub-modules containing each a LAr TPC with more than 10 kton fiducial mass. The details of the detection/acquisition techniques of the tracks left by ionizing secondary particles may differ from one module to the other. In the following, the single-phase LAr TPC technology foreseen for the first DUNE Far Detector module will be presented and then the Dual Phase technology that has been studied as a possible design for the second module.

3.2.2 Single-Phase (SP) LAr TPC technology

The single-phase LAr TPC technology is not new, even if it has never been implemented on modules of this size. Indeed, this detector design has already been used successfully by the ICARUS [36], ArgoNeuT [37], MicroBoone [38], LArIAT [39] experiments and implemented in a DUNE prototype built at CERN, ProtoDUNE single-phase [40]. The name single-phase implies that all the detection of the ionization is entirely performed in liquid Argon.

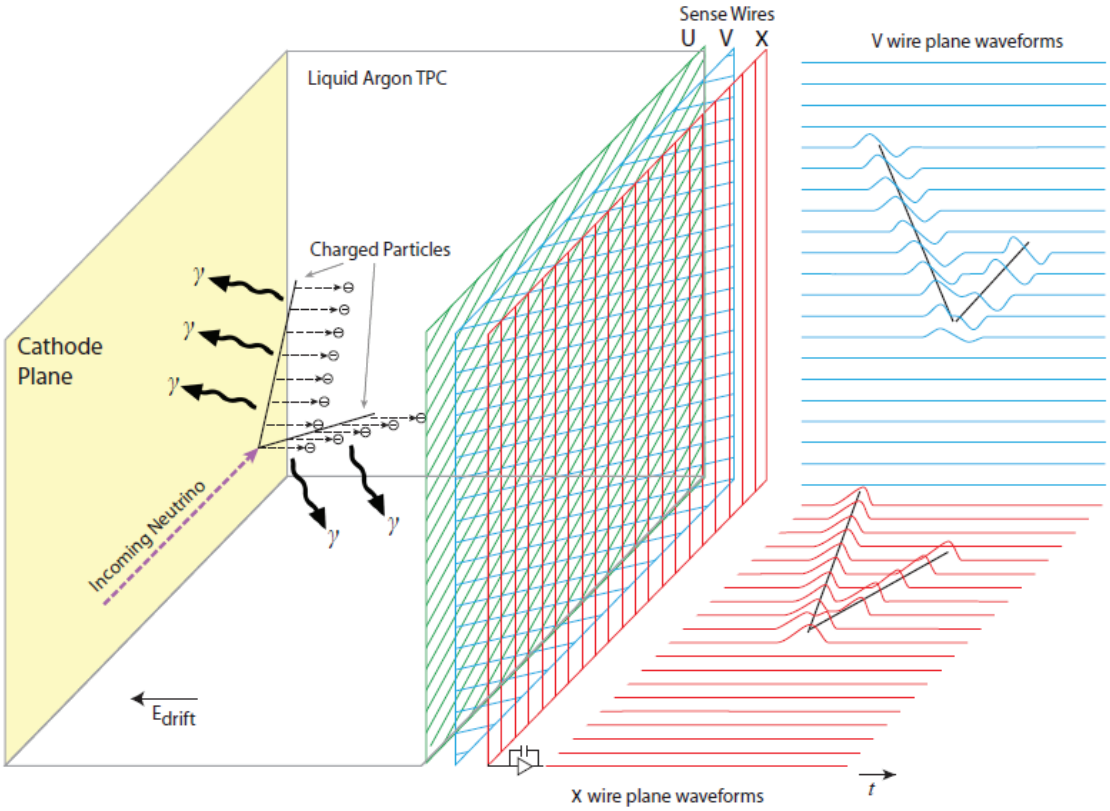


FIGURE 3.6: LAr TPC principle in the case of the SP technology

Figure 3.6 illustrates the working principle of the LAr TPC in case of its single-phase version. The detection volume filled with liquid Argon is surrounded by an anode, a cathode and a set of equipotential rings at degrading voltages, constituting the field cage. The LAr volume is then immersed in a uniform electric field of several hundred volts per centimeter. This field allows the electrons, produced by the passage of charged particles in the active volume of LAr, to drift in a few ms towards the detection surface on the anode. The anode is normally constituted of wire planes, with a few mm pitch in between wires, forming a grid which allows the reconstruction of the X and Y coordinates of the detected charges.

Part of the ionization produced by charged particles moving in LAr, will locally recombine along the track, emitting scintillation light at a wavelength of 127 nm. This wavelength can be converted to visible light and captured by photon detectors. The detection of the prompt component (emitted in a few ns since the time of the ionization) of the scintillation light provides information about the time at which the particle has ionized the LAr. By measuring the arrival time of the electrons at the anode and by knowing the value of constant velocity of the electrons in the drift field, it is then possible to reconstruct the 3rd spatial coordinate Z, perpendicular to the anode plane, of the detected charges.

The module built with the single-phase technology for DUNE will be placed underground at 1500 m depth at the SURF laboratory in South Dakota [41]. The external dimensions of the cryostat will be $65.8m \times 17.8m \times 18.9m$. Four volumes (see Fig. 3.7) of 3.5m each will be defined between 5 alternating cathodes and anodes planes where each plane will have a surface of $58m \times 12m$.

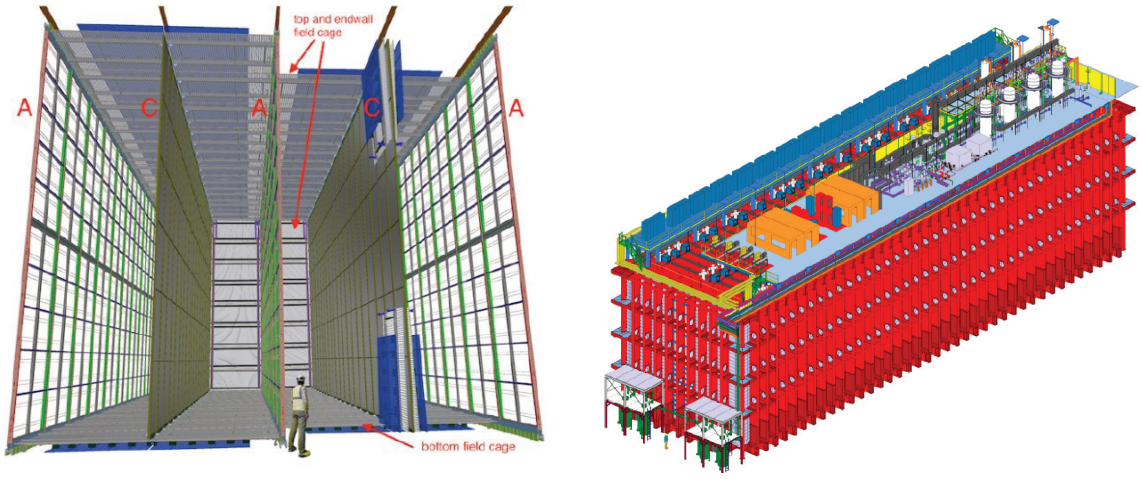


FIGURE 3.7: Single-phase DUNE Far Detector first module

Each cathode plane is composed of an array of Cathode Plane Assemblies (CPA). Each CPA is a $4m \times 1.2m$ panel. Each cathode plane is composed of 150 CPA. The CPAs operate at a voltage of -180 kV, which define, together with the anode potential close to ground, a uniform electrostatic field of 500V/cm. The electrons drift at 1.6mm/us in this field. The maximum drift time from a cathode plane to a anode plane is 2.2ms.

The anode planes are composed of 50 Anode Plane Assemblies (APAs). The APA (see Fig. 3.8) are wire planes embedding also the photon-detectors, each one covering an area of $6m \times 2.3m$. The APAs are mounted vertically in the detector, so that 2 APAs occupy the full height of the detector and 25 APAs the full length.

The front-end acquisition electronics is connected to the top part of the top APA and to the bottom part of the bottom APA. The readout amplifiers are immersed in liquid Argon and thus benefit of large noise reduction. Different ASIC circuits are used to shape, amplify and digitize the signals at this level.

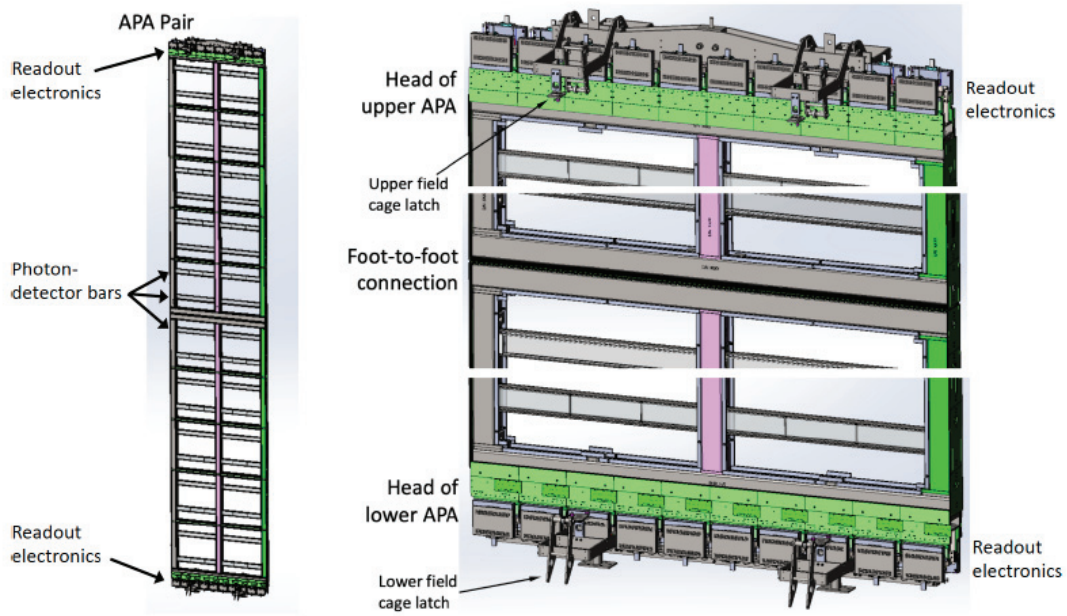


FIGURE 3.8: Single-phase Anode Plane Assembly (APA)

3.2.3 Dual-Phase (DP) LAr TPC technology

The module based on the dual-phase design shares the same scientific goals as the single-phase module [42]. In order to achieve them, as shown in the previous section, the single-phase LAr TPC technology allows a very good spatial and energy resolution of the detected secondary particles. The dual-phase technology provides an increase in the space resolution and an improvement in the signal to noise ratio, which can be exploited in order to lower the detection threshold for certain phenomena occurring at low energies. The dual-phase technology also allows for the exploitation of larger drift volumes, thus reducing the amount of passive materials which need to be installed in the active LAr volume.

The basic principle (see Fig. 3.9) of the DP detection technology is similar to SP. As in SP, a charged particle passing through the LAr TPC ionizes the Argon. Thanks to a uniform electric field the electrons produced along the track then drift towards the detection surfaces.

The main difference between the two technologies is that the anode readout channels will not operate immersed in LAr but will be placed just above the liquid argon surface, in the Argon gas phase present in the ullage space of the cryostat. In the gas phase, it is possible to amplify the signal of the electrons by exploiting avalanches occurring in high field regions present in micro-pattern gas detectors. An avalanche is a process where the electrons are accelerated by an electric field at the point of acquiring enough energy to be able to produce additional ionization when hitting other Ar atoms. From a single electron, it is then possible to produce several secondary electrons. Given the mean free path of the electrons, avalanches are not feasible in the liquid phase since they would require very high electric fields in order to be able to accelerate enough the electrons before they enter in collision with Ar atoms. The gain obtainable thanks to the avalanches in the gas provides the advantage of boosting the signal to noise ratio, being the noise determined by the noise performance of the cryogenic amplifiers, similar to the ones of the single-phase.

In a DP module this operation principle implies drifting the electrons vertically and extracting them from the liquid to the gas phase, while the electrons drift horizontally towards the APAs in a SP module. In DP the electrons drift vertically in LAr until reaching an extraction grid located just below the liquid-gas interface. The grid defines, in between its position and the one of the anode in

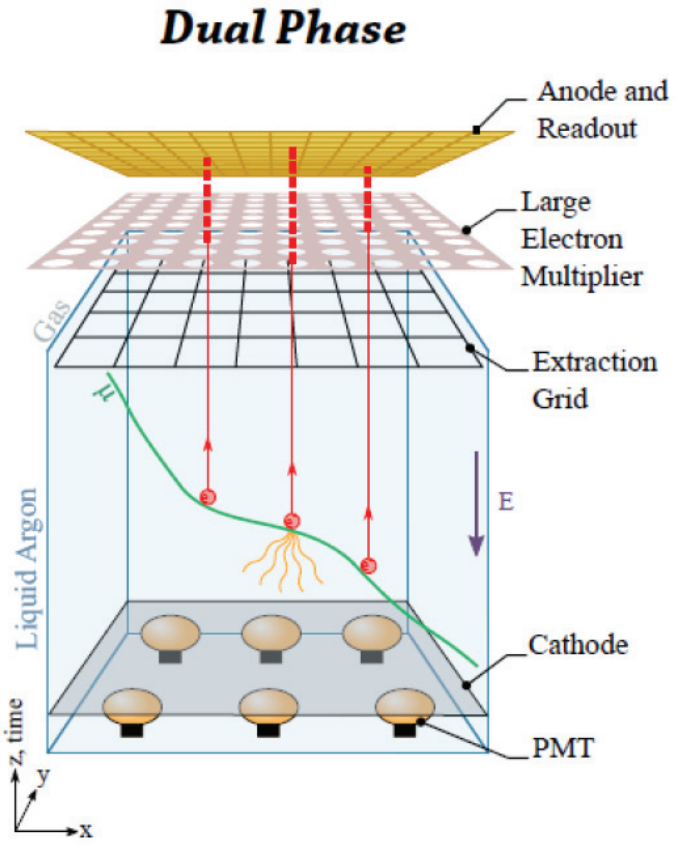


FIGURE 3.9: Dual-phase LAr Time Projection Chamber principle

the gas, a region with an electric field intensity more than four times larger than the one in the drift region. This higher extraction field allows the electrons passing from the liquid to the the gas phase. At this point, the electrons encounter micro-pattern gas detectors, the Large Electron Multipliers (LEM) which allow their amplification in avalanches. The typical gain of the LEM operating in pure argon is about 10. The amplified electrons are then collected by a segmented anode plane, implemented in a printed circuit board subdivided in two 2 orthogonal sets of copper strips with a pitch of 3.125mm, allowing the spatial reconstruction of X and Y coordinates.

Finally, similarly as for the SP, the Argon scintillation light is captured by Photo-Multiplier Tubes (PMTs) located under the cathode in order to obtain the information of the time when the ionization is produced. The measured drift time is then converted into the Z spatial coordinate, which allows the reconstruction of a 3D image.

The use of DP technology allows instrumenting very large detection volumes without any dead material inside. The operation over long drifts is made possible by the gain in the avalanches which compensates for the electron losses during the drift. Drifting over 12m takes of the order to 8ms. Oxygen is an electro-negative element susceptible of absorbing the electrons. The original goals for the DUNE LAr purification system were foreseeing to further purify commercial LAr (typically delivered by the industry with a few ppm oxygen contamination) at the level of containing a residual oxygen contamination of 0.1 ppb. This level of contamination can be translated in a lifetime of the electrons in LAr of about 3ms, to be compared to 8ms needed for the maximal drift time. The amplification of the electrons by the LEMs allows then compensating for signal losses related to impurities encountered over these very long drift distances. The readout granularity can also be further increased with respect to the SP version, reaching a 3mm pitch. Moreover, despite the finer pitch, because there are only

collection strips and a single detection plane due to the long drift, in DP there are less readout channels to be instrumented: 153k channels for a DP module versus 384k for SP.

In DP, the front-end cryogenic electronics is not immersed in LAr but it is operating in dedicated volumes located in the cold gas phase and tightly separated from the pure argon volume. A final advantage of DP is that the cryogenic front-end electronics is then externally accessible from the cryostat roof without contaminating the pure argon volume. The replacement of the front-end cards can be performed even when the detector is still in operation, differently than in a SP module, where the electronics is condemned inside the LAr volume and not accessible without emptying the detector.

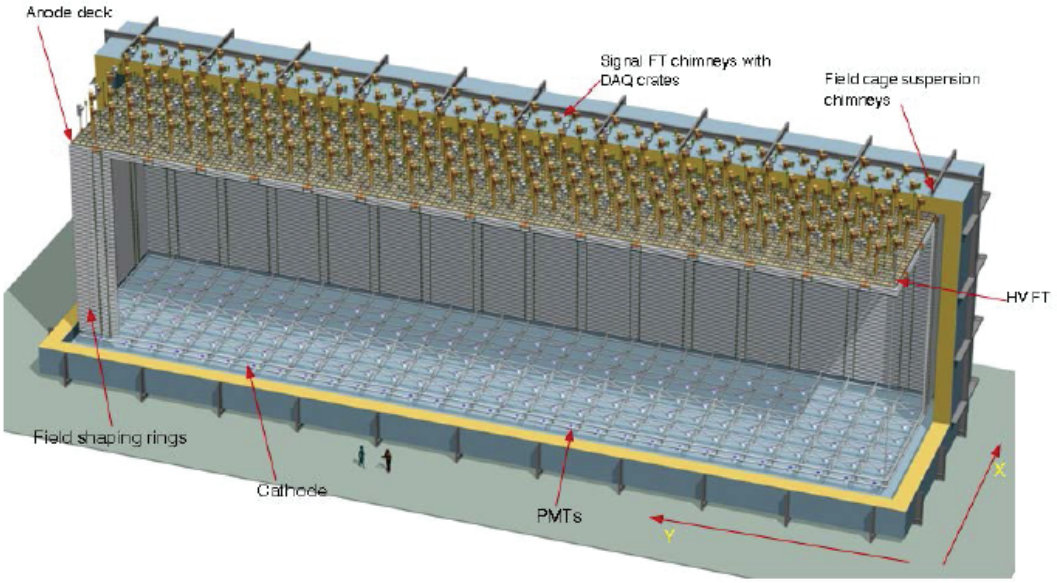


FIGURE 3.10: Dual-phase DUNE Far Detector module

The DP module foresees a LAr detection mass of 12kton for a module containing a single detection part of 60m x 12m x 12m. The cathode and PMTs will be placed at the floor level of the cavern while the anode and all the acquisition electronics will be placed at the roof level of the detector.

3.3 DUNE prototypes at the CERN Neutrino Platform

The DUNE prototyping efforts are supported by the CERN Neutrino Platform where two large scale prototypes (protoDUNEs) for the SP and DP technologies were built since 2017. These cryostats of 11m external dimension have both an inner volume of $8.5 \times 8.5 \times 8.5 m^3$.

ProtoDUNE SP operated in between the fall of 2018 and summer 2020, protoDUNE DP operated in between the summer 2019 and the fall 2020. ProtoDUNE DP had a maximal drift of 6m, achieved in a field cage surrounding an active LAr volume of $6 \times 6 \times 6 m^3$. The anode detection surface, organized in four independent Charge Readout Planes (CRP) of $3 \times 3 m^2$, represents about 1/20 of the one of a full DP DUNE module, which should include 80 Charge Readout Planes (see Fig. 3.11).

The readout strips of the CRPs are connected to the cryogenic analog electronics read out by the digital front-end system described in the next chapter. Both systems have been developed at IP2I Lyon as the outcome of an R&D effort started in 2006. These readout systems performed well in protoDUNE DP and represented the demonstrator of the readout system for a DUNE DP detector module. Some examples of images of cosmic ray events collected with protoDUNE dual-phase are shown in Fig. 3.12.

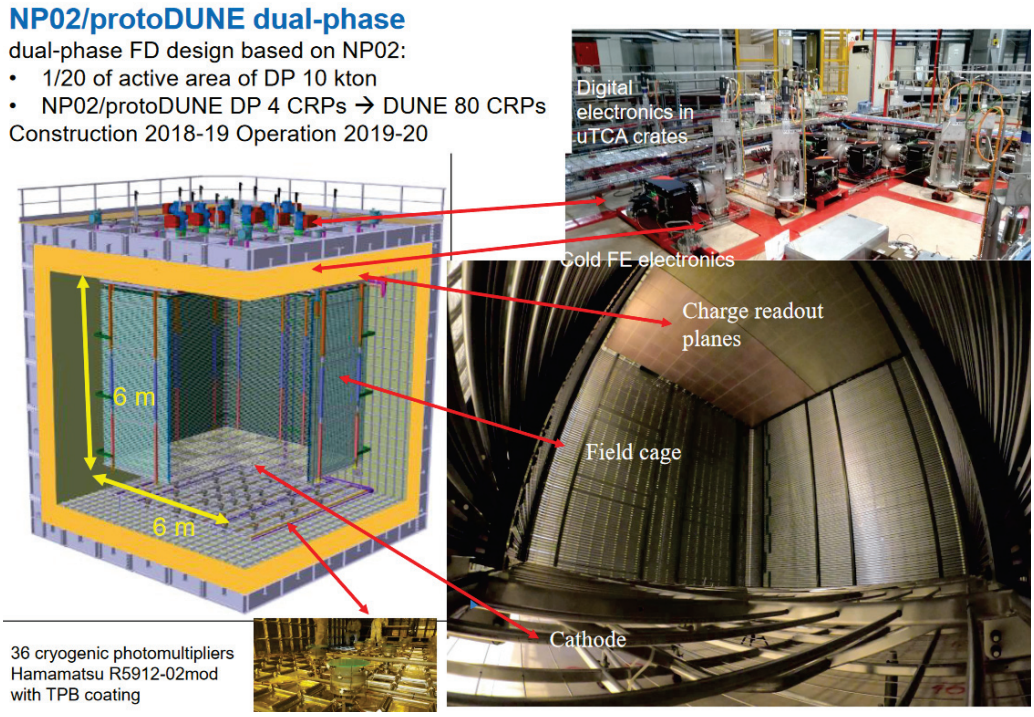


FIGURE 3.11: The protoDUNE dual-phase LAr TPC at the CERN Neutrino Platform.

The developments presented in these thesis concern then the DAQ back-end system to be interfaced to the DP front-end digitization system. The DAQ back-end has the task of collecting the data from the various front-end units and defining the triggers used to make the decision to write the data on disk, during the operation of the underground detector module.

Differently than the underground DP far detector module, protoDUNE-DP was operating on surface, exposed to a large flux of cosmic rays. This flux is suppressed by a factor one million in the underground location at SURF.

The protoDUNEs could also be exposed to test-beam particles produced by the CERN accelerators chain. Given its modus operandi on surface the readout of ProtoDUNE-DP was organized on the basis of external triggers which could be produced either by the beam particles, by cosmic ray counters or by random triggers generators.

Given the presence of an external trigger a drift window of 4ms (starting at the trigger time and corresponding to 6m drift) was then entirely readout and stored on disk. The DAQ back-end was not operating any selection of the events and it was designed to cope with an external trigger rate going up to 100Hz, corresponding to a data bandwidth reaching 20GB/s in case data were produced from the digitization units without applying any compression.

Being the dual-phase front-end electronics the same for protoDUNE-DP and the DP far detector module, the readout of the DUNE underground module, has to be organized differently. The readout of the DP module underground implies the continuous data streaming from the front-end digitization units to the DAQ back-end system. Internal triggers are issued by the back-end system itself when self-detecting the presence of interesting events.

Being the cosmic rays flux strongly suppressed at the underground location, the activity of the detector is dominated by the natural radioactivity of LAr due to the ^{39}Ar isotope (1Bq/kg). Interesting events are very rare. Cosmic rays cross the entire module at the level of about one muon every ten seconds. The neutrino beam from Fermilab can produce just up to a couple of thousands interactions per module per year.

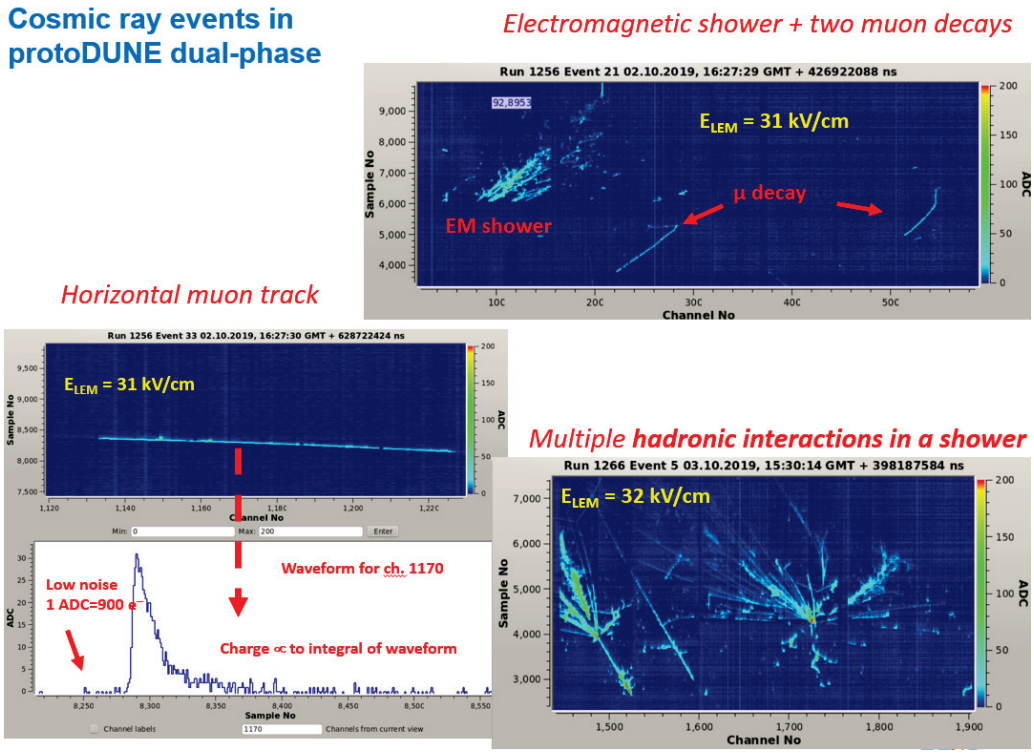


FIGURE 3.12: Examples of images of cosmic ray events collected during the operation of NP02/protoDUNE dual-phase.

3.4 Evolution of the dual-phase technology to Vertical Drift

The operation of the two ProtoDUNEs until fall 2020 allowed gathering experience and learning about several aspects. This resulted in a further simplification and optimization of the dual-phase design.

The initial requirements in terms of lifetime of the electrons in LAr were corresponding to 3ms. However the protoDUNEs achieved purity records about one order magnitude larger, strongly reducing the need for amplification at the level of the anodes. However, the detector design and configuration developed for the dual-phase were still very appealing for the DUNE experiment from practical and economical points of view.

Recent developments since 2020, as well as the experience acquired with protoDUNE dual-phase, showed how to evolve the dual-phase design into a new version (**Vertical Drift**) which no longer requires amplification in the gas phase.

The Vertical Drift preserves all advantages of the dual-phase design, based on strips implemented on printed circuit boards for the readout of the electrons on the anodes, instead of wires as for SP.

This new Vertical Drift design, based on perforated anodes operating directly in LAr, is also more robust than DP with respect to issues related to the long term stability of the micro-pattern detectors used in the gas phase (development of sparking activity) and to their dependence on environmental conditions inside the cryostat, such as: bubbles, perturbations of the LAr surface at the interface in between the liquid and the gas and possible presence of floating debris.

Perforated anodes represent a simplification in a single and simpler PCB of the two PCB stack used in DP made by the perforated PCB of the LEMs and the one of the segmented anodes (see Fig. 3.13).

As in DP, perforated anodes are integrated in CRP planes of about 3m size. They are capable of providing readout signals as strong as a DP anode operating in the gas phase at a gain of 6.

However the Vertical Drift perforated anodes can operate simply immersed in LAr, avoiding thus all the complications of the DP CRPs which have to be precisely aligned in the gas at a few mm distance from the LAr surface and which are sensitive to the conditions of the LAr-gas interface.

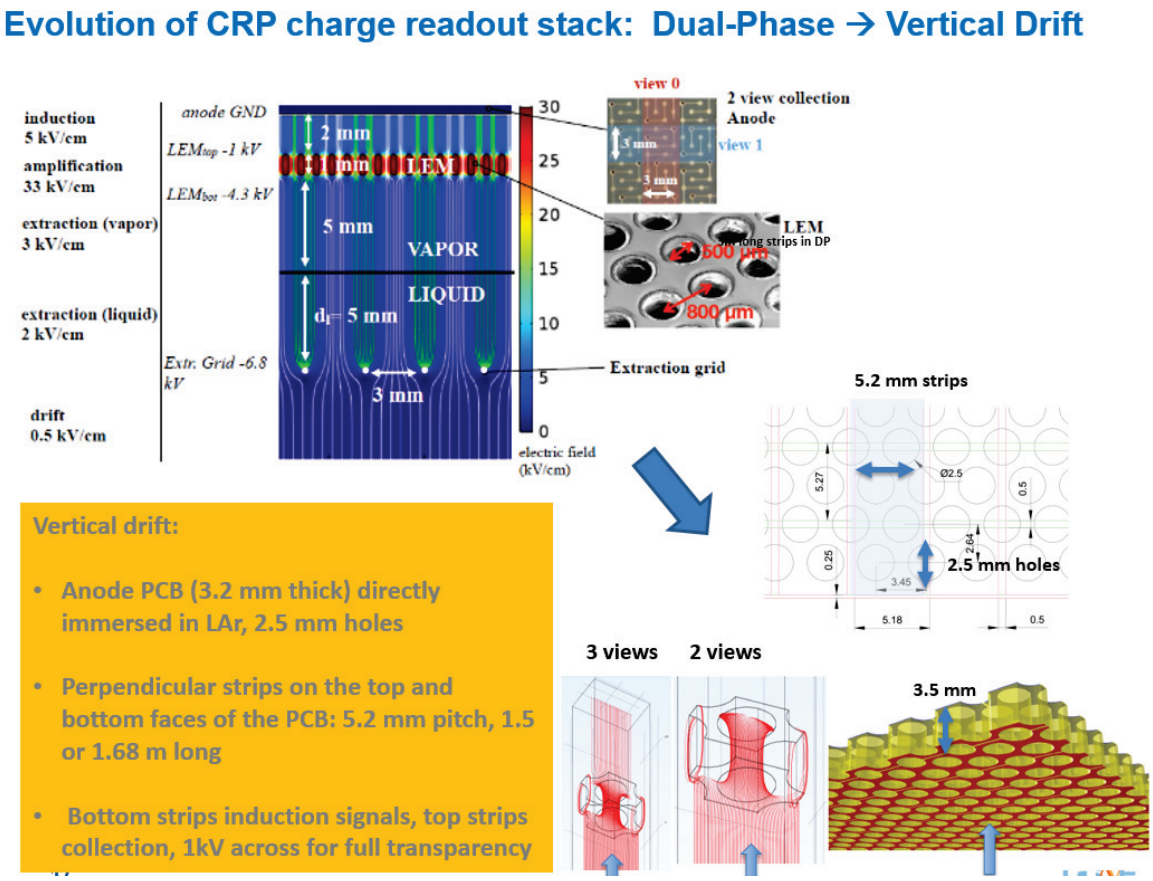


FIGURE 3.13: Evolution of the dual-phase anode stack design to the simpler Vertical Drift perforated anodes.

The DP construction and installation advantages related to the CRP design and the modular vertical geometry of the field cage are completely maintained in the Vertical Drift.

Another optimization process for the Vertical Drift design was performed by splitting the drift space in two vertical volumes (top-drift and bottom-drift, see Figure 3.14). This arrangement reduces the high voltage requirement at the cathode. In order to generate a uniform electric field of 500V/cm -300kV are needed instead of -600 kV in the DP design. The lower requirement on the high voltage removes the necessity for the additional R&D which was still ongoing in order to achieve 600 kV.

Skipping the need for additional R&D steps concerning the optimization of the LEM gas detectors and the High Voltage to reach 600 kV, together with the relaxation of environmental aspects on the liquid-gas interface inside the cryostat, contributed overall in speeding up the construction program for the second DUNE Far Detector module.

The top-drift volume preserves also full accessibility to the charge readout electronics associated to the top CRPs, which are suspended from the roof, as in the dual-phase design (see Figure 3.15). The electronics design developed for the dual-phase is also better suited to the layout of top-drift CRP structure (see Figure 3.16), with beneficial aspects on the CRP structures mechanics, heat dissipation and mitigation of risks which may affect the top of the cryostat, as the presence of bubbles.

From the point of view of the physics sensitivity a large design effort was made, thanks to the Vertical Drift configuration, to ensure a maximal exploitation of the cryostat LAr volume. The fiducial

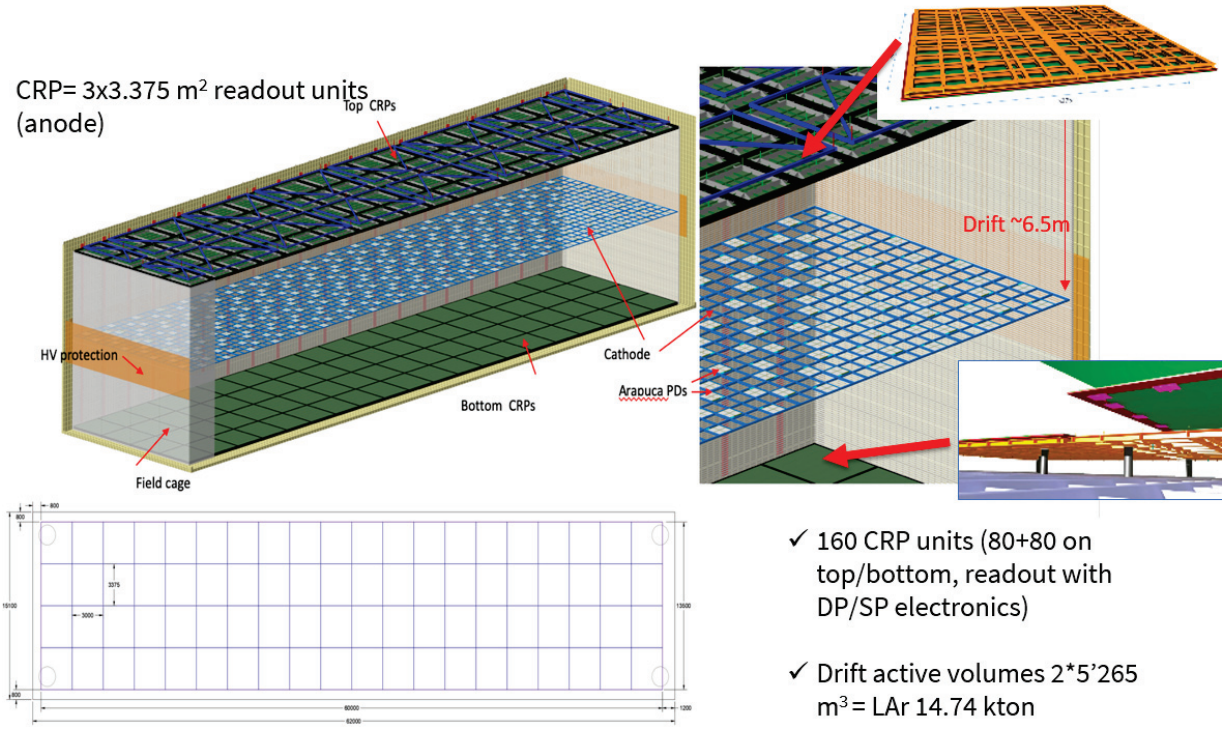


FIGURE 3.14: 3D view of the Vertical Drift DUNE Far Detector module with the top and bottom drift volume separated by the cathode at middle height of the detector.

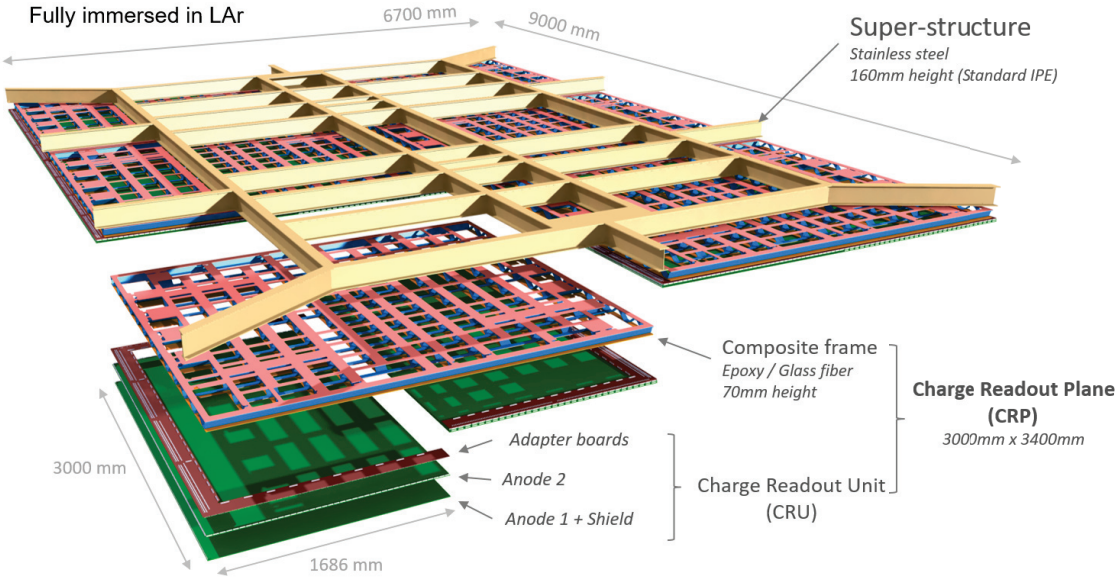


FIGURE 3.15: Vertical Drift CRP superstructures for the top-drift volume.

mass for the second DUNE Far Detector module based on the Vertical Drift design reaches almost 15 kton. The optimization of the target mass for neutrino interactions, together with cheaper and faster construction and installation schedules affecting the second Far Detector module, represent an important enhancement for the physics program. The charge readout reconstruction performance, per unit of target mass, for the second DUNE Far Detector module is similar to the one of the first DUNE module but the larger target mass allows collecting more neutrino interactions.

The Vertical Drift design became then in 2020 the choice of the DUNE collaboration in order to implement the second Far Detector module. Being the front-end readout system the same between

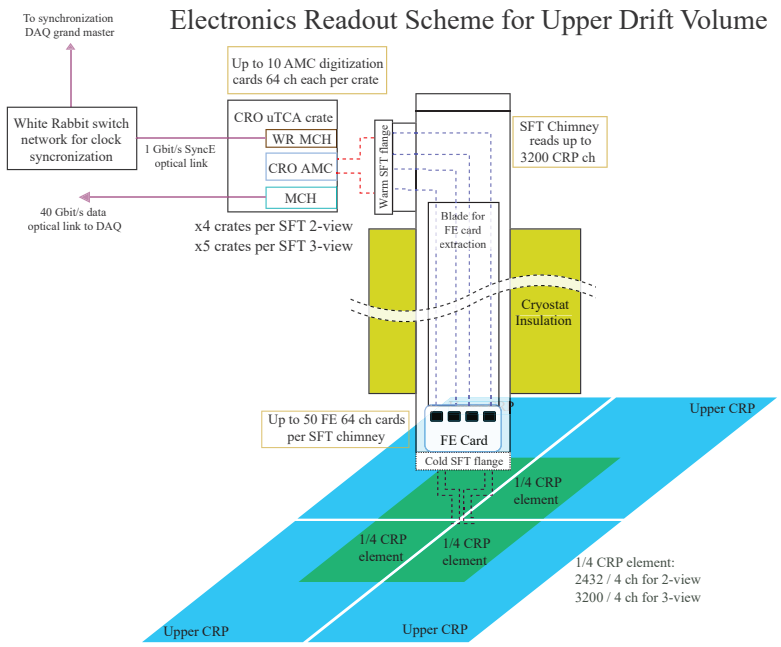


FIGURE 3.16: System architecture of the top-drift charge readout electronics.

the DP module and the top-drift of the Vertical Drift module, many of the studies performed in this thesis are also applicable to the Vertical Drift.

3.5 Detection of core-collapse Supernova neutrinos in DUNE

In order to meet its various scientific objectives, DUNE is planning to observe several types of events in its LAr TPC far detector modules:

- Neutrinos from the neutrino beam generated at Fermilab
- Neutrinos produced in cosmic ray showers in the atmosphere
- Neutrinos from core-collapse Supernovae
- Possible proton decay candidate events

Neutrinos from core-collapse Supernovae and their detection in the DUNE will be detailed in this chapter because they then play an important role in the data acquisition aspects which will be detailed later.

3.5.1 Supernova neutrinos detection channels

As stated in the section on neutrinos from Supernova bursts, a stream of neutrinos will be released by the Supernova explosion, mostly over a period of about ten seconds. A very particular neutrino signature is thus expected in case of a Supernova explosion. The interest that DUNE has compared to other detectors is its unique sensitivity to electron neutrinos, which represents an important part of the flux from Supernova.

The dominant interaction of Supernova neutrinos is the Charged Current absorption of electron neutrinos with ^{40}Ar,

$$\nu_e + {}^{40}\mathrm{Ar} \rightarrow e^- + {}^{40}\mathrm{K}^* \tag{3.1}$$

For which the observables will be the track of the resulting electron and the de-excitation products of potassium 40.

Other observable interactions that can occur with other neutrino flavors are:

- CC absorption of $\overline{\nu_e}$: $\overline{\nu_e} + {}^{40}\mathrm{Ar} \rightarrow e^+ + {}^{40}\mathrm{Cl}^*$
- Elastic Scattering (ES) with electrons : $\nu_x + e^- \rightarrow \nu_x + e^-$ $(x = e, \mu, \tau)$
- Neutral Current (NC) interaction : $\nu_x + {}^{40}\mathrm{Ar} \rightarrow \nu_x + {}^{40}\mathrm{Ar}^*$ $(x = e, \mu, \tau)$

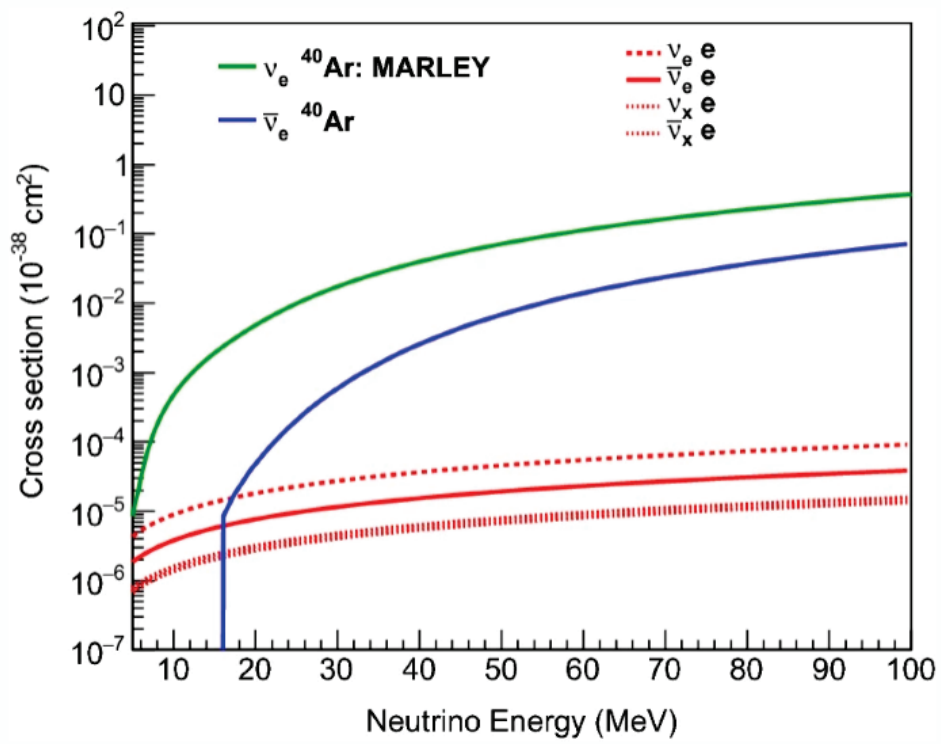


FIGURE 3.17: Interaction cross-sections of Supernova neutrinos as a function of their energy

Fig 3.17 shows the interaction cross-sections (this is a quantity defining the probability of the interaction, the event rates are proportional to the cross-sections) in Argon of Supernova neutrinos as a function of their energy. These cross sections have never been directly measured at these neutrino energies, they come from the modelization implemented in MARLEY (Model of Argon Reaction Low Energy Yields) which is a software generator of low energy neutrino interactions.

3.5.2 Supernova neutrino events rate in DUNE

Based on the different models of the neutrino fluxes from Supernova, estimates of the number of neutrinos observable for each type of interaction are available in Table 3.1. Three models (Livermore, GKVM and Garching) predict from a thousand to a few thousand interactions in the detector for a Supernova at 10kpc distance and Supernova and a total far detector mass of 40kton.

Channel	Livermore	GKVM	Garching
$\nu_e + {}^{40}\mathrm{Ar} \rightarrow e^- + {}^{40}\mathrm{K}^*$	2648	3295	882
$\overline{\nu_e} + {}^{40}\mathrm{Ar} \rightarrow e^+ + {}^{40}\mathrm{Cl}^*$	224	155	23
$\nu_x + e^- \rightarrow \nu_x + e^- \ (x = e, \mu, \tau)$	341	206	142
Total	3213	3656	1047

TABLE 3.1: Number of neutrinos events expected in the detectors for different models and interaction channels

These three models differ due to several factors. First, no flavor changes are taken into account in the Livermore and Garching models, while the GKVM model includes collective effects. The other differences are mainly differences in numerical simulation and physical parameters.

It is important to note that these results are directly related to the detection volume as well as the distance at which the Supernova occurs as shown in Fig. 3.18.

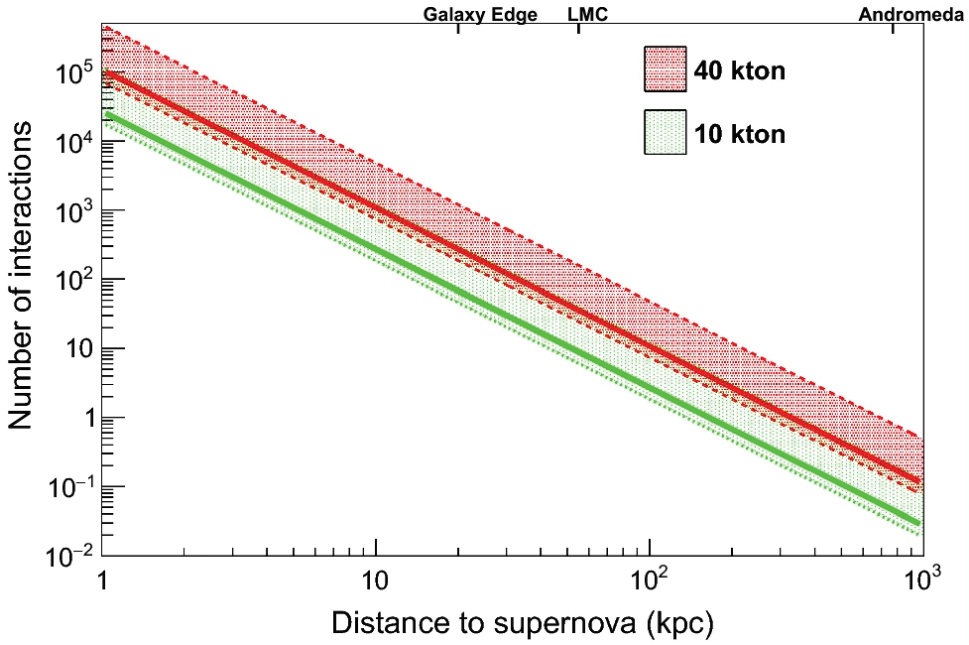


FIGURE 3.18: Number of interactions detected in DUNE as a function of the distance of the Supernova

On Fig 3.18, the solid lines represent the number of interactions for the Garching model and the colored areas represent what the different models could expect as a function of the distance of the Supernova. The ideal case would be a Supernova in the galaxy for which of the order of a thousand events could be observed in the detector.

The number of interactions is distributed as shown in the figure at the left (Fig. 3.19) over the 10 seconds after the Supernova explosion. The evolution of the number of events reflects the different phases of the core collapse mechanism, as predicted by the model. Real measurements will allow a refinement of the models. The figure at the right shows the energy distribution of the neutrinos that will be observed in the detector, where the average energy value of these neutrinos will be around 15MeV. These two figures correspond to a Supernova at 10kpc distance simulated with the Garching model.

The interactions of these neutrinos will therefore occur in the detector and their identification will be necessary in order to have informations on the flux. Each interaction channel should have significantly different final states and tracks:

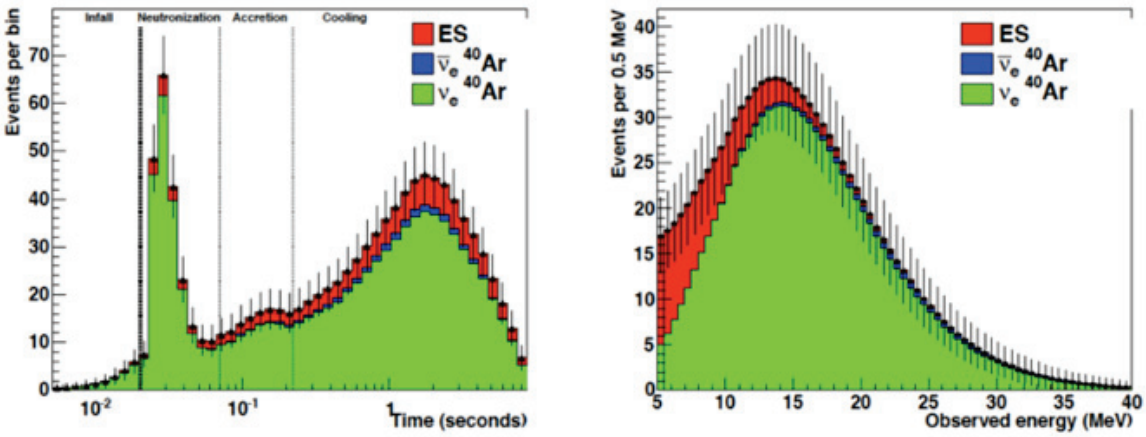


FIGURE 3.19: Time and energy distributions of the Supernova neutrino interactions in DUNE for a Supernova at 10kpc distance

- CC : Track of an electron and nuclear de-excitation photons (emission of a nucleon also possible)
- NC : De-excitation gammas at different wavelengths than in CC interactions
- ES : Track of an electron without gammas

Because the energy of the neutrinos to be observed in this case is relatively low, the resulting tracks of the electrons produced in the interactions should be contained within a few centimeters in liquid argon. Figure 3.20 illustrates a ν_e CC interaction with $E_\nu = 30.25 MeV$.

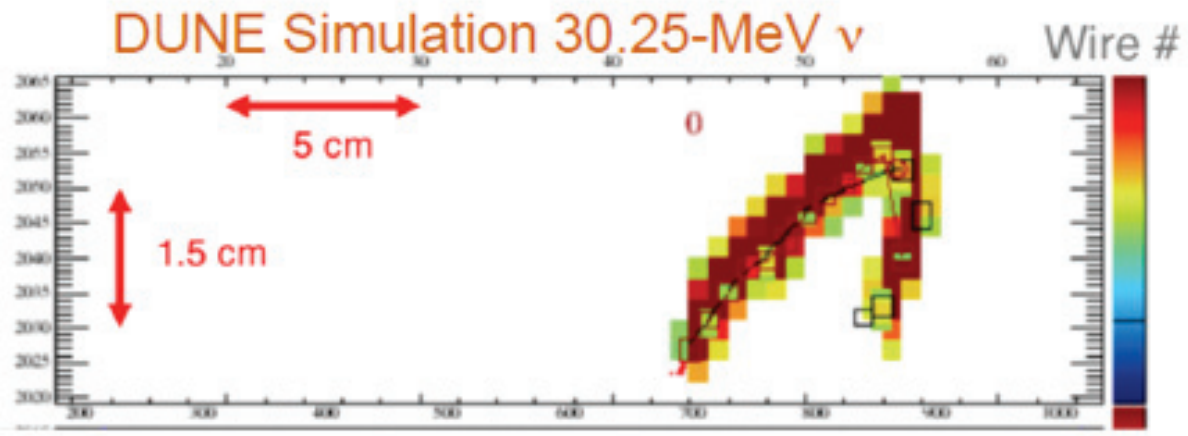


FIGURE 3.20: Simulated interaction of a Supernova Neutrino in DUNE

The detection of Supernova neutrinos is therefore a challenge because the energy deposited in the detector is low and the neutrino emissions happens over a long time period, which is not known in advance. This detection is made even more difficult by natural Argon radioactivity due to the ^{39}Ar isotope, which has an activity of 1Bq/kg and will generate a constant background which can be confused in a first instance with interactions of Supernova neutrinos, although the energy of the electrons from these decays (decay endpoint about 0.6 MeV) is lower than the one of Supernova neutrinos.

Chapter 4

The protoDUNE/DUNE dual-phase charge readout systems

4.1 The protoDUNE dual-phase charge readout system

The dual-phase charge readout is the outcome of an R&D process started in 2006 at IP2I Lyon and aimed at optimizing cost and performance. It implements low noise cryogenic analog amplifiers that connect to external front digital warm electronics through the signal feedthrough chimneys in the cryostat roof. A general description of the concept is presented in Figure 4.1.

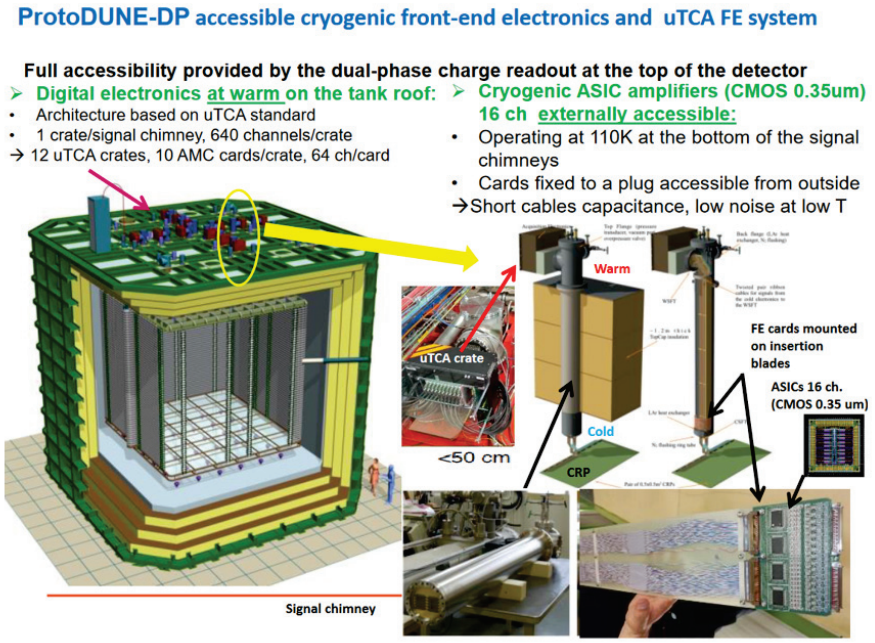


FIGURE 4.1: Schematic layout of the protoDUNE dual-phase charge readout system

The cryogenic ASIC amplifiers are located on analog front-end cards that are mounted on extractable blades, allowing servicing or replacement from the cryostat roof via a hot swapping procedure. Each card plugs into a cold flange at the bottom of a chimney capable of hosting 10 cards. The chimneys, hosting the sliding blades with the front-end cards, penetrate the cryostat insulation down to have their bottom flanges at a few tens centimeters distance from the CRP anodes.

The chimney cold flanges define the charge readout electronics interface to the Charge Readout Planes (CRP). Flat cables going to the CRP anode connectors are plugged into the bottom side of the cold flanges inside the cryostat.

The external warm flanges on a chimney connect via VHDCI cables, the amplified signals to the front-end digitization system, which is housed in μTCA crates located on the cryostat roof, close to the chimneys warm flanges (see Fig 4.2).

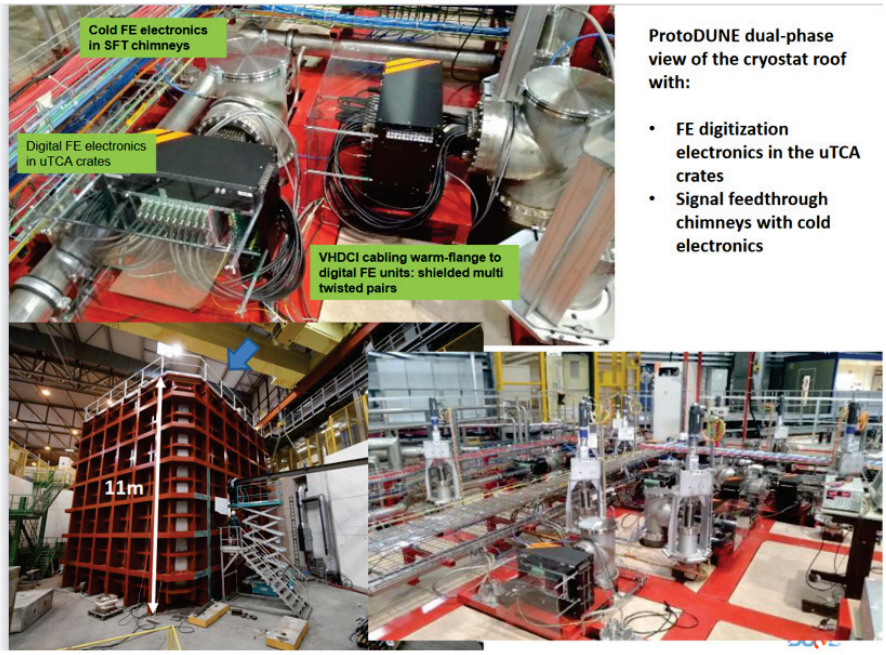


FIGURE 4.2: Pictures of the μTCA digitization system on the cryostat roof of ProtoDUNE dual-phase

Installing the digital electronics on the cryostat roof allows exploiting electronics based on the μTCA standard commonly used in the telecommunications industry.

There is a clear environmental separation between the digital and the analog electronics which is completely shielded from the environment by the metallic structure of the chimney and operates at cryogenic temperatures around 110K. Each analog front-end card has 64 readout channels.

Each μTCA crate (see Fig 4.3) can host up to 12 charge readout AMC front-end digitization units of 64 channels each but in the protoDUNE dual-phase implementation hosts 10 AMC cards. The signal sampling frequency exploited in protoDUNE dual-phase is 2.5 MHz.

Each μTCA crate has 10 Gb/s connectivity from its MCH, with optical fiber Ethernet links to the DAQ back-end system.

The front-end digitization system is coupled to a timing distribution system based on the White Rabbit standard.

4.2 The protoDUNE dual-phase analog cryogenic electronics

The cryogenic amplifier ASIC LARZIC is the principal component of the front-end analog cards.

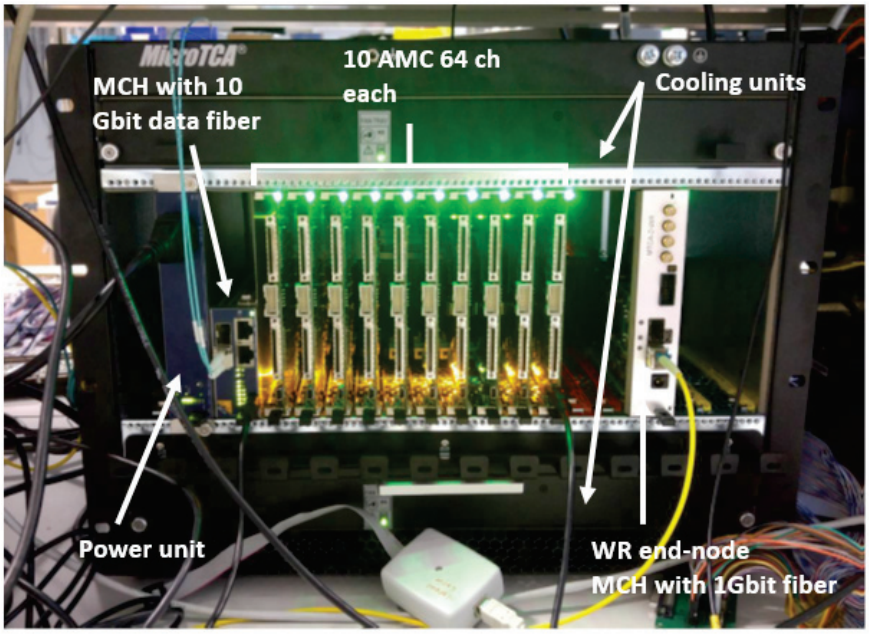


FIGURE 4.3: Picture of a μTCA crate of the protoDUNE dual-phase digitization system

Its design is based on CMOS 0.35 μm technology for which R&D began in 2006. The chips are produced with CMP[1] runs at the AMS Full Service Foundry [2].

This technology remains fully exploitable for production on the timescale of the DUNE far detector modules.

Each ASIC chip contains 16 amplifier channels with differential line buffers and has a power consumption of 11 mW/channel.

Each cryogenic front-end card holds four amplifier ASIC chips and a few passive discrete components (see figure 4.4). The front-end cards have decoupling capacitors and biasing resistors on board in order to bias the dual-phase CRP anodes. These components are rated for operation at several kV.

The front-end cards host as well ESD protection components with diodes pairs, which were selected to withstand, among other possible protection components with a dedicated R&D programm, LEM discharges at 3 kV .

The front-end cards amplify bipolar signals that are propagated with differential analog lines to the digitization system located in the μTCA crates on the cryostat roof.

These cards were operating in protoDUNE dual-phase with a typical intrinsic noise around 600 electrons. On top of this intrinsic good performance of the front-end electronics, overall noise levels in protoDUNE dual-phase were dominated by environmental conditions inside the cryostat (pickup of EMI signals by the CRP strips acting as antennas) related to grounding imperfections of the slow-control connections at their dedicated cryostat flanges (LEM and grids HV connections, temperature probes, cameras, CRP instrumentation). Imperfect grounding at the cryostat flanges of these service connections resulted in injecting noise inside the cryostat radiated by these cables and then picked-up by the strips. A careful mitigation action based on improving the grounding of these coaxial cables at the level of the flanges largely reduced to minimal levels this external noise.

[1] Circuits Multi-Projets®, `https://mycmp.fr/`
[2] ams®, `https://ams.com/full-service-foundry`

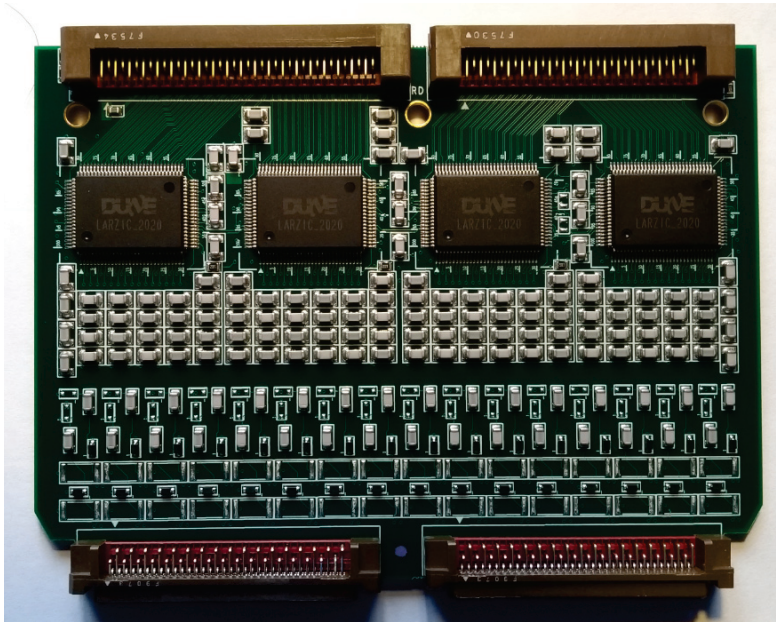


FIGURE 4.4: A front-end cryogenic amplifier card with 4 cryogenic ASIC Larzic chips

The front-end cards are mounted on sliding blades that can be inserted into or extracted from the chimney (see Figure 4.5).

The chimney, ultra-high-vacuum-sealed on the top and bottom, enables access to the front-end analog electronics for repair or replacement while the detector is operational, without affecting the inner cryostat volume. This capability was well demonstrated during the operation of protoDUNE dual-phase (see Figure 4.6).

In addition, the metallic structure of the chimney acts as a Faraday cage, isolating the analog front-end cards from environmental interference.

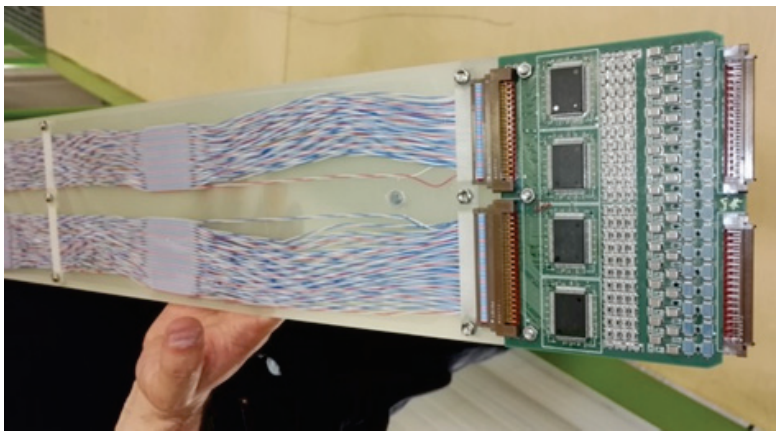


FIGURE 4.5: A front-end cryogenic amplifier card mounted on a signal feedthrough chimney sliding blade

The vacuum-tight feedthrough flanges at both ends of the chimney dispatch the signal and slow control lines to and from the front-end electronics inside it. The bottom (cold) feedthrough flange isolates the inner volume of the detector from the chimney volume and interconnects the signals from the CRP to the front-end analog FE cards.

FIGURE 4.6: Access to a front-end cryogenic amplifier card in a signal feed-through chimney

The (warm) feedthrough flange at the top seals the chimney from the outside environment, passes the low-voltage and control lines to the enclosed front-end electronics and feeds out the differential analog signal lines from the FE amplifiers.

A low voltage power supply coupled to a filtering/distribution unit is located on the cryostat roof and distributes to the warm flanges, by using 10m long multiwire shielded cables. Five different voltages are needed for the operation of the ASICs.

4.3 The protoDUNE dual-phase μTCA digitization system

The warm digital electronics, located on the cryostat roof, digitizes the analog signals coming from the chimneys and transmits them to the DAQ back-end system.

Each μTCA crate can host up to a maximum of 12 charge readout Advanced Mezzanine Cards (AMC), which are in charge of the signals digitization. The configuration implemented in protoDUNE dual-phase deploys only ten AMCs per crate.

Each AMC card (see Figure 4.7) has eight Analog Devices AD92574[3] ADC chips, two dual-port memories, (Integrated Device Technology IDT70T33395), and a FPGA (Altera Cyclone V[4]) on board.

The FPGA provides a NIOS II virtual processor that handles the readout and data transmission. A complete synoptic of the cards functionalities can be seen in figure 4.8.

Given the programming flexibility provided by the firmware, which can be flashed to all AMCs via the crate network connection, this digital front-end stage offers the possibility to operate in different configurations including the one where data are losslessly compressed before transmitting them over the network.

The components were chosen and sized to meet functional design requirements and with the goal, kept since the beginning of the R&D process in 2006, to be able to reduce costs for large scale production.

[3]Analog Devices, `www.analog.com`

[4]Altera Cyclone®, `https://www.intel.com/content/www/us/en/programmable/b/cyclone-v.html`

FIGURE 4.7: Picture of a AMC digitization card

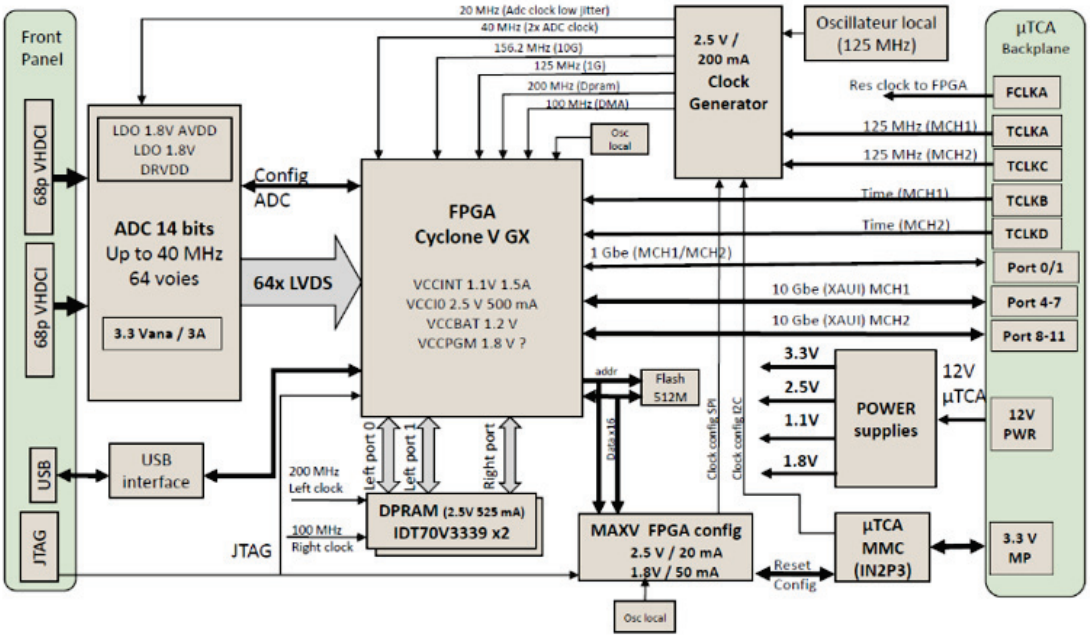


FIGURE 4.8: Synoptic of a AMC digitization card

The AMC generates a continuous stream of data for each readout channel at 2.5 MHz sampling rate and 12 bits which are written in the dual-port memory. However these data output conditions are deliberately downgraded for this application from the real acquisition condition at the level of the ADCs. The ADCs have 14 bits and operate at a factor 10 higher sampling frequency. Each AMC has 64 channels and digitizes the signals coming from a single analog front-end card. The AMCs can acquire drift windows based on external triggers or can continuously stream all the sampled data to the DAQ system.

Each AMC card has 10 Gb/s individual connectivity to the backplane of the μTCA crate. Each μTCA crate contains a network switch in its MCH, through which data collected from the hosted AMC cards are sent to the DAQ back-end system. The timing synchronization of the AMCs is achieved via a White-Rabbit MCH module (also housed in the μTCA crate in the slot for the second MCH). The White-Rabbit MCH, connected to a standard White-Rabbit network, ensures the timing and triggers distribution to the digital front-end units.

The MCH and White-Rabbit MCH require one optical fiber link each, respectively operating at 10 Gb/s and 1 Gb/s.

Having been designed as an experiment operating with external beam triggers, the operation mode of the AMCs in protoDUNE dual-phase has always been based on external triggers.

In this case the AMC firmware selects in the dual-port memory a drift window of 10k samples starting exactly at the time of the electrical signal related to the external trigger. This signal is time-stamped by a dedicated node on the White-Rabbit timing system. The data of this time-stamp are then transmitted received by all AMCs still via the White-Rabbit network via a dedicated packet. They are decoded by the AMC and the time-stamp of the trigger and can be then compared to the time of the acquired digitization samples stored in the memory of the AMC. The time of the sampling being expressed on the same timing base of the White-Rabbit network.

Two different firmware versions were deployed in protoDUNE dual-phase: the standard one for the acquisition on external triggers and data transmission without compression and a second version which was including online lossless Huffman compression before data transmission.

4.4 The protoDUNE dual-phase timing system

The timing distribution at the level of the front-end digitization units uses a White-Rabbit network that combines the synchronous 1 Gb/s Ethernet (SyncE) technology with the exchange of PTP(V2) packets to synchronize clocks of distant nodes to a common time, while automatically compensating for the propagation delays. This idea was also part of the R&D process started at IP2I in 2006.

The White-Rabbit (WR) system is fed by a high-stability GPS disciplined oscillator GPSDO, providing a clock reference signal to be distributed over the physical layer interface of the White-Rabbit Ethernet network.

The network topology uses specially designed switches that have standard IEEE802.1x Ethernet bridge functionality with additional WR-specific extensions to preserve clock accuracy. Time and frequency information is distributed to the nodes on the WR network via optical fibers.

The WR protocol automatically performs dynamic self-calibrations to account for any propagation delays and keeps all connected nodes continuously synchronized to sub-nanosecond precision independently of the network topology and changes in environmental conditions.

The WR switch act as Grand Master of the WR network and it is connected via 1 Gb/s optical links through a cascade of secondary WR switches to the WR end-node slave cards present within each μTCA crate (WR-MCH), keeping these synchronized to its reference time (see figure 4.9).

This card includes a commercial White-Rabbit end-node (WRLEN) inserted on a dedicated card developed at IP2I in order to interface it to the μTCA backplane via the dedicated TCLKC and TCLKD lines reserved to the second MCH slot. The WRLEN works as a standard White-Rabbit end-node and it is able to align itself on the WR network.

The firmware of the WRLEN has also been re-written, adding on top of the standard WR core, the functionalities to include the generation of specific timing and trigger frames to be propagated, together with the WR 125MHz clock, on the dedicated lines of the μTCA crate backplane.

The WRLEN distributes the 125MHz WR clock on the back-plane of the crate via TCLKC. It produces and distributes every ms synchronization frames to the AMCs on the back-plane of the crate via TCLKD. The WRLEN also decodes the external triggers timestamps sent to the entire WR network

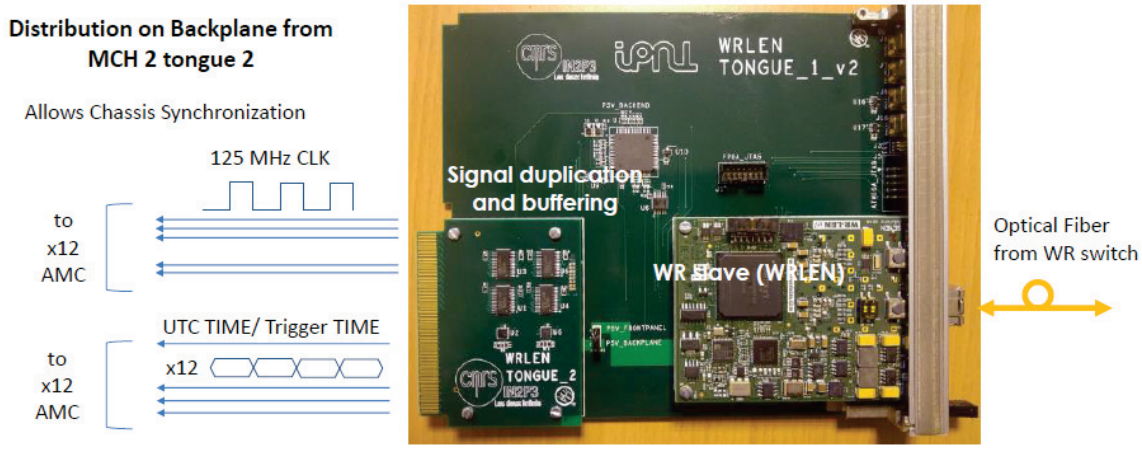


FIGURE 4.9: White-Rabbit MCH card

as ethernet packets. Once these are received by the WRLEN they are also distributed, in a similar format as the timing frames, via the back-plane of the crate to the AMCs.

At the level of the AMCs a specific firmware deals with the exploitation of the clock transmitted via TLCKC and the treatment of the timing/trigger frames arriving via TCLKD.

Time-stamping of the electrical signals of the trigger is performed with a standard White-Rabbit DIO card. The data of these time-stamps can be transmitted via the White-Rabbit network itself to all end-nodes.

White-Rabbit is a 1 Gb synchronous Ethernet network where the clock is replaced by a very precise clock with atomic standards. The WR network itself has very little occupancy, just needed to exchange a few PTP packets for the synchronization and determinations of delays. So there is very little impact in using the same network to transmit a few other packets. The transmission of the packets containing the time-stamps in TAI of the external trigger corresponds to trigger rates up to 100Hz, so it also generates very little network bandwidth.

From the instant of the time-stamping of the electrical signal of the trigger to the time the time-stamp information reaches the AMC via the White-Rabbit network the typical latency which is below 200 us.

This time corresponds to about 500 samples and it is largely compensated by the memory buffer of the AMCs while the acquired drift windows include 10k samples. This timestamp distribution mechanism allows then acquiring "a posteriori" drift windows with the first sample aligned with zero delay with respect to the external triggers.

4.5 The protoDUNE dual-phase DAQ back-end system

The NP02/protoDUNE dual-phase DAQ system is a Ethernet network based DAQ system which can acquire data at very high bandwidth (up to 20 GB/s). A global sketch of this system is presented in figure 4.10.

Even if this system has some differences, as it will be shown later, with respect to the DUNE Far Detector DAQ system, by working only on the basis of external triggers and not having to find out by itself the trigger conditions, it will be described in detail in this section since it may constitute an illustrative example of the computing and network hardware which is needed in order to setup a back-end system for a DUNE dual-phase module.

One could see the DUNE system as something using similar computing and network elements but of which the system of protoDUNE dual-phase represents one out of the 20 modules working in parallel, eventually needed for DUNE. The storage system is already of the size of what is needed for DUNE, since even if a DUNE module is 20 times larger than protoDUNE dual-phase, the data volume corresponding to the rare interesting events to be written on disk is not very different than the one of protoDUNE which was designed to operate at quite high trigger rate of 100 Hz.

The main difference with respect to the DUNE implementation is that in protoDUNE dual-phase the event builders host standard network cards while in the DUNE case the equivalent machines, "back-end nodes", would interface to the crates by using FPGA network cards with several optical links per card. These FPGA network cards have been exploited and studied in this thesis work. The FPGA network cards will take care of analyzing the data received from the crates on real time and producing the trigger primitives while the equivalent machines to the L1 event builders will keep the data in their RAM memories while waiting for a trigger decision.

Another important difference with respect to the DUNE architecture is then the absence in the protoDUNE dual-phase case of a server receiving the trigger primitives from the L1 event builders and orchestrating a trigger decision in order to write the data on the EOS storage servers.

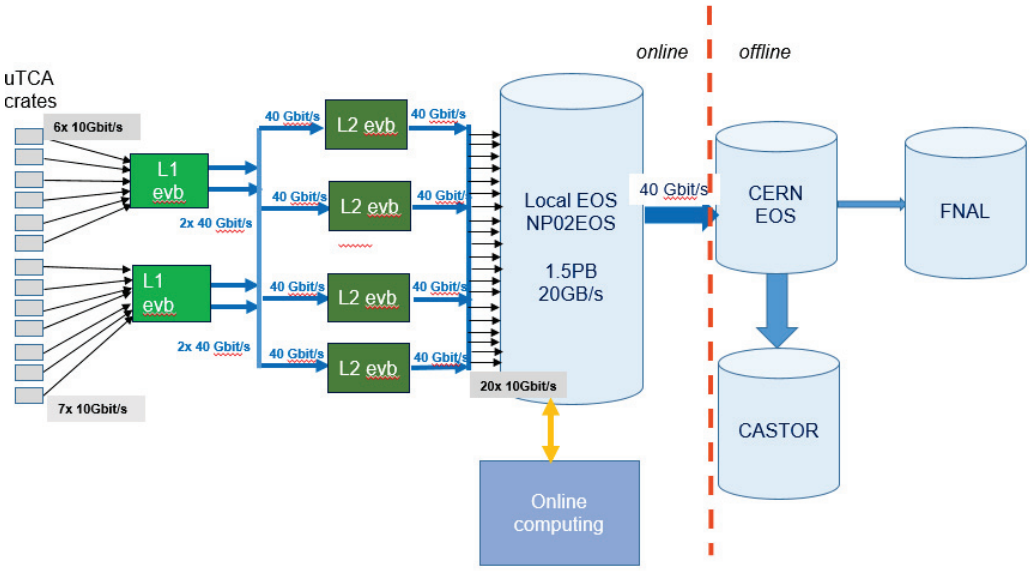


FIGURE 4.10: Sketch of the protoDUNE dual-phase DAQ system

In protoDUNE dual-phase the front-end digitization units (AMCs) are contained in μTCA crates. Each charge readout crate, located in front of the corresponding signal feedthrough chimney on the cryostat roof can include up to 10 AMC reading each 64 channels for a total of 640 channels per crate digitized at 2.5 MHz.

A μTCA crate is a 10 Gb/s network system connecting the AMC with its own switch included in the crate controller (MCH). The MCH of each crate is connected with a dedicated 10 Gb/s to a Level 1 (L1) event builder.

Two L1 event builders are used to read several crates corresponding to a detector half. The L1 event builders are connected via several links on a high speed network at 40 Gb/s to the Level 2 (L2) event builders and to a high bandwidth distributed storage system (EOS), the network infrastructure ensures total 20 GB/s bandwidth. Details of the network infrastructure are presented in figure 4.11.

Each L1 event builder puts together the data, corresponding to the drift window starting with the trigger timestamp, acquired from the connected crates to build an event half on its ramdisk. The L1

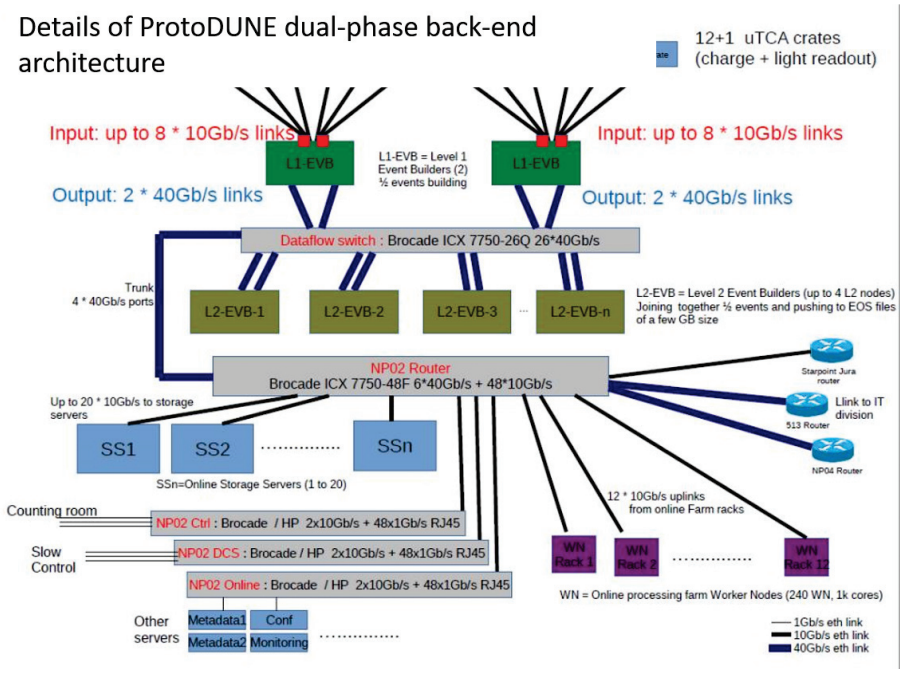


FIGURE 4.11: Sketch of the protoDUNE dual-phase DAQ system

ramdisks are visible via the network to the L2 event builders who assemble together the two event halves in the final event format and assemble on their own ramdisks the events in 3 GB files which are then pushed through the EOS high bandwidth storage system which can absorb up to 20 GB/s data writing on disk. Four L2 event builders work in parallel by sharing evenly the events produced by the L1 event builders and producing the final 3 GB data files to be written on disk.

A trigger server (see figure 4.12) handles the white-rabbit time-stamping of external trigger signals (beam counters, cosmic counters, PMTs trigger, calibration triggers) and the transmission of these timestamps to the AMCs via the white-rabbit network and of the trigger information to the L1 event builders via a dedicated Ethernet network.

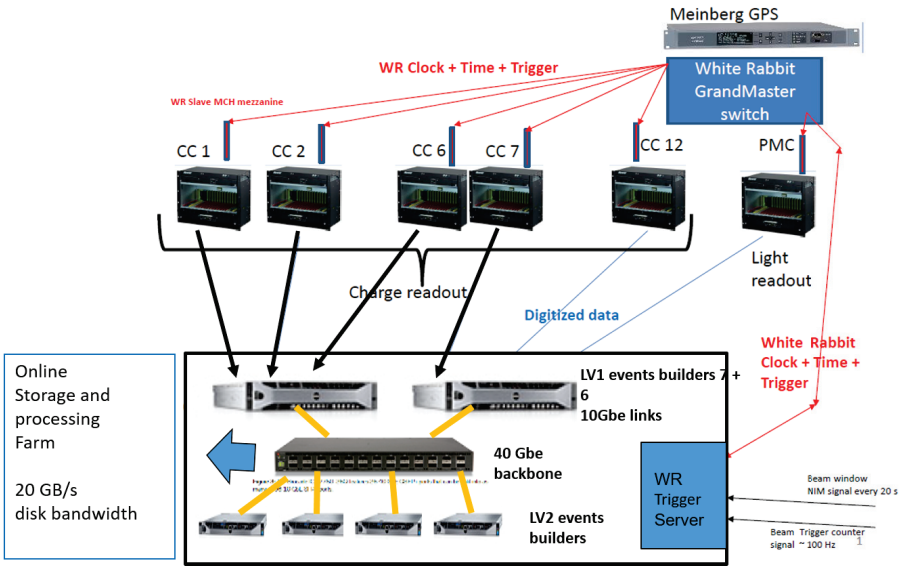


FIGURE 4.12: Integration of the DAQ and timing systems

The White-Rabbit network ensures also the timing and synchronization of the AMC digitization units. The run control interface ensures the control and monitoring of all the components of the system in

order to start and stop runs and transfer the data to the local EOS and to the final storage for offline exploitation.

A picture of the NP02/protoDUNE dual-phase DAQ back-end system in the dedicated DAQ room is shown in figure 4.13.

FIGURE 4.13: Integration of the DAQ and timing systems

The online storage and processing facility network architecture includes a back-end network infrastructure based on a 40 Gb/s dedicated switch (Brocade ICX7750-26Q) connected to a 40/10 Gb/s router (Brocade ICX 7750-48F).

A dedicated 10 Gb/s multi-fibers network connects to the μTCA crates and a dedicated trigger network is used for the communication between the trigger server and the two event builders. Two links operating at 40 Gb/s ensure the connectivity of the NP02 DAQ system to the CERN computing center (IT division).

The storage servers are constituted by 20 machine DELL R510, hosting 72 TB per machine for 1.44 PB total disk space for 20 machines. Each storage server has 10 Gb/s network connectivity.

The DAQ cluster, in addition to the two LV1 event builders (DELL R730 384 GB RAM) plus the four LV2 event builders (DELL R730 192 GB RAM), includes several service machines (9 DELL Poweredge R610 units): 2 EOS metadata servers, a configuration server, an online processing server, a batch management server, a control server, etc ...

This configuration is completed by an online computing farm allowing to reconstruct (with a fast tracks reconstruction algorithm) and analyse on real time the data taken by the DAQ back-end system. This farm is based on 40 servers Poweredge C6200 for a total of 450 cores.

Automatic file transfer systems transfer the data from the local EOS system to the CERN IT division and Fermilab.

4.6 Operation mode of the dual-phase DAQ system

As previously mentioned, the protoDUNE dual-phase DAQ system has been operating in 2019-2020 in external trigger mode. The AMC cards were continuously sampling the analog electric signals and storing the samples in their dual-port memories. The sampling of all AMCs is precisely aligned with the White-Rabbit timing distribution system which is used to distribute a precise clock and synchronization frames to all AMCs. The AMC knows the absolute time in TAI of the samples. When a trigger frame arrives, the time-stamp of the trigger is used to select the first sample of the drift window. Data of all channels corresponding to consecutive 10k samples are then selected and formatted in UDP packets to be transmitted to the back-end system via the ethernet network. This mechanism allows operating with different external triggers (beam counters triggers, triggers from the photo-detection system, triggers provided by cosmic ray counters, randomly generated triggers) and to acquire the corresponding 10k samples drift windows which correspond to a drift time of 4ms needed for a maximal path of 6m at the nominal drift field of 500V/cm.

The maximal trigger rate is not limited by the connectivity of the AMC card itself but it is limited by the bandwidth of the ethernet link of the MCH in the μTCA crate itself to the DAQ back-end (10 Gb/s for the MCH used for protoDUNE dual-phase). A DAQ scheme where the AMCs in the same crate output the data one by one in succession by saturating the 10 Gb/s MCH link allows reaching 50Hz maximal trigger rate. The continuous streaming of all acquired samples by the 10 cards reading 640 channels in the same crate would require a total network bandwidth of 18 Gb/s, not supported by the standard MCH operating at 10 Gb/s.

The AMCs can also compress the data without losses by using an optimized version of the Huffman algorithm. The firmware operating in the AMC is more complex since it has to compress the data of the drift window before formatting them in ethernet packets and transmitting them via through the network. This firmware version including lossless compression was operated in ProtoDUNE dual-phase in the summer 2020. The lossless compression had as goal a compression rate of 10. This factor 10 allows exploiting the 10 Gb/s links for higher trigger rates than 50 Hz, reaching also continuous data streaming conditions (which, given an acquisition window of 4 ms, could be expressed in an equivalent trigger rate of 250 Hz) with a safety margin of about a factor five.

The performance of the compression algorithm is based on the stability of the pedestals and the fact that with the dual-phase electronics the noise is of typically less than 1 ADC count. It has also to be taken into account that the occupancy of the detector is extremely low and that most of the acquired data just consist in pedestals affected by the electronics noise. So this configuration, in absence of presence of physical signals, results in a baseline which is essentially constant with small variations of 1 ADC count.

Being the ADC dynamics of 14 bit while only the 12 most significant bits used, the pedestals values can be finer adjusted in order to be optimally placed at the center of the bit for the 12 bit range. A sophisticated procedure to measure the pedestals and adjust optimally their offsets was put in place in the DAQ system and the AMC firmware incorporating the Huffman compression. This system was tested in protoDUNE dual-phase in summer 2020.

It was measured that when a run is started after the pedestals optimization procedure the compression factor is indeed around 10. However the pedestals over 14 bit dynamics have some small thermal effects related to the temperature of the AMCs in the μTCA crate. Although when looking at it with 12 bits dynamics the position of the pedestals keeps constant, these thermal effects move the real position of the pedestal with respect to the center of the bit in the 12 bits dynamics and affect the compression performance.

Overall by taking into account these effects and the noise the measured performance of the Huffman algorithm was corresponding to a factor 5 instead than a factor 10. This figure, although not corresponding to the initial goal is still satisfying the requirements for DUNE dual-phase in order to exploit the bandwidth of the 10 Gb/s links for continuous streaming, with a safety margin factor of 2.5 instead than 5.

In summary protoDUNE dual-phase has been operating in external trigger mode by using two different versions of the AMC firmware without and with lossless compression. The compression operation mode, which enables higher trigger rates than 50 Hz, was successfully tested in view of being used for the DUNE 10 kton module in order to support continuous data streaming by keeping the 10 Gb/s bandwidth going from the MCHs to the DAQ back-end system.

4.7 The DUNE DAQ system and corresponding developments of the dual-phase electronics

The charge readout for the underground operation of the dual-phase 10 kton module is based on the same front-end hardware developed for protoDUNE dual-phase but operating in a continuous data streaming from all the AMCs.

This operation mode, as described in the previous DUNE chapter is needed since the occupancy of the detector is very low and the DAQ back-end system must be able to define by itself triggers related to the activity present in the detector, like cosmic rays, proton decay candidates or bursts of SuperNova neutrinos. Beam triggers are treated in a similar way, however the knowledge of the beam extraction timing which can be transmitted trough the network from Fermilab to SURF could allow the DAQ back-end extracting precisely the corresponding drift window from the continuously streamed data, independently on activity present in the detector. The simultaneous presence of activity in the detector and the beam timing information allow extracting the interesting drift windows in time with the beam and associating them to the beam activity.

The global DAQ layout of the 10 kton dual-phase module, extracted from the dual-phase TDR is presented in Figure 4.14.

The DUNE dual-phase DAQ back-end system, as described in the TDR of the dual-phase module, similarly as for a single-phase module includes a set of event building and trigger machines, high-performance network elements, and a high-bandwidth distributed storage system based on an array of storage servers operating in parallel.

In particular, the DAQ back-end system is expected to :

- Collect the high-bandwidth data volume coming from the data links of the FE digitization crates;

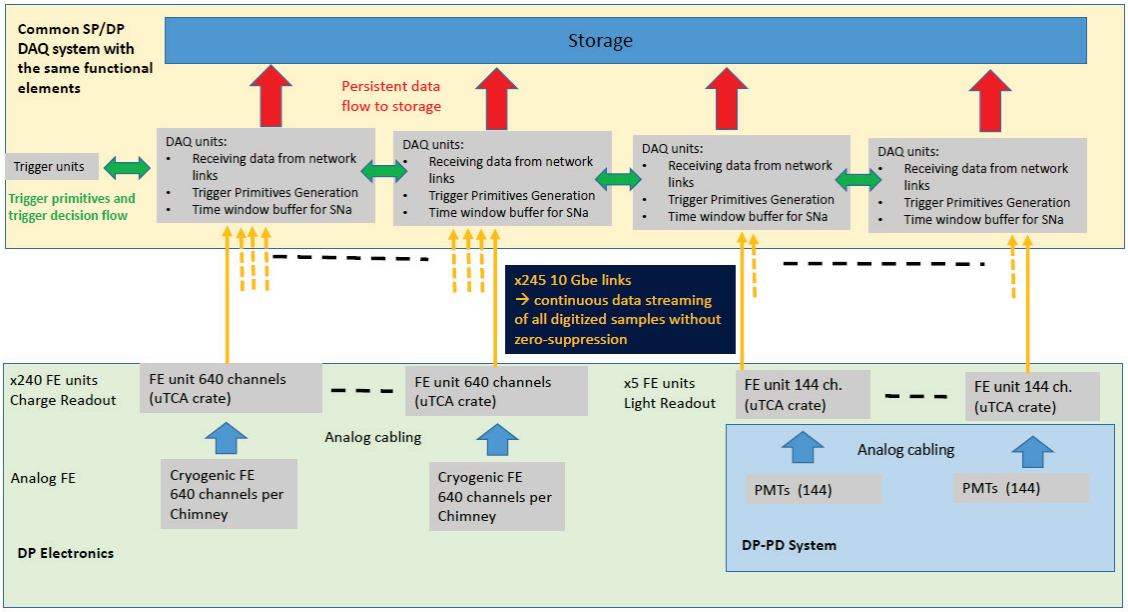


FIGURE 4.14: Organization of the DAQ for the DUNE dual-phase module

- Aggregate the data streams from different crates in ROI (Region Of Interest) or over the entire detector volume (A ROI is typically the size of a ProtoDUNE-DP four-CRP surface because events are contained in such a region.);
- Process this data flow using an online trigger farm as a prelude to selecting relevant events to be recorded on disk (both neutrino beam and off-beam events)
- Produce charge-readout triggers independently of the light-readout triggers and beam-spill information. (For SNB events in particular, the trigger farm would issue triggers over a sliding timing window of approximately 10 seconds based on the presence of low-energy depositions; the entire content would be dumped to disk.)

Continuous data streaming from the AMC cards in the μTCA crates can be implemented both with lossless compression or without compression. The use of 10 Gb/s links at the level of the MCHs implies the operation with compression. Compression can be avoided if the MCH is replaced with a compatible one supporting higher bandwidth. A development for MCHs with 40 Gb/s, allowing to operate without compression, was carried on in parallel in collaboration with NAT and it is reported in the next section. In both cases (compression or no compression) continuous streaming implies a simplified working scheme and firmware than the external trigger mode.

In the external trigger mode (protoDUNE dual-phase) drift windows of consecutive 10k samples are selected in the AMC dual-port memory by using the timestamp of the external trigger transmitted to the AMC by the white-rabbit system. Data are then formatted in UDP packets and transmitted to the back-end over the ethernet network. In no compression mode each channel occupies 3 UDP packets and the data of the 64 channels are transmitted in sequence by each AMC. The formatting of the data to be transmitted for the compressed mode is more complex, given the dynamic size of the data, however the data are still organized in UDP packets.

In continuous streaming mode (DUNE dual-phase) the principle of front-end digitization and DAQ is the same, however it is implemented in a simpler mode. Data formatting and data transmission are performed following a similar scheme as in trigger mode. However, the transmission of drift windows is no more regulated by the reception of external timestamps but simply drift windows are transmitted automatically one after the other with no time gaps in the sampling.

Since all data are transmitted anyway, the size of the drift windows in terms of number of samples can be optimized in order to match the size of the UDP packets. This is in particular effective in no-compression mode, where an optimal fixed number of transmitted samples is fixed per drift window. The way the packets are transmitted is similar as in external trigger mode. Each consecutive drift window has embedded in the UDP packet data format a header containing the time-stamp information of its first sample, together with other necessary information to reconstruct the data in the back-end system, such as the AMC number, crate number, channel number etc.

The work described in this thesis is not focused on the front-end and what happens in the AMC digitization units, for which a valid scheme had been already tested, but on the treatment of the data at the level of the interface to the back-end system.

Namely:

- If data are transmitted compressed, in order to fit within the bandwidth of the 10 Gb/s links, the DAQ back-end will have to decompress on the fly the data in order to examine them to check if there is any activity in the detector and contribute to making an internal trigger decision. The first step for a trigger decision is the assessment of the detector activity at single channel level (search for trigger primitives), which must be pipelined on real time with the data decompression. Which is the best scheme to implement this mechanism ? Is it possible to implement real time decompression and search for trigger primitives in FPGA network cards ?
- An alternative scheme, by having the availability of higher bandwidth MHC is to transmit data without compression. This eliminates the need for data decompression, however the search for trigger primitives must be still performed and possibly in FPGA network cards.

In all cases data must be stored for a long window, of at least 10s, in the memory of the DAQ back-end nodes waiting for a trigger decision. In case of compressed data, these data can be directly kept in the memory of the DAQ machines waiting to be stored on disk after the trigger decision arrives. In this case it is not needed to store the uncompressed data produced for the analysis of the trigger primitives, these data can be eliminated after the trigger primitives search. In the scheme based on the transmission from the front-end of uncompressed data, one has to evaluate the opportunity of compressing the data before permanent storage in order to optimize the storage bandwidth.

4.8 Developments for the Vertical Drift top-drift charge readout design

This thesis is based on the developments initially carried on for the dual-phase 10 kton detector. As described in the DUNE chapter, during last year the dual-phase design has been evolving to the Vertical Drift. All the front-end electronics developed for the dual-phase is foreseen to be used for the readout of the top-drift volume of the vertical drift volume with very minor modifications as the optimization of the ADC dynamics for possibly smaller (vertical drift has no gain in the gas phase) and bipolar signals. Meanwhile the development on the 40 Gb/s MCH had also a remarkable progress and it is known today that the 10 Gb/s MCH can be replaced with these new models working at 40 Gb/s at similar costs. This solution has been extensively and successfully tested in 2021, as mentioned in the next session. The availability of 40Gb/s links, allows for data transmission in continuous streaming without compression at about 50% of the link bandwidth. This possibility simplifies the design and the complications related to compression. In addition, the signal dynamics optimization for Vertical Drift does not limit anymore the noise levels below 1 ADC count, as it was the case of dual-phase, making the compression rate less favorable. For all these reasons the baseline for the Vertical Drift operation is to work in continuous streaming using 40 Gb/s links without data compression. The developments

on the trigger primitives search at the level of the FPGA network cards are still of actuality also for the Vertical Drift design.

4.9 Operation of the charge readout electronics with 40Gb/s MCH

The development based on the 40Gb/s MCH was launched with NAT in 2018. This development reached maturity with the first systems delivered by NAT in spring 2021. They were extensively tested at IP2I (see Fig. 4.15) integrating them with the DUNE top-drift charge readout electronics and the MCH firmware was debugged in collaboration with NAT.

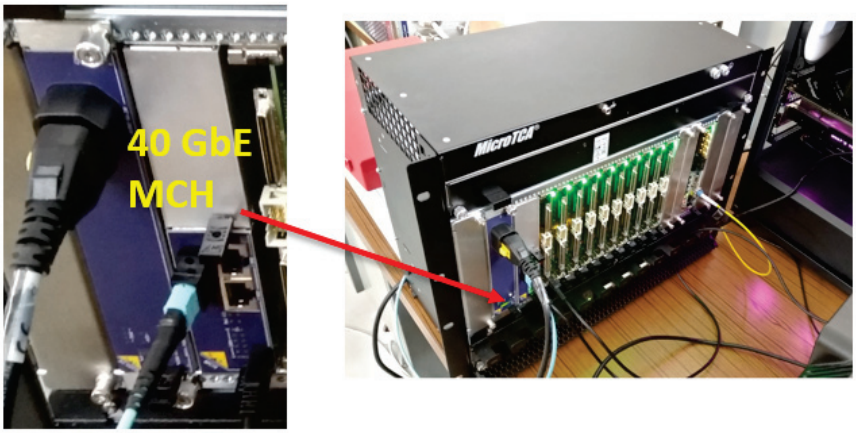


FIGURE 4.15: μTCA system with MCH operating at 40 Gb/s being tested at IP2I in spring 2021 with the VD top-drift electronics readout

These system reached maturity in June 2021 at the level to be integrated in a large scale application system. An intensive test campaign was held at the CERN Neutrino Platform in 2021 for the Vertical Drift detector elements, including the test of the new Vertical Drift Charge Readout Plane design. The new Vertical Drift CRP was successfully and extensively tested in the fall/winter 2021 (see Fig. 4.16) with the both the top (TDE) and bottom (BDE) drift charge readout electronics in a newly installed dedicated cryostat (cold-box).

Three full top-drift μTCA systems with MCH at 40 Gb/s (see Fig. 4.17) were exploited in this test campaign and they demonstrated their full functionality and integration with the top-drift electronics, confirming their application for the baseline design of the top-drift readout of the Vertical Drift DUNE 2nd Far Detector Module.

The cold-box CRP test was preceded by a full integration test, in a specific area of the EHN1 hall at CERN of the top-drift electronics. A dedicated high bandwidth network (at 200 Gb/s to support up to five 40Gb/s μTCA systems, foreseen for the full top-drift CRP in 2022) and DAQ back-end infrastructure was setup as well in order to support the cold-box test of the Vertical Drift CRP and electronics. Some event-display pictures of cosmic ray particles acquired during the cold-box tests with the top-drift electronics are shown in Figure 4.18.

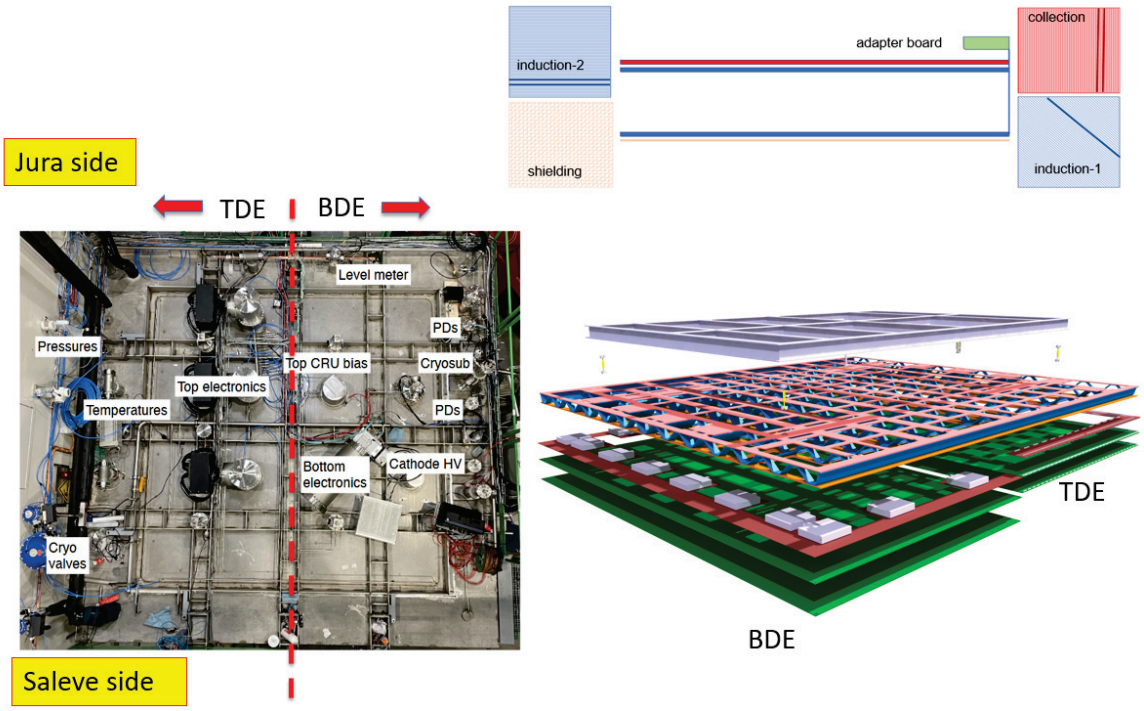


FIGURE 4.16: Cold-box tests at the CERN Neutrino Platform of a Vertical Drift Charge Readout Plane readout with bottom the top-drift (TDE) and bottom-drift (BDE) electronics

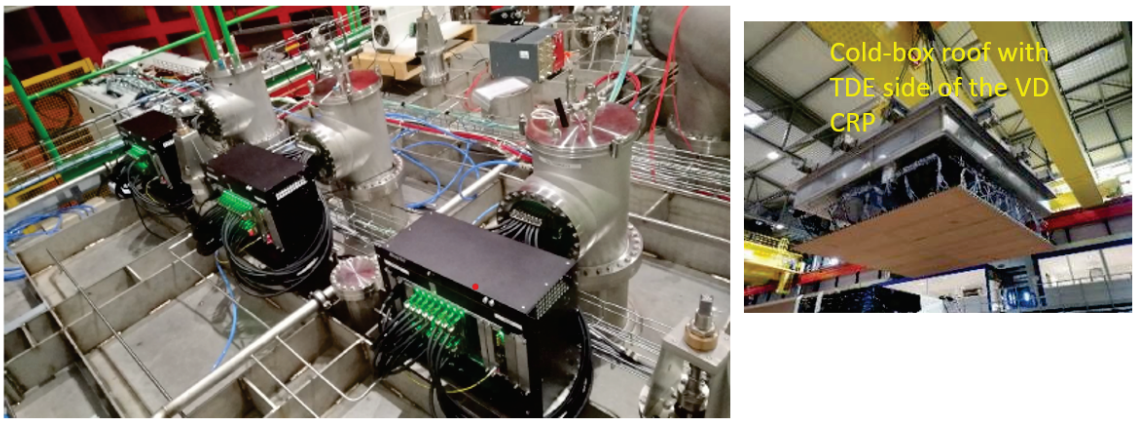


FIGURE 4.17: View of the cold-box roof with the Vertical Drift Charge Readout Plane and three 40Gb/s μTCA systems integrated with the top-drift electronics

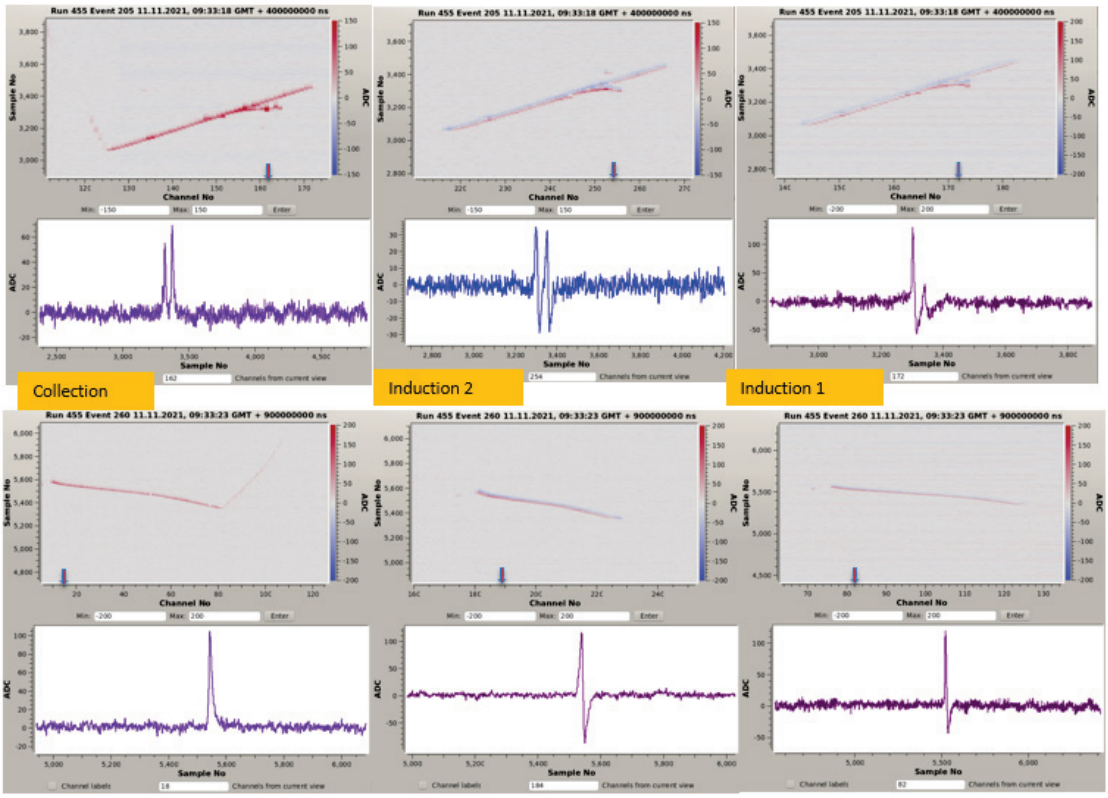


FIGURE 4.18: Event-display pictures of cosmic ray particles acquired during the cold-box tests with the top-drift electronics

Chapter 5

Data compression and decompression

5.1 Huffman compression algorithm

In ProtoDUNE dual-phase the μTCA MCH switches provide 10 Gb/s bandwidth. On the other hand, the maximal data volume that can be generated by a crate equipped with 10 AMCs that digitize signals at 2.5 MHz and 12 bit resolution is 640×12 bits $\times$ 2.5 MHz = 19.2 Gb/s, which is larger by about a factor of 2 than the available bandwidth. In DUNE the acquisition system needs to operate in a continuous streaming mode. Therefore, in the case of 10 Gb/s MCH, the data must be compressed before they exit AMCs. The compression algorithm adopted here is a lossless compression based on Huffman coding. The implementation of Huffman encoding compression [43] in FPGAs for data reduction has already been successfully exploited by the MicroBooNE collaboration [44] to support continuous readout of their LAr TPC detector [38].

The principal idea behind the Huffman coding based compression is to replace the ADC codes of 12 bits representing the data by shorter codes representing the difference between successive ADC samples:

$$\Delta_{\mathrm{ADC}} = \mathrm{ADC}_i - \mathrm{ADC}_{i-1}. \tag{5.1}$$

The differences between ± 3 ADC counts are encoded, while for larger Δ_{ADC} one stores the actual ADC code. To increase compression rate for low noise pedestal baseline, where ADC codes do not vary, a special 1 bit long code, `0b1`, is used to encode four successive ADC samples if the Δ_{ADC} remains the same. This encoding scheme is summarized in Table 5.1. The codes are packed into 13 bit blocks with the first bit reserved for a flag indicating whether the next 12 bits correspond to the actual (flag value `0b0`) ADC code or a sequence (flag value `0b1`) of Huffman codes. The compressed data stream is zero-padded to the nearest two-byte (a 16 bit word) boundary at the end.

ADC difference Δ_{ADC}	Huffman code
0	`0b01`
-1	`0b001`
$+1$	`0b0001`
-2	`0b00001`
$+2$	`0b000001`
-3	`0b0000001`
$+3$	`0b00000001`
Repeated 4 times	`0b1`

TABLE 5.1: Huffman codes for Δ_{ADC}, the difference between the current and previous ADC code.

An example can be taken to illustrate the behavior of this type of compression. The sequence to be encoded will be the following sequence of raw ADC samples: [40, 41, 40, 40, 40, 40, 40, 43, 39, 36] and is represented on Fig 5.1.

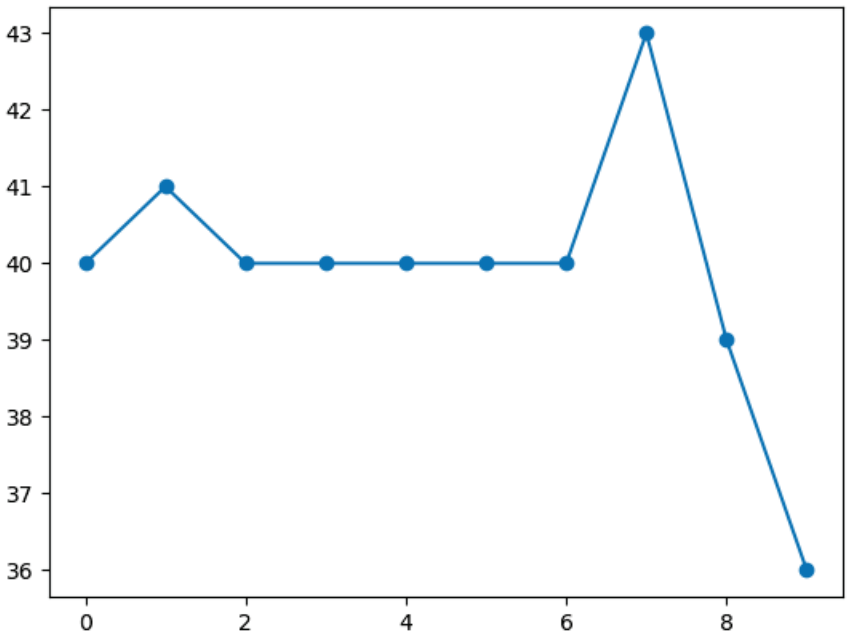


FIGURE 5.1: ADC Sample sequence to encode

The Δ_{ADC} values for this sequence are: [+1, -1, 0, 0, 0, 0, +3, -4, -3]. The first step is then to write the value of the first sample with the corresponding flag of a raw ADC value `0b0`. The first 13 bits word is then:

- 0 0000 0010 1000 - Flag `0b0` / Raw ADC value 40 on 12 bits

After the first value being written, if $-3 \leq \Delta_{\text{ADC}} \leq 3$ the Huffman codes are written after the flag `0b1`. If the difference is greater than |3| and if the previous written codes are not aligned on a 13 bits word, zeros are added. Once the sequence is aligned, a raw ADC value will be written in the same manner as the first one. The rest of the encoded sequence is then:

- 1 0001 0010 1100 - Flag `0b1` / Code +1, -1, 0, 4×0 / Start of the code +3
- 1 0000 0100 0000 - Flag `0b1` / End of the code +3 / Zero padding
- 0 0000 0010 0111 - Flag `0b0` / Raw ADC value 39 on 12 bits
- 1 0000 0010 0000 - Flag `0b1` / Code -3 / Zero padding

The compression factor is primarily determined by how well the noise in the baseline compresses, since most of the data correspond to the pedestals. As Table 5.1 illustrates, the compression is most optimal if the variations are small and one therefore uses the shortest codes to represent ADC values. Conversely if changes in the pedestal (noise) are such that Δ_{ADC} predominantly stays above 3 counts, the compression becomes completely inefficient.

Apart from the noise arriving from the analog front-end cards, the stability of the pedestal also depends on the position of the actual pedestal analog level within the ADC quantization step. If the pedestal

value falls in the middle between two ADC codes, the pedestal would frequently change by ±1 count reducing the overall compressibility of the data. While for the levels close to an ADC digitization step, it would remain nearly constant leading to high compression rates. As will be discussed later in this chapter, the AMC ADC chip (AD92574) allows for small offset adjustments for each channel and this feature can be used to tune the pedestals in order to increase the data compression factors.

5.2 AMC firmware with compression

5.2.1 Firmware

The internal architecture of the firmware with the Huffman compression implemented in the Cyclone V FPGA used in AMCs is presented in Fig 5.2. Each part performs a very precise task in this firmware to be able to process data from 64 channels:

1. Part which manages the trigger and the timestamp of the data is dedicated to the use of AMCs in ProtoDUNE DP. It will be reworked in the case of DUNE where there will be no more an external trigger and only timestamp management will be necessary.
2. Part that receives the analog data from the 64-channel analog cards and digitizes them using 12-bit ADCs at a frequency of 2.5MHz.
3. Part that compresses the buffered digitized data while waiting for the trigger.
4. Part that stores the data in the external memories DPRAM0 and 1. These memories have dedicated spaces for each of the channels, sized to contain uncompressed data for the readout of the entire drift (4 ms) window in ProtoDUNE DP.
5. Part that controls the application globally with the help of the Nios softcore processor. In addition to supervising the application, the Nios builds the Ethernet packets destined for the MCH of the μTCA crate which will then be sent in the rest of the acquisition chain.

5.2.2 Compressor module behavior

The architecture of the compressor module is illustrated in Fig 5.3. It contains three FIFOs. The first one, Huffman FIFO, is the input FIFO of the component which buffers the values coming from the ADCs. It operates at 2.5MHz, the actual sampling speed for the detector readout. The data in this FIFO are processed by **Huffman Compression** State Machine (SM) and encoded at a faster clock than the one of sampling. They are then stored in an intermediate FIFO, Huffman description FIFO, which describes the encoding and how the data should be compressed.

The data are therefore arranged in the description FIFO (right FIFO in Fig 5.3) in 19-bit words and described by several fields:

- *Bit type - 1b* : Compression flag - `0b0` if the data to store is a value, `0b1` if it is a delta;
- *Data size - 4b* : Data size - 12 for a value, between 2 and 8 for a delta, 1 if the delta has been repeated four times or 0 if it is the end of the sequence to encode;
- *Data - 12b* : The data - ADC value or delta code value depending on whether the detected difference is greater than |3|.

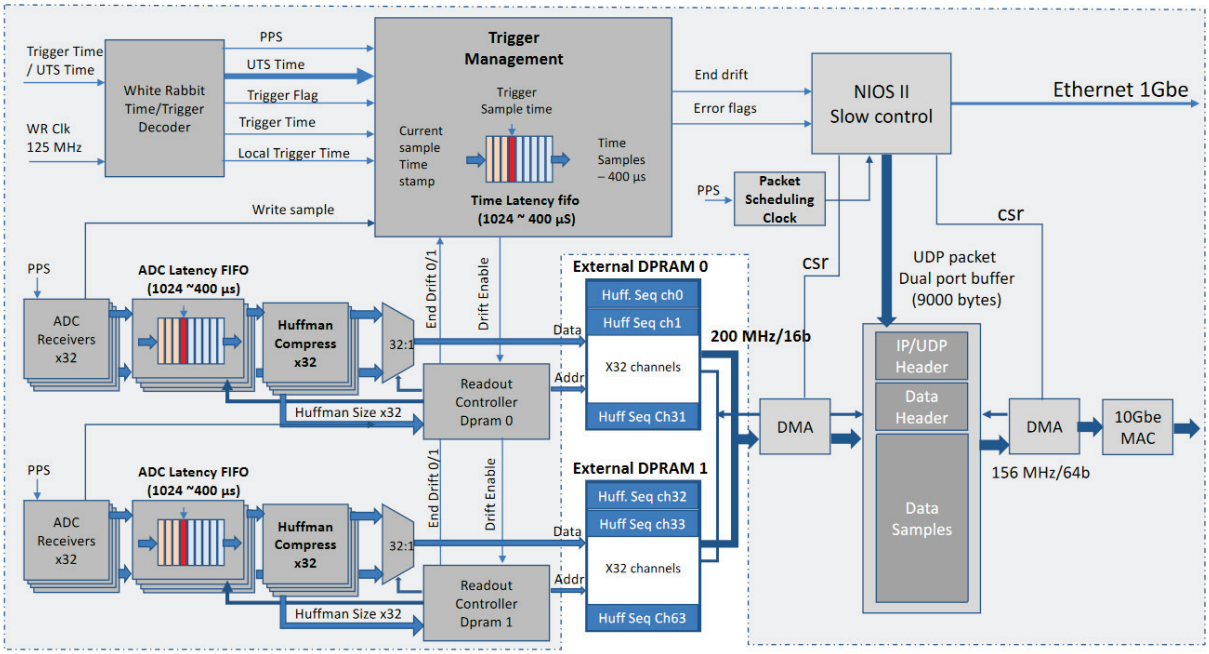


FIGURE 5.2: AMC Firmware in ProtoDUNE DP with compression.

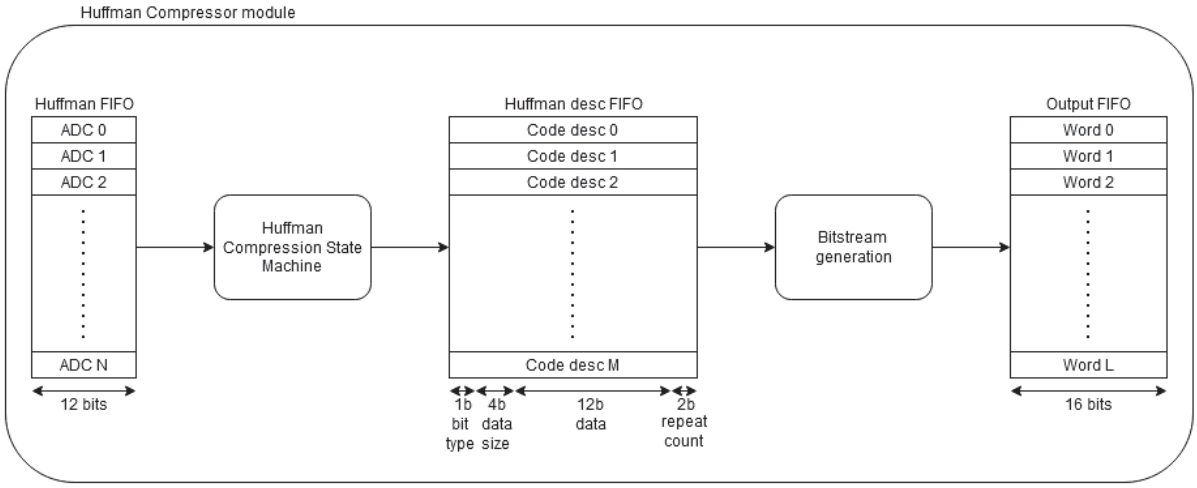


FIGURE 5.3: Compressor module architecture

- *Repeat count - 2b* : Count the repetitions (maximum 3, because at 4 repetitions there is insertion of the corresponding code).

The data from Huffman description FIFO pass through another state machine, **Bitstream generator**, that produces the final compressed data stream encoded by the Huffman algorithm explained previously. These generated data are stored in the output FIFO as 16-bit words to respect an alignment of the data sending protocols.

The synopsis of the Huffman Compression SM is shown Fig 5.4. This SM is initialized in the state **Get New ADC** which waits for the input FIFO to contain data before sending a read request. It then passes into the state **Store New ADC** where it is sure that the value that comes out of the FIFO is a raw value to be stored. A write request is thus made to the description FIFO with the information relating to the first value. This value is also stored in an intermediate signal *ADC0*. Once the value is stored, the SM goes to the **Get Next ADC** state where the same operation as in the

Get New ADC state is performed. Then, the SM reaches the state **Store Next ADC** where the value coming out of the input FIFO is saved in the *ADC0* signal while its previous value is copied to an *ADC1* signal. After this step, the SM goes into the state **Do Delta** where the difference between *ADC0* and *ADC1* is calculated and placed into the signal Δ_0, while the previous Δ_0 is moved to Δ_1 (Δ_0 is initialized to 4096, which is outside the range for any possible differences of two 12-bit values). When the SM arrives in the state **Test Max Delta** there are two possible paths to follow:

- If $|\Delta_0| \leq 3$, manage the addition of codes in the description FIFO as well as the counting of the number of successive equal differences;
- If $|\Delta_0| > 3$, check that no previous code was in the process of counting successive occurrences and to store it if necessary, as well as, store the current raw value of ADC.

In the state **Test Repeated Delta** (first case), one checks whether the current difference is the same as the previous one ($\Delta_0 \stackrel{?}{=} \Delta_1$). A counter named *repeat_count* is then incremented if the two are the same or reset to 0 otherwise. Thus this state forks into two branches:

- For $\Delta_0 = \Delta_1$, the SM enters the state **Test x4 Repeated Delta** where the counter *repeat_count* is evaluated. If the value of this counter is less than 4, the SM returns to the state **Get Next ADC** beginning the processing of next ADC value. Otherwise, it goes to the state **Insert x4 Repeat Code**. There the Huffman code (`0b1`) corresponding to 4 successive equal differences is inserted into the description FIFO and the counter *repeat_count* is reset to 0. The SM then returns to the state **Get Next ADC**;
- If $\Delta_0 \neq \Delta_1$, a new Huffman code must be stored. However, one first has to check the current number of counted code repetitions, *repeat_count*. The current value of *repeat_count* is checked in the state **Test xN Repeated Delta**. Finding *repeat_count* $\neq 0$ implies that there is a Huffman code yet to be stored that has been repeated *repeat_count* times. The SM therefore switches to the **Insert Delta xN Code** state where a write request is made to the description FIFO to insert this code the corresponding number of times. The Huffman code for the current ADC difference is stored in **Insert Delta Code** state, which follows **Test xN Repeated Delta** immediately if *repeat_count* $= 0$ or via **Insert Delta xN Code** otherwise.

In the second case, where $|\Delta_0| > 3$, from the **Test Max Delta** state, the SM goes into the **Test Pending xN Delta Code** state. The last ADC value read from the input FIFO must be stored as a raw value in the descriptor FIFO. However, first it is necessary to check whether there are any repeated codes that need to be still written. If this is not the case, the SM performs a write request to the description FIFO of the ADC value read from the input FIFO and returns to the state **Store New ADC**. Otherwise, the SM enters the state **Insert Pending xN Delta** and a write request is made to the description FIFO with the information about the code whose repetitions were being counted.

After the intermediate step of translating the ADC values into the description FIFO, a second state machine **Bitstream Generation SM**, illustrated on Fig. 5.5, is used to generate the sequence of encoded bits corresponding to the sequence of input ADC values.

The SM starts in the state **Init** which is the initialization phase where we wait for there to be words in the description FIFO, which is the input FIFO for this stage of processing. Once the FIFO is no longer empty, the SM enters the **Get Next Code** state where a read request is sent to the description FIFO. It spends one clock cycle in the **Read Next Code** state to synchronize the read and then goes into the **Load Repeat Code** state where the number of repetitions of the code/value read is

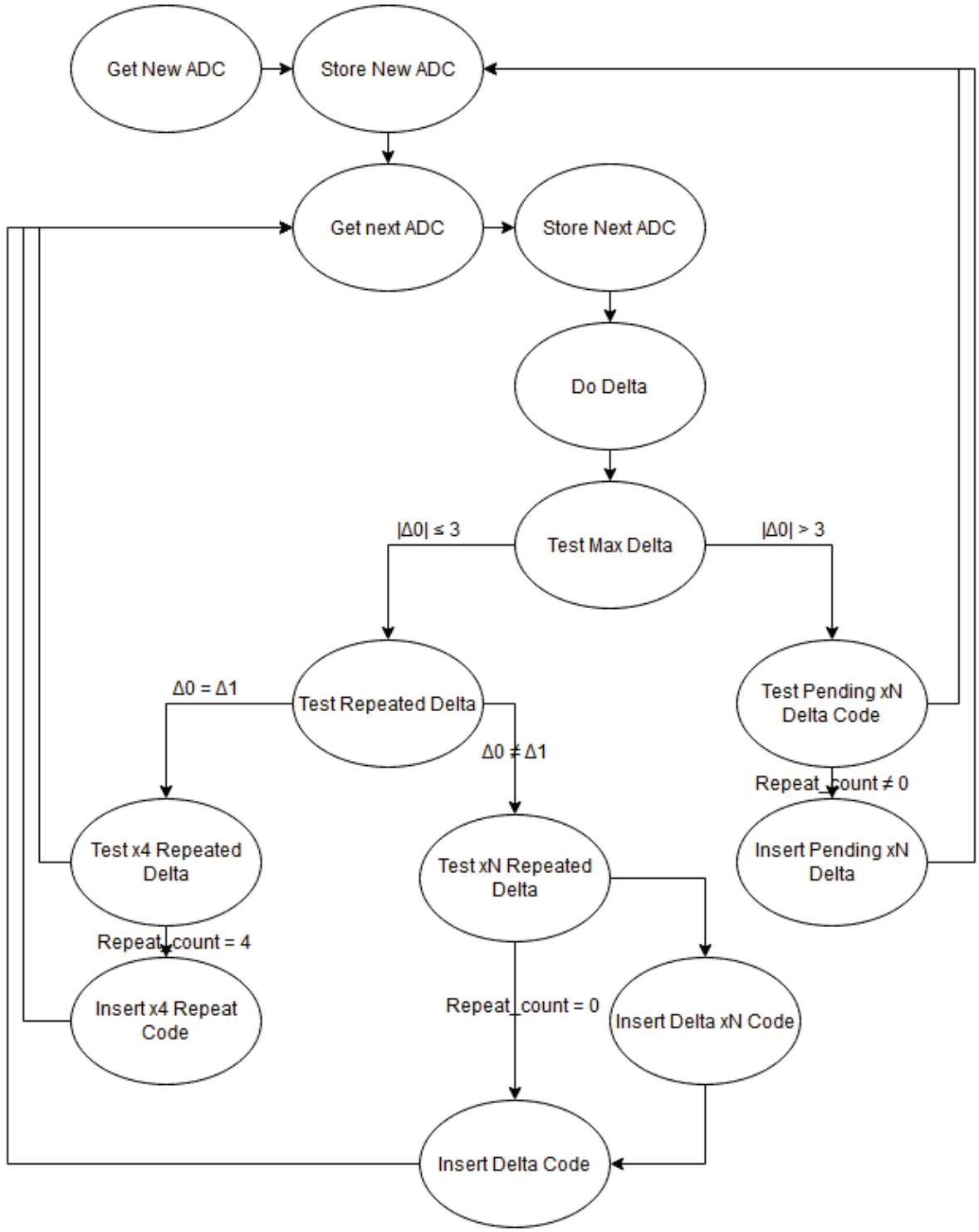


FIGURE 5.4: Compressor module Huffman compression state machine.

saved in a *shift_repeat* signal. The SM then goes into the **Load Code Reg** state where the other information contained in the word exiting the description FIFO is loaded into different signals. In addition, a *code_shift_empty* signal is passed to 0 to signify that there is data to process.

Then, the SM moves to the state **Test Padding** where it will evaluate how to continue filling the current sequence. There are several cases:

- The sequence to be encoded is finished, then the SM will go into the **Do Word Padding** state. In this step, the remaining bits in the last 16-bit word of the sequence will be filled with zeros.
- The word coming from the FIFO is a raw ADC value and the encoded sequence is not aligned on a 13-bit word (called pattern here), then the SM will pass into the **Do Pattern Padding** state. Here, and during the following stages which are **Shift Padding Request** and **Shift Padding Bit**, it will be checked that the sequence is well aligned on the 13-bit pattern to be able to insert the new raw value with its corresponding flag. Once this word has been padded, the SM returns to the **New Bit Pattern** state where the counter *bitcount*, which represents the number of bits already written in the current pattern, is reset to 0.
- There are bits to insert in the sequence and thus in the current pattern, then the SM switches to the state **Shift Code Request**.

When the SM is in the **Shift Code Request** state, what comes out of the FIFO can be either a difference code or a raw ADC value but being sure that the pattern is empty ($bitcount = 0$). Different scenarios can arise:

- If the *bitcount* is equal to 13, this means that a pattern has just been completed, then the SM goes into the **New Bit Pattern** state resetting bitcount to 0.
- If the *bitcount* is equal to 0, then a new pattern is started and a compression flag must be inserted. A bit 1 is inserted if the processing concerns a code or a bit 0 is inserted if the processing concerns an ADC raw value. The SM then goes to the state **Shift Comp Word Request**.
- If the *bitcount* is different from 0 and 13, then the SM switches to the **Shift Comp Word Request** state so that the code or the value is inserted in the sequence bit by bit.

Once all the bits have been inserted one by one after the SM has cycled around these 3 previous states, the *code_shift_empty* signal will change to 1. This will cause the SM to go into the **Dec Repeat Count** state after the last bit has been inserted and the SM has returned to the **Shift Code Request** state. In this state, the signal *repeat_count* will be decremented if it is not already at 0 and return to **Load Code Reg** to repeat the insertion of the same code. If it is at 0, then the SM will return to the state **Get Next Code** to fetch new data to insert in the sequence.

This insertion is performed bit by bit in a shift register. An external process counting the number of bits inserted in the register makes a request to the output FIFO once the word is filled.

5.3 Decompression and validation tests

5.3.1 Decompression architecture

In order to test and validate the operation of the AMC firmware with the data compression, a system that receives these data, decompresses them, and checks their integrity has to be implemented. Moreover, the implementation of the decompression in firmware must also permit studying the processing of data in real-time. As described in the previous chapter, the data acquisition for the DUNE far detectors is to operate in a streaming mode with fragments of the data written to storage only if they contain potentially interesting events. The latter is determined by an analysis to be performed in real-time on the detector data stream. Due to the lack of resource available to each AMC it is not

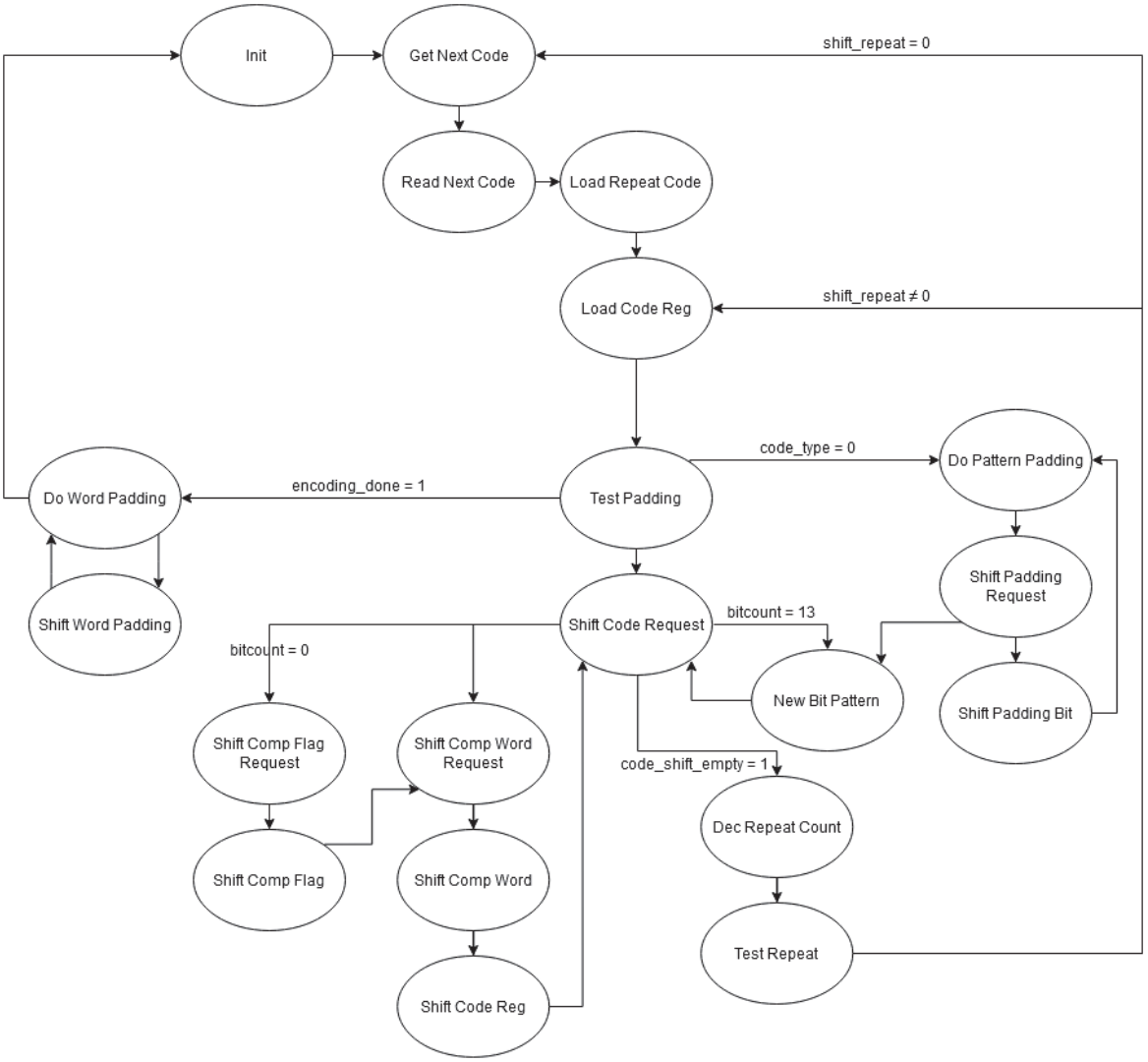


FIGURE 5.5: Compressor module state machine 2 - Bitstream Generation.

feasible to perform the required analysis processing tasks in situ on the cards prior to compressing the data. Thus this has to be done elsewhere and the decompression then becomes a mandatory first step for any further analysis processing. This section will therefore also focus on the evaluation of the feasibility of a decompression architecture for the real-time data processing.

The choice was made to work on FPGA which allows to process continuous data flows in real time and large quantities of data coming potentially from several μTCA crates. The architecture allowing this data decompression has been realized in an A10P3S board from Bittware. Fig. 5.6 illustrates the setup for the study of the decompression architecture with data from a single μTCA crates as a first step. The data of 10 AMCs are sent via the 10GbE link, are decompressed in real time on the A10P3S card and statistics from the results of this decompression are read via JTAG.

The block diagram of the firmware for the decompression implemented on the board is shown Fig 5.7. The data packets are received via a 10GbE Ethernet link. They are then analyzed by a module **RX Packet Filter** which carries out an identification of these packets, their redirection and the suppression of their header. In practice, 2 types of packets can be received by this interface:

- control packets containing AMC identification information and timestamp;
- compressed data packets.

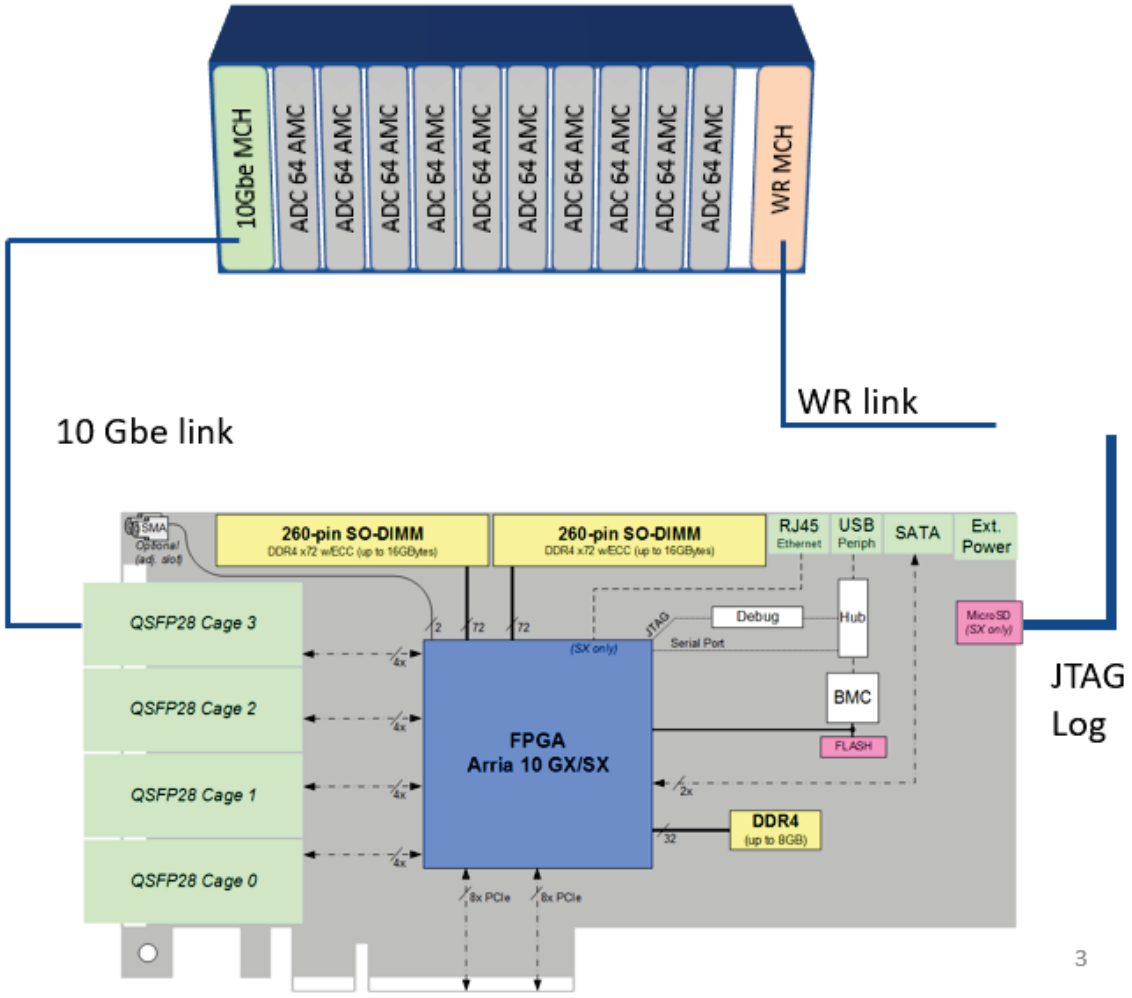


FIGURE 5.6: Decompression architecture setup

The control packets are redirected to the Nios and are processed according to their content. The data packets are sent to the following the decompression architecture. In order to decompress the data, it is necessary to work at the level of the data stream of each channel, since the Huffman compression algorithm was applied on the channel by channel basis. This is accomplished by splitting the stream into 640 distinct channels to carry out their decompression in parallel. This 1:640 demultiplexing uses 3 stages:

- a first stage with a division factor of 10, which extracts the 64 channels from each AMC card;
- a second stage with a division factor of 4, which extracts 4 groups of 16 channels: channels 0-15, 16-31, 32-47 and 48-63;
- a third stage with a division factor of 16, which uniquely identifies each channel.

Before each demultiplexer, a module **Extract Channel** reading the header of the data packets builds a signal *channel* to redirect each packet to the right place in the next demultiplexer stage that follows. Following the demultiplexors are 640 **Huffman Decompressor** modules that decompress the data for each channel. The decompressed data can then be passed to another architecture that performs necessary analyses. However, in this chapter only basic statistics (discussed in Section 5.3.3) are collected in order validate the functionality of the architecture. This information is created by the **Pedestals Counters** modules and then stored in memories accessible by JTAG.

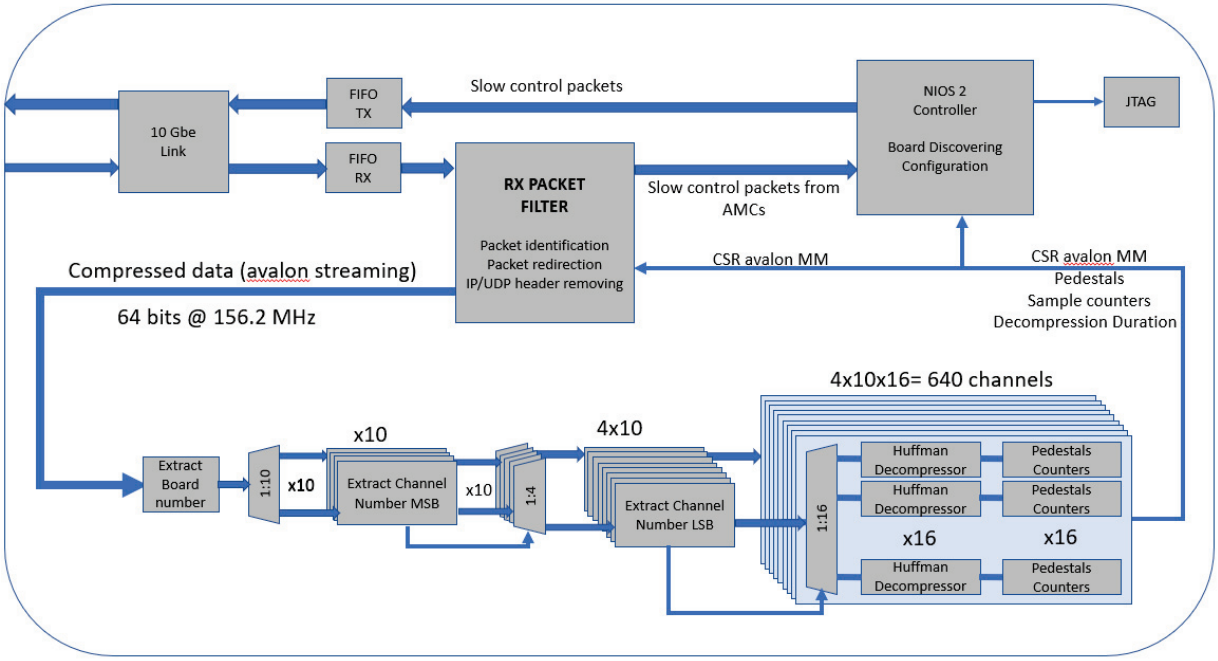


FIGURE 5.7: Decompression architecture implemented on an Arria 10 FPGA.

5.3.2 Decompressor module

Each **Huffman Decompressor** module performs decompression of the data packets coming from the same channel. Its internal architecture is shown in Fig 5.8. The module receives the data in an input FIFO **Huff Dec FIFO** byte by byte. These data are then read by a state machine interpreting the various codes encountered. Finally, once the codes have been translated, the raw values of the ADCs are recreated and stored in an output FIFO **ADC FIFO**. As previously stated, these values will only be read by the module **Pedestals Counters** to extract some basic statistics.

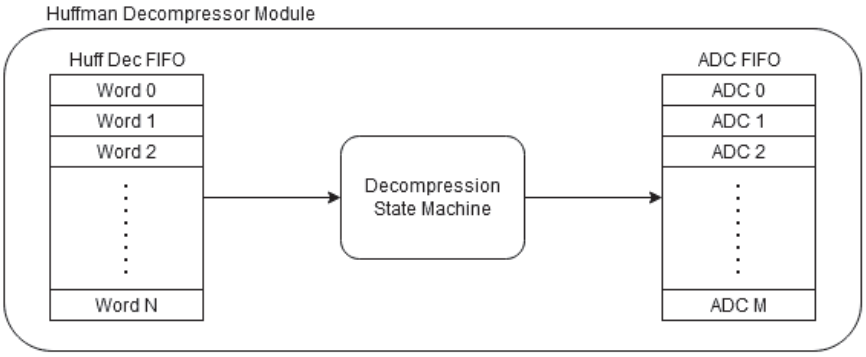


FIGURE 5.8: Huffman Decompressor Module architecture.

Words of compressed data arriving into the input FIFO of the decompressor are transferred in a shift register of 8 bits. From this register each bit is then read one by one and interpreted in order to convert the valid Huffman codes to the appropriate ADC values. The decompression module is thus described by a combination of two independent state machines. The first state machine, shown in Fig 5.9, reads words arriving in the input FIFO. The SM is initialized to the state **Read Next Byte**. In this state, it is checked that the shift register is empty and that the input FIFO contains data. When these conditions are met, the SM enters the **FIFO Read Request** state and a read request is sent to the input FIFO to prepare the next word for processing. Once the word has been retrieved,

the SM enters the **Load Shift Reg** state, which loads this word into the shift register used to analyze the codes. The SM finally passes into the state **Wait Load** for the time of a clock cycle to wait for the data to be well in the shift register then returns to the initial state **Read Next Byte**. The cycle then restarts as long as data are present in the input FIFO.

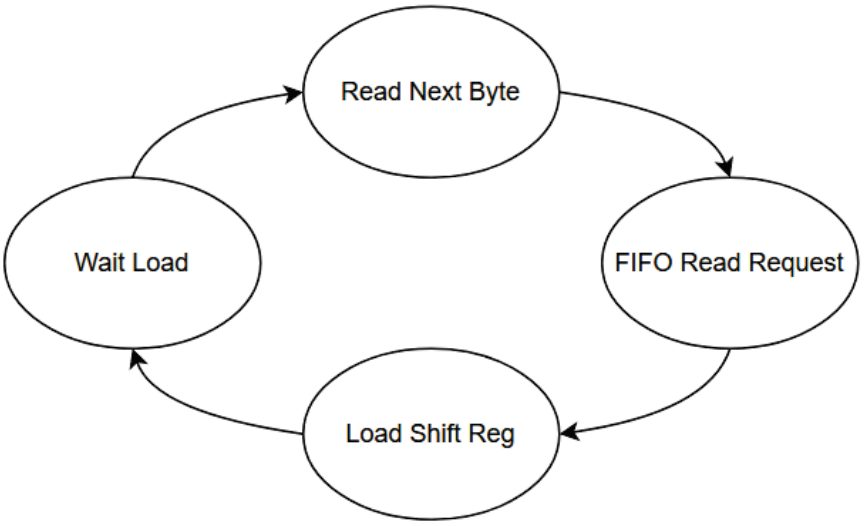


FIGURE 5.9: Entry FIFO State Machine

The second SM, shown in Fig 5.10, processes the compressed data bit by bit to interpret the codes that are loaded into the shift register. This SM is initialized to the state **Wait available bit** and awaits for notification from the first SM that the shift register has data to process. It then goes into the state **Test Shift Count** which counts the number of compressed bits processed. This counter grows to the value 13, since the Huffman compression algorithm generates 13-bit pattern words. Several cases arise during this stage. The first case (right part of the SM on Fig 5.10) is when the number of bits is a multiple of 13. In this case, the bit encountered is a compression flag and the SM enters the state **Test Comp Flag**.

- If the flag is `0b0`, then the next 12 bits will be a raw ADC value. To process them, the SM enters the state **Shift ADC Comp Flag** which shifts the flag bit inside the shift register. Then it enters the state **Shift ADC** where the 12 following bits will be shifted and kept one by one. Once the 12 bits are recovered, the SM goes into the **Store New ADC** state and a write request is made to the output FIFO with the raw value of the recovered ADC.
- If the flag is `0b1`, then the following bits will be compressed value codes.

In the other cases, the number of bits tested by **Test Shift Count** is not a multiple of 13. Moreover, the raw values of ADC being treated in the case explained previously, only codes of compressed values will be met in the continuation. Two cases are then presented, either the value of the current bit is `0b0`, or it is `0b1`.

- If the current bit is `0b0`, then the code encountered is not yet complete. In this case, the SM enters the state **Incr Index**, where the index is incremented. The SM then goes into the **Next Bit** state, which checks if there are any bits left to process in the shift register. Finally, if there are any bits left to process, it goes into the **Shift Bit** state where the shift register is shifted by one bit.

- If the current bit is `0b1`, then the encountered code is finished. The SM then enters the state **Store Index** where the index that was previously incremented is stored. Based on the previous raw ADC value and the new stored index, the new raw ADC value is calculated. Finally, the SM enters the state **Store ADC Delta** where a write request is sent to the output FIFO to store the decoded raw ADC value.

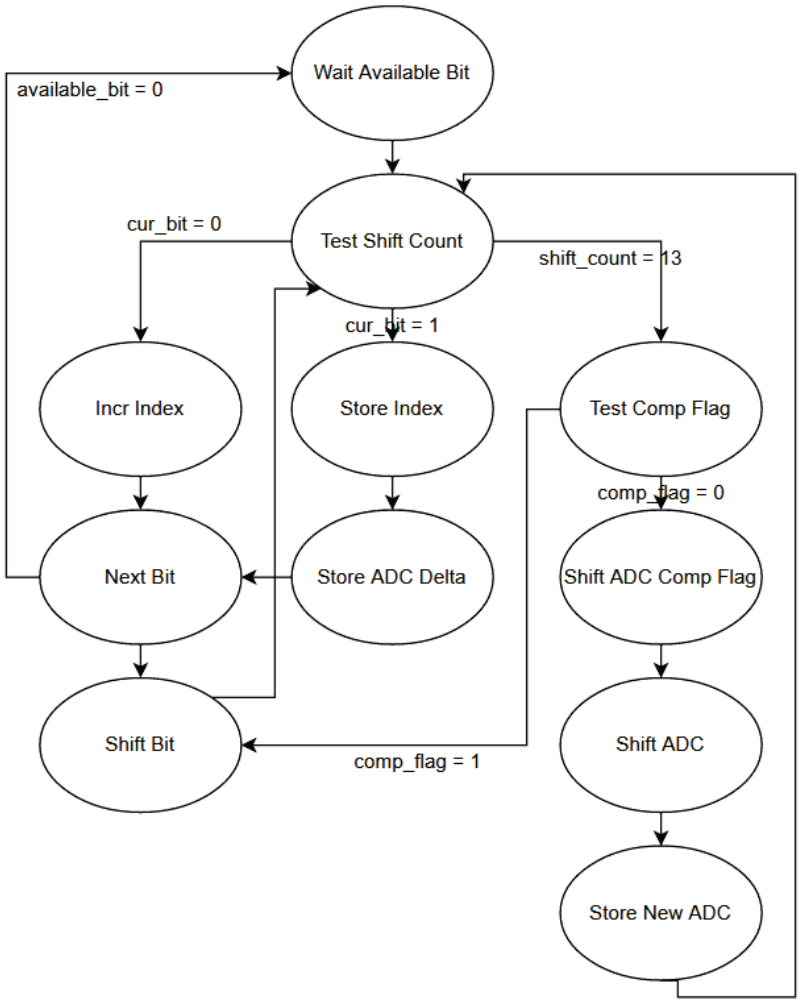


FIGURE 5.10: Decompression State Machine.

5.3.3 Decompression architecture results

The implemented decompression module architecture was tested with a 10Gb/s stream at the output of a μTCA crate. It was realized with the AMC firmware developed for ProtoDUNE DP, which in practice means that events are acquired on an external trigger with each card sending 10000 ADC samples per channel. The functional validation of the architecture could be performed using the statistics obtained for each of the decompressed channels. The statistics are presented in the form of the following example log re-transcribing a communication by JTAG after receiving the results.

```
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "A10P3S [USB-1]", device 1, instance 0
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)

Waiting for ARP: ..... OK
```

```
Waiting for board initialization: OK 9 boards replied
***** Complete Drift received => Huffman results ******
Board[0],Channel[0] => Count=10000   Flags=28   SeqSize=274   Pedestal=74   Time=187152
Board[0],Channel[1] => Count=10000   Flags=28   SeqSize=96    Pedestal=54   Time=81152
Board[0],Channel[2] => Count=10000   Flags=28   SeqSize=212   Pedestal=60   Time=150764
Board[0],Channel[3] => Count=10000   Flags=28   SeqSize=122   Pedestal=71   Time=96804
Board[0],Channel[4] => Count=10000   Flags=28   SeqSize=352   Pedestal=39   Time=234132
Board[0],Channel[5] => Count=10000   Flags=28   SeqSize=306   Pedestal=85   Time=206700
Board[0],Channel[6] => Count=10000   Flags=28   SeqSize=352   Pedestal=48   Time=234496
Board[0],Channel[7] => Count=10000   Flags=28   SeqSize=364   Pedestal=36   Time=242006
Board[0],Channel[8] => Count=10000   Flags=28   SeqSize=152   Pedestal=30   Time=113864
...
...
...
```

From the log several things should be noted:

- The test architecture starts by discovering the μTCA crate and the AMC cards present inside. Here, nine AMC are present in the crate.
- Each channel decompressor returns independent statistics on the data it has decompressed.
- These statistics include:
  - A flag field used for debugging, here `28` (`0b101000`) means that the sequence has been fully decompressed with no errors detected.
  - A SeqSize field containing the size of the compressed data sequence in 64-bit words.
  - A Pedestal field, which returns the average of all decompressed values.
  - A Time field, which measures the decompression time of each sequence, here in number of clock cycles at 156.25MHz.

With the model of the decompressor having already been validated in a simulation, the verification of the calculation of the pedestals with respect to their measured values allows a functional validation of the architecture. In order for the architecture to run in real time, the time for decompression must also be less than the acquisition time for a complete drift of 10000 samples. With the sampling rate of 2.5 MHz, this corresponds to a time window of 10000×400 ns $= 4$ ms. Fig 5.11 illustrates the measured decompression times as well as the compression factors or ratios per channels from a μTCA crate equipped with nine AMCs boards.

As can be seen in the figure the decompression time (light blue curve) for a channel is correlated with the compression ratio (dark blue curve). The higher the compression ratio, the faster is the decompression. This is because there are fewer bit sequences for the decompressor to read and therefore less processing to do. The decompression times are between 0.4 and 1.8 ms, below the 4 ms limit.

These data were acquired from a test bench system with only AMCs (i.e., not connected to the front-end analog electronics). As this corresponds to the lowest possible noise conditions, the compression ratios can be higher than what can be achieved once the digital stage is connected to the analog. As described later on in this chapter, the compression ratios measured in ProtoDUNE DP were on average around 5. However, even for such compression ratios, as Fig 5.11 demonstrates, the required decompression time is around 1.8 ms, much below the 4 ms limit. The implemented solution thus fulfills the conditions imposed by the acquisition of the detector.

The last evaluation to be carried out is the study of the resources taken by the architecture in the FPGA in order to assess how many μTCA crates can be processed by a single card . The amount of FPGA resources taken by the architecture to treat a single crate is presented in the Table 5.2. As can be seen, about 60 % of the logic and 77 % of the internal memory are used.

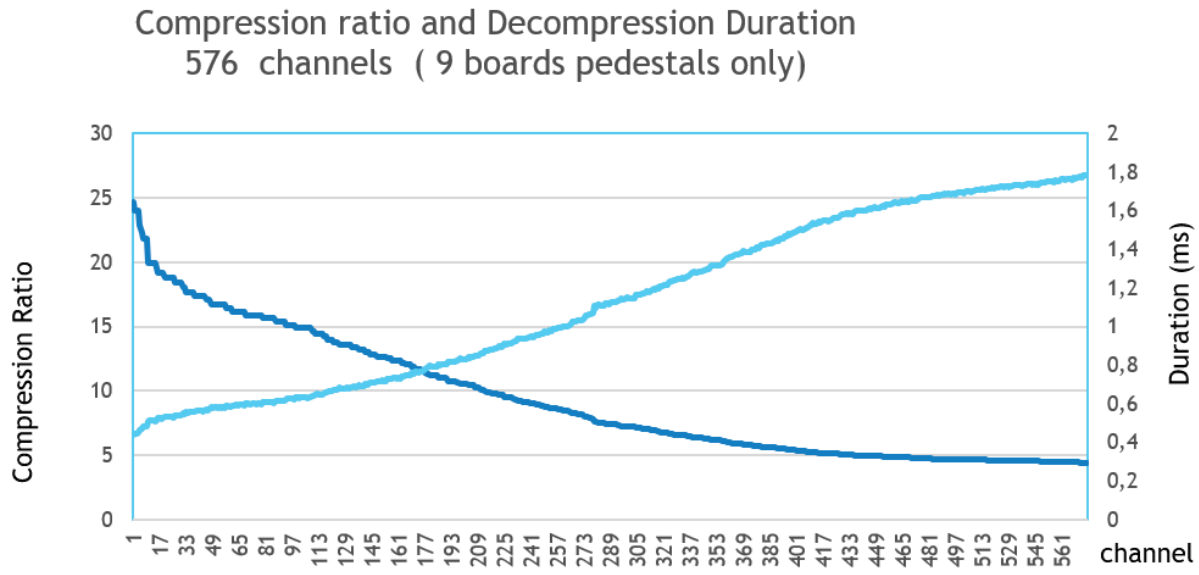


FIGURE 5.11: Compression ratio (compressed data size divided by the raw data size) in dark blue and associated decompression time in light blue measured for each channel.

Device family	Arria 10
Logic Utilization (in ALMs)	254,931 / 427,200 (60 %)
Total block memory bits	42,948,388 / 55,562,240 (77 %)

TABLE 5.2: FPGA resources used by the decompression architecture to treat one crate.

The amount of resources consumed by the architecture allows to decompress only data from a single crate. In other words, it is not possible to simply duplicate this architecture several times on the board in order to process multiple crates. The implementation of 640 independent decompressor modules takes up a lot of logical resources. In addition, in order to perform the decompression in real time, each packet must be buffered just before being decompressed requiring high usage of the internal memory.

5.4 Pedestal optimization procedure

The stability of the pedestal is affected by the position of the pedestal analog level within the ADC digitization step. If the pedestal is close to being aligned with the quantization level of a given ADC code N, the digitized baseline would mostly contain N with occasional fluctuations to $N \pm 1$. As the pedestal levels get closer to the middle between two codes, the resultant digital signal would contain more ± 1 code transition. In terms of the compression factors, this would therefore lead to worsening of the data compression. This effect is illustrated in Fig. 5.12, which shows the measured dependence of the compression factors on RMS of digital signals. The channels with pedestal RMS close to 0.5 ADC correspond to the cases where the analog pedestal level falls in the middle of the two ADC codes and have the lowest compression factors of around 5. As the RMS decreases—pedestal analog level moves closer to an ADC code boundary—the compression factors increase reaching the maximum of around 30.

The AD9257 ADC provides functionality to adjust offsets for each channel independently in a range from -128 to $+127$ counts (in 14 bits resolution). Since only 12 bit are acquired, one can exploit this feature in order to tune pedestals for each channel closer to 12 bit ADC code boundary thus improving the compression. Fig. 5.13 shows the block diagram for performing pedestal adjustment over the network. The slow control software sends a single Ethernet control frame containing 64 offset

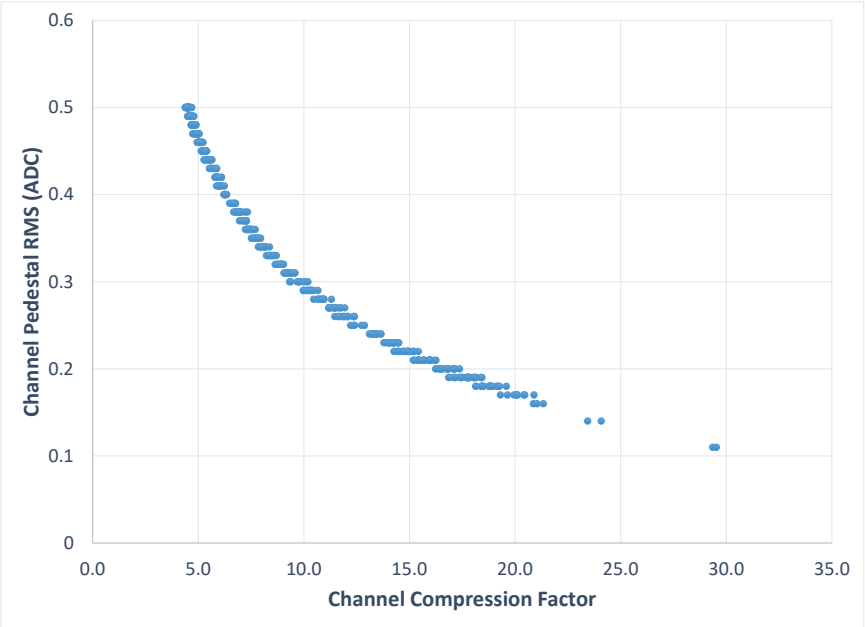


FIGURE 5.12: Measured dependence between pedestal RMS and compression factor.

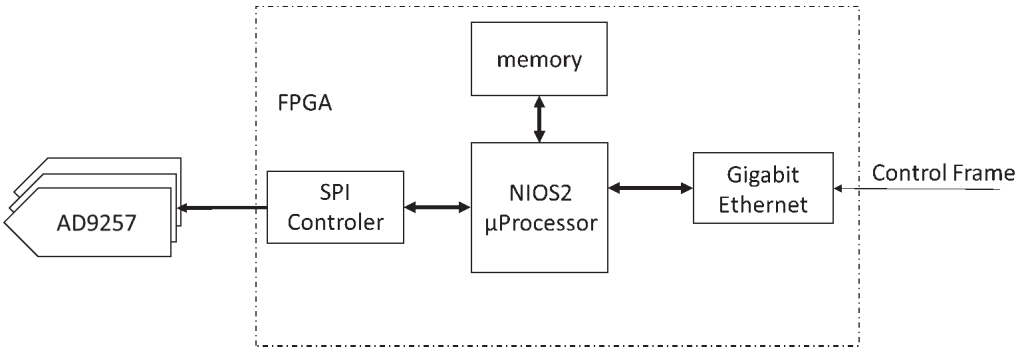


FIGURE 5.13: Front-end ADC pedestal adjustment mechanism.

values for of the AMC channels. These are of *char* type and thus are ranging from −128 to +127 counts matching the offset adjustment limits.

Apart from tuning the pedestal fluctuations to improve compression, the ADC offset functionality at the same time also allows to align the mean values of the pedestals for each channel to a common target. Typically the AMC pedestals show a channel-to-channel dispersion with a RMS of about 15 ADC counts (in 12 bit resolution) illustrated by the plot on the left in Fig. 5.14 which provides an example of the pedestals measured for 10 AMC cards (640 channels) in one of the ProtoDUNE DP μTCA crates. The right plot of the same figure shows the measured mean pedestal values for the same set of channels after performing the pedestal alignment by adjusting the offsets to bring all pedestals to a common mean value of 80 ADC (in 12 bit resolution). As can also be seen from this plot, due to a limited range of the offset parameter, which in 12 bit resolution is −32 to +31 counts, some of the channels that lie too far from the target value cannot be fully aligned.

For the pedestal RMS minimization one has to find an optimal offset to be applied to each ADC channel. The key idea is to move offsets by ±1 count in 12 bit resolution using the fact that the

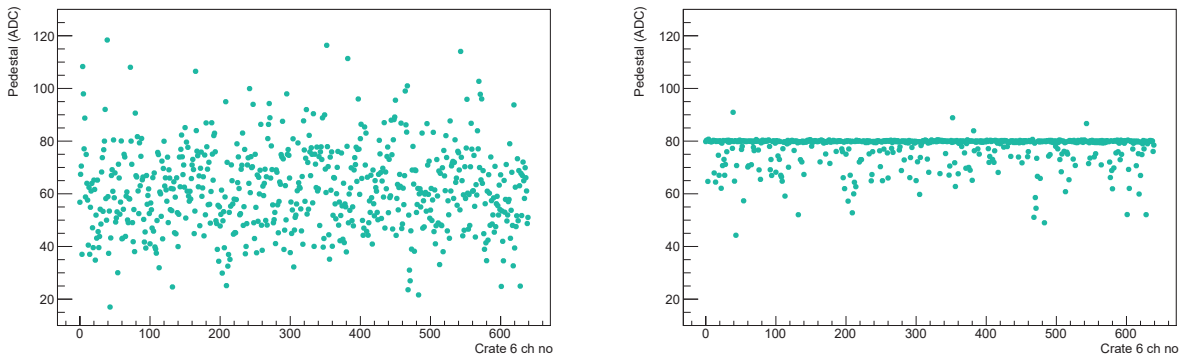


FIGURE 5.14: Illustration of the ADC pedestal adjustment: image on left (right) shows the example of pedestal values for channels in a ProtoDUNE DP μTCA crate before (after) pedestal adjustment. All the values are reported in 12 bit resolution.

intrinsic ADC resolution is 14 bits and consequently one code step in 14 bits is 1/4 in 12 bits. If one defines:

- P_i initial mean pedestal value prior to any optimization procedure for each channel,
- T target value for pedestal alignment, which in practice is computed as average over all P_i before the start of the optimization procedure,
- $\sigma_i^{(J)}$ pedestal RMS value for a given channel i after an optimization step J,
- $O_i^{(J)}$ an offset to be applied at step J;

this optimization procedure is then the following:

1. With all offsets set to 0 measure P_i and compute T,
2. In range $J = -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$ program offsets $O_i^{(J)} = T - P_i - J$ and measure resultant $\sigma_i^{(J)}$,
3. Calculate a final set of optimal offsets O_i^f by picking for each channel an offset $O_i^{(J)}$ out of the 11 values that gives the lowest $\sigma_i^{(J)}$,
4. Load final O_i^f into ADC.

Fig. 5.15 shows the pedestal RMS values on the left (right) measured in one of the ProtoDUNE DP μTCA crates before (after) offset optimization was performed. As can be clearly seen the pedestal RMS values move to lower values after the optimization.

5.5 Results from ProtoDUNE DP operation

The performance of the compression was studied with the electronics system installed in ProtoDUNE DP. The compression firmware was loaded on AMCs at the end of the prototype running period to evaluate the compression factors in realistic conditions where the AMCs were connected and reading the cryogenic analog front-end cards. Fig.5.16 summarizes the key results obtained in that period. In order to evaluate the impact of the pedestal optimization procedure described previously measurements

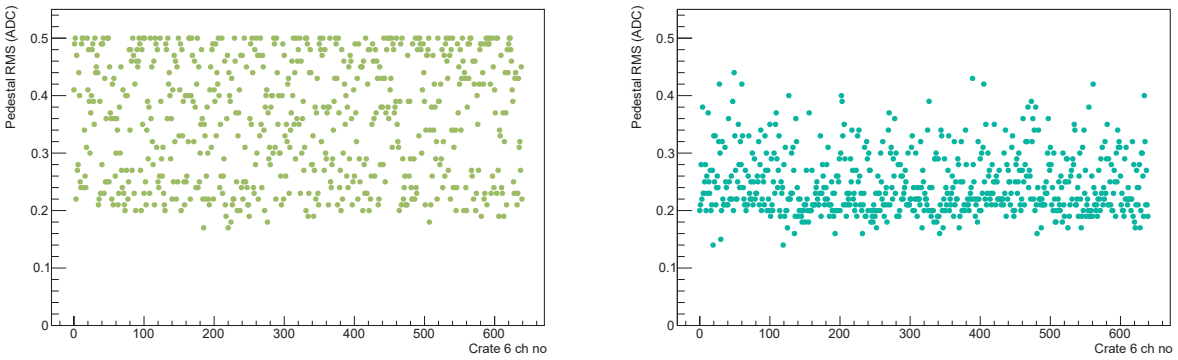


FIGURE 5.15: Illustration of the ADC pedestal RMS optimization: image on left (right) shows the example of pedestal RMS values for channels in a ProtoDUNE DP μTCA crate before (after) pedestal adjustment. All the values are reported in 12 bit resolution.

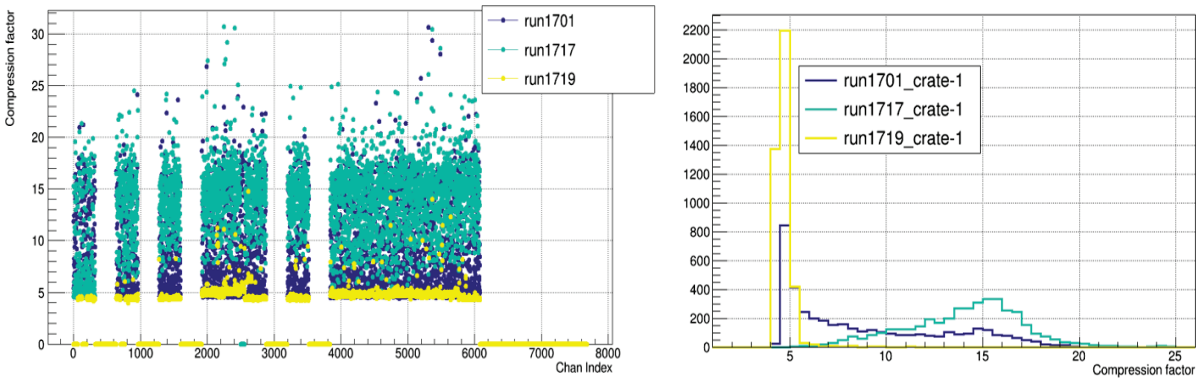


FIGURE 5.16: Data compression factors measured in ProtoDUNE DP. The left image shows the compression factors for all instrumented ProtoDUNE DP CRP channels (gaps correspond the channels for the non-instrumented CRPs). The points in blue (green), or runs 1701 and 1717 respectively, correspond to measured compression factors before (after) pedestal optimization with analog front-end electronics switched off, while those in yellow, run 1719, are the results obtained after the pedestal optimization when the front-end electronics was turned on. The image of the right shows the projection of these data for 1st μTCA crate with the color scheme as described previously.

were performed with front-end electronics disconnected at first to avoid the noise contributions coming from the inside of the cryostat. The measured compression factors obtained before and after pedestal optimization are shown in dark blue (*run 1701*) and green (*run 1717*), respectively, in the figure. As can be seen the pedestal optimization led to a significant improvement in the compression with average compression factor between 10 and 15. After the optimization was performed, the front-end analog front-end cards were powered and the compression factors for each channel were remeasured. The results are superimposed in yellow (*run 1719*) in Fig.5.16. Despite the gains in the compression due to the pedestal optimization procedure, the noise coming after connecting front-ends leads to the overall degradation in compression with the mean value of around 5. Nevertheless, as discussed in the beginning of this chapter this is still sufficient to support the continuous streaming with 10 Gb/s MCH switch.

Chapter 6

Search for trigger primitives

6.1 Introduction

6.1.1 Trigger primitives definition

The Trigger Primitives (TP) constitute the primary information used by the DUNE DAQ system in order to decide that some interesting activity was observed in the detector and that it is worth to record the corresponding data permanently on disk. DUNE deals with different data streams related to the various sources of interactions in the detector:

- Beam events: interactions in the Far Detector modules of neutrinos from the beam produced at Fermilab. The proton beam is delivered in a narrow 10 μs time window (beam spill) with a period of a about a second. This defines precisely the timing for the beam-related events.
- Cosmics: this data stream includes all interactions related to cosmic ray activity (muons or atmospheric neutrinos) and potential proton decay candidates. These events are out of the beam time window and typically deposit an energy greater than 100 MeV in the detector.
- Supernova neutrinos: events from interactions of neutrinos from a Supernova burst. They have energies of the order of a few tens MeV and are distributed over a long time window of a few tens seconds.
- Radioactivity and solar neutrinos: low energy events (<10 MeV) related to radioactivity contaminants in LAr (like the ^{39}Ar isotope which corresponds an activity of the order of 1 Bq/kg) or other radioactivity present in the detector materials as well as events from low-energy solar neutrino interactions.
- Calibrations: these events are related to externally introduced calibration signals (FE electronics pulsing, laser signals in LAr, neutron guns, etc ... and they have a corresponding external trigger produced at the time of the generation of the physical calibration signal

Whatever its physical source may be, an observable event will always imply the presence of some basic energy depositions (hits), visible in a set of channels, related to the ionization produced by the particles detected by the liquid argon time projection chamber. The trigger primitives correspond to these hits which represent, for a given channel, the clouds of electrons produced by the primary ionizing particle. These clouds drift at constant speed inside the TPC until when they are collected by the anode planes and amplified by the cryogenic front-end electronics and read out by the digitization system, which produces waveforms representing what each channel has seen as a function of time. This

stage of the readout chain is continuously sampling with ADCs the electrical signals of each channel at 2.5MHz on 12bits, producing a sample every 400 ns. In absence of signals the recorded waveforms have a pedestal baseline, affected by small variations due the noise of the front-end electronics. Given the fact that the DUNE far detectors operate underground, they will be idle for most of the time and the majority of data will just correspond to the electronics noise affecting the pedestal baseline. In practice, the TPs will correspond to signal peaks detectable above this background noise. Each TP is a data record containing the basic information of the detected hit, such as its energy, starting time, time duration, and channel number.

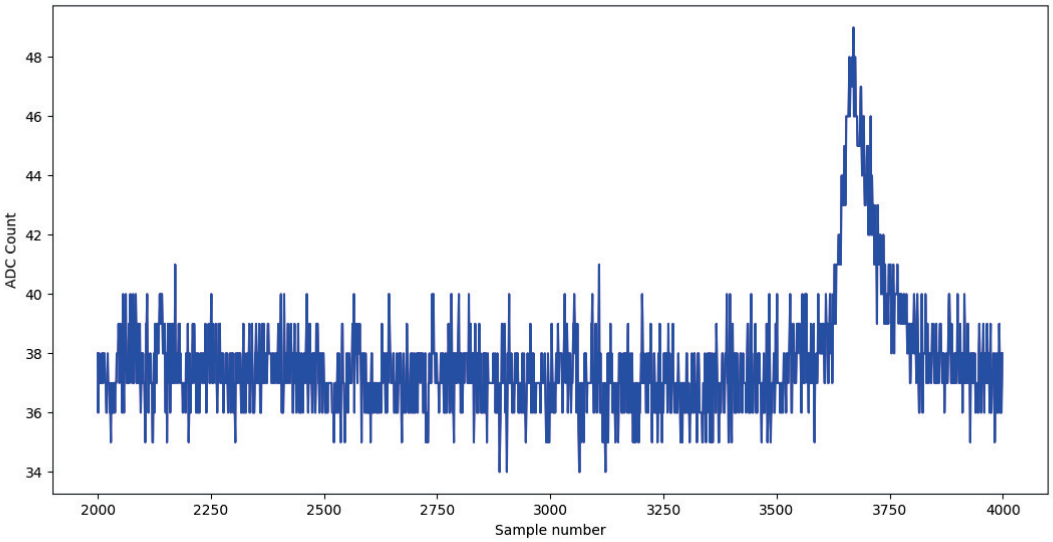


FIGURE 6.1: En example of a raw data waveform from one of the ProtoDUNE dual-phase channels.

Fig. 6.1 shows a part of the waveform from one of the ProtoDUNE dual-phase channels collected during one of the cosmic ray runs. A hit and its corresponding TP could be defined in between sample number 3650 and sample number 3750. Moreover, Fig. 6.1 also illustrates an important characteristic of the detector which is that most of the data from the different channels are pedestal data (background noise). In this particular case, out of 2000 samples shown, there are only about 100 indicating the possible presence of an interesting energy deposition.

The samples selected in the plot are biased in order to illustrate the TP. When considering the full record contain 10000 samples per channel and the fact that most of channels in the detector will not contain any hits, the overall data occupancy is much lower in the whole detector. Nevertheless, once the condition to record an interesting event is met, given the rarity of the DUNE events and the sophistication of the associated analysis, it is very important to store on disk all possible information of the event without suppressing any sample, including the overwhelming majority of noise samples. The trigger primitives constitute then a reduced online data set which allow a fast processing by the trigger system in order to make a decision and then deal with the storage of a much larger amount of information including all the noise in between hits in the complete drift window associated to the event.

The Trigger Primitives are the basic tool to assess the activity present in the detector during the fast online processing performed by the DAQ trigger system and that is why it is important to study them and develop their online definition system.

As described in Chapter 4, the detector generates enormous volumes of data. Enabling the observation of extremely rare supernova events requires continuous scanning and buffering of the data stream for

the presence of the potential supernova neutrino interactions within a large time window spanning tens of seconds. It is not feasible to simply store the data continuously and to analyze them with classical resources (processor farm, GPU in offline analysis). Thus it is necessary to reduce the data rate to be able to record only potentially interesting events. For that, the study of a part of the data really representative of the activity of the detector in real time is needed. This section will therefore try to explain how and with what means, the study of TP seeks for a solution to the reduction of the data flow without losing any data.

6.1.2 Detection of Trigger Primitives

In order to reduce the data flow with the help of TPs, it is necessary to define them in real time in the huge data volume produced by the detector. Several constraints must be considered in order to choose the right hardware for the detection of the TPs:

- In order that a TP corresponding to an interesting energy deposition is defined, several samples must have a value significantly higher values than the pedestal.
- It is therefore necessary to work on a sequence of digitized data, so that a digital architecture can work on it.

These constraints limit the places where the detection of primitives could be done in the acquisition chain. Moreover, the TP detection must remain in the online detection chain, i.e. before the global data storage.

The most interesting place meeting all these criteria is therefore located immediately after the μTCA crate hosting the AMC cards used to digitize the signals and performing a first concentration step (640 channels per crate).

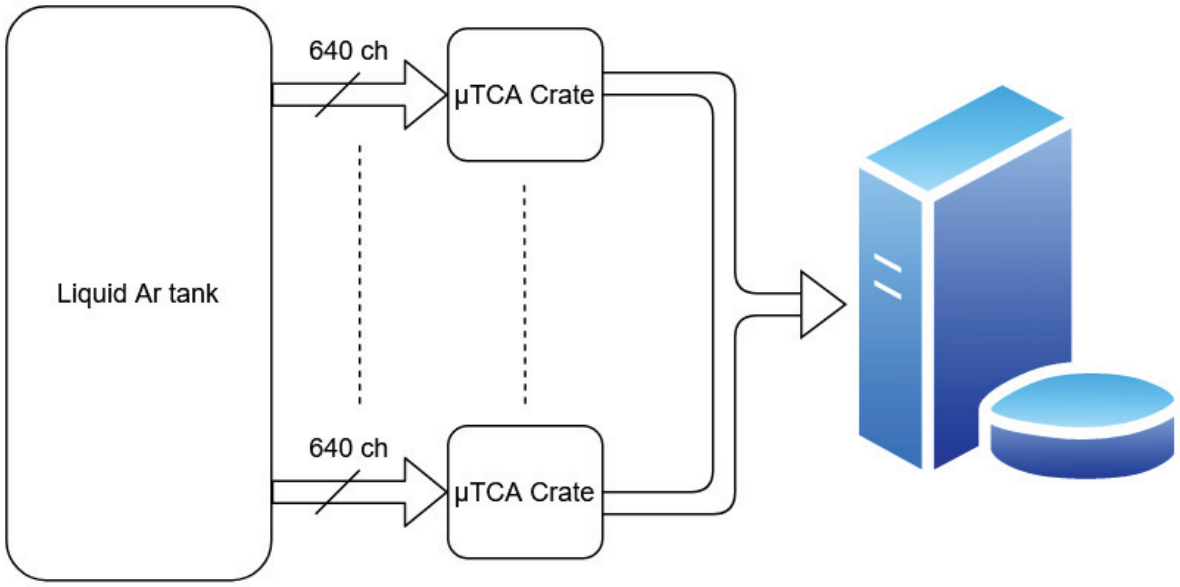


FIGURE 6.2: Simplified acquisition chain.

Fig 6.2 presents an oversimplified view of the acquisition system where the data from the μTCA are meant to be stored after being acquired.

Three types of classical TP processing architectures could be considered here:

- Processors
- GPUs
- FPGAs

Each one of this solutions has advantages and disadvantages:

- Processors are more adapted to general sequential applications and allow a great flexibility in the treatments but they are relatively slow on very specific and repetitive treatments because they require many transfers to and from the memory.
- GPUs are well adapted to more specific and parallel processing because of their large number of computing units and memory. However, they require a large amount of data to gain in computation time, which means that they have to wait for a certain amount of data before performing calculations, which makes them less suitable for real-time applications.
- FPGAs are perfectly adapted to specific and parallel processing. They are fast and flexible but have, in general, little memory. However, they have a large available bandwidth due to the numerous high-speed links. They therefore meet the conditions for real-time processing.

FPGAs were thus chosen to carry out the study on the TPs presented in this work. Their high bandwidth and their flexibility matches well the needs of the acquisition system. However, it could have been interesting to evaluate as well the performance of GPUs for this task. By working to optimize an architecture and to control a certain latency on this platform, it is not impossible that a solution based on GPUs could be developed as well.

In order to be as close as possible to the data to perform the processing, the FPGAs chosen for the processing will be placed after μTCA crate. These crates normally send data to the storage architecture using 10GbE links from their respective MCHs.

The FPGA boards used for the real-time TP study are the S5PHQ and A10P3S boards [45] from vendor BittWare. They are respectively based on a Stratix V FPGA and an Arria 10 FPGA from Intel. These two FPGAs are among the latest and most powerful on the market. They have relatively equivalent resources in terms of on-chip memory (50Mb). However, Arria 10 has twice as large the number of logic elements (500k vs 250k). The cards are also equipped with 40G QSFP+ links (2 links for the S5PHQ and 4 links for the A10P3S) that have the ability to split into 10G links each. This means that up to 16 crates can be concentrated for processing on the larger of the two cards used. The treatment is to be implemented on only one FPGA at a time, they were just both evaluated separately for the same treatment.

Fig 6.3 shows how the acquisition could look like, including the evaluation on the TPs. The data flow would be duplicated right after the crates and while the data would be temporarily stored, the TPs would be computed and evaluated to take a decision on the final storage of these data.

It should be noted that the 10 AMC cards present in the μTCA crates, which digitize and concentrate the signals of 64 channels per card, carry out these operations thanks to a Cyclone V FPGA. A solution could be imagined as well with a calculation of the TPs on these FPGAs. However these are smaller and less expensive FPGAs that do not allow such processing in addition to the operations already implemented, in case they have to perform also data compression.

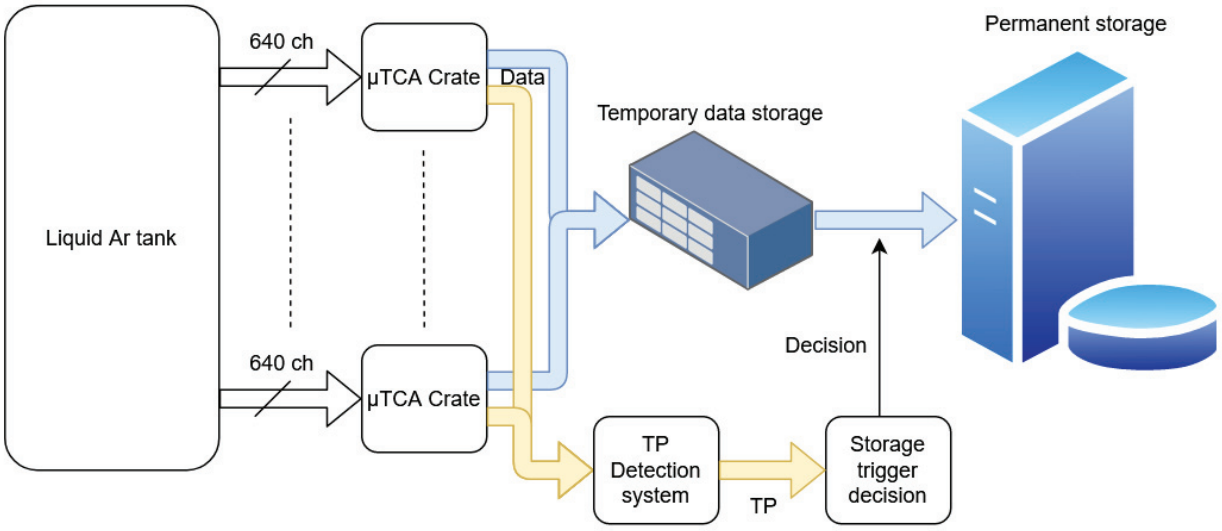


FIGURE 6.3: Simplified acquisition chain with TP treatment.

6.1.3 Trigger Primitives data format

In order to efficiently reduce the amount of data to be analyzed, each TP should be uniquely recognized while keeping only the most useful information. To do this, the TPs must send at least the following information:

- Crate number (0-239);
- AMC card number inside the crate (0-9);
- Absolute time from the beginning of the data packet;
- Number of the sample in the data packet where the TP starts;
- Total number of samples in the TP;
- Sum of the charges of the samples in the TP.

These TPs will then be processed in order of arrival and sorted by their crate, AMC card and channel numbers by the architecture making the decision to retain the data or not.

6.1.4 TP algorithms and processing

The algorithms used here have been evaluated with real data from ProtoDUNE runs. They work as close as possible to the real conditions of acquisition on raw data of ADC on 12 bits (values between 0 and 4096).

These TP detection algorithms have several requirements due to the conditions in which they will be applied. They must be able to :

- Process the samples in real time;
- Keep pace with the rate of acquisition.

The main objective of these algorithms will be to try to evaluate the minimum amount of resources needed to perform the TP search. To do this, the chosen algorithms will focus on:

- Find the pedestal baseline of the raw data signal (which is non-zero);
- Apply a threshold with respect to this baseline to detect the TPs.

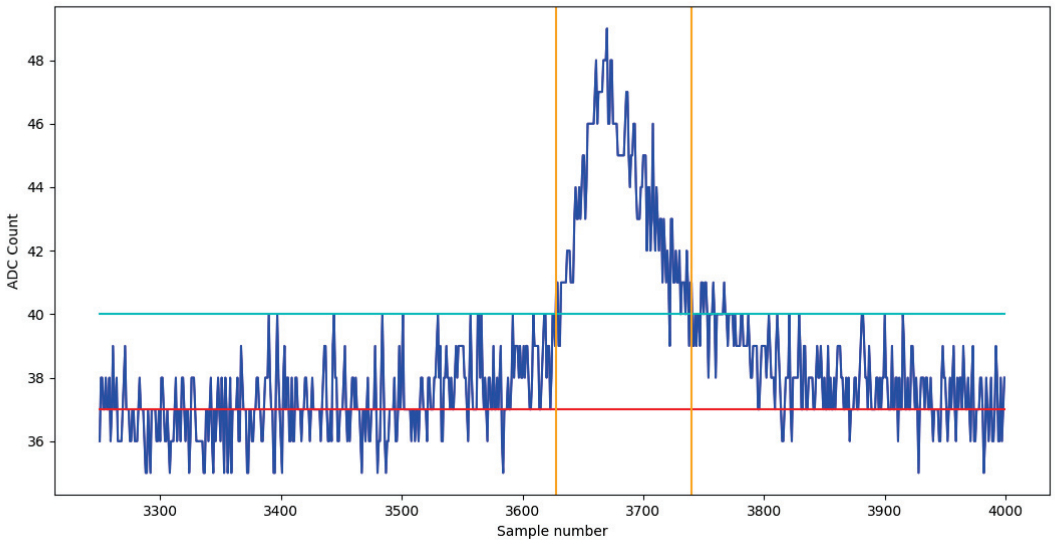


FIGURE 6.4: Raw data representation of one channel example with TP.

Fig 6.4 illustrates graphically the different elements that will be important for TP search. First, the algorithm needs to find the baseline pedestal of the signal (red line). Then, a threshold relative to this baseline will be applied (blue line). Finally, samples above the threshold will be classified as part of a TP. In Fig 6.4 the start and end sample of the TP (around 3625 and 3725) are indicated with the yellow vertical lines.

Several options are available to perform the two steps previously described. First, to find the baseline of the signal, the calculation consists in approximating the median of the signal because it is a robust value against large variations like TPs. This step is necessary in case the pedestal of the signal would drift with time. The most accurate way to approach this median would be to sum a large number of samples and divide them by this number, on a sliding window for example.

However, this method is too costly in terms of resources because it would require a buffer memory per channel to perform this calculation. With the number of channels being very large, the loss of resources would be too important compared to the resources available on FPGA. It was therefore chosen to work in a streaming mode using "frugal streaming" algorithm.

This algorithm works as follows, the median noted m and the raw data packet value i (ADC Count) noted v_i:

1. Initialize m, and i = 0
2. v = v_i
3. If v > m, m += 1 or if v < m, v -= 1
4. $i+ = 1$, go back to 2.

The algorithm has some problems to be operational.

First, if we initialize m to 0, a calibration phase would be necessary before we can effectively detect the TPs. Such a phase at the beginning of each acquisition launch could be envisaged because the acquisition is supposed to work without stopping. However, it is possible to avoid this step because runs dedicated to calibration are already planned where a calculation of the pedestals is performed. The initialization of m directly to the value of the pedestal is therefore possible by storing these values in the FPGAs' memories after the calibration.

Secondly, when the algorithm will encounter a TP, the median will follow the signal. This behavior is all the more problematic on small TPs where the peak of the signal would have a weak slope of rise. To solve this, it is interesting to add a variable that "protects" the value of the median when a TP is encountered. Indeed, by looking if the next sample (or another sample a little further) is above the threshold, it is possible to stop the update of the median in advance.

The algorithm can now be presented in the following form:

1. Initialize m = pedestal, i = 0, $1 < k < 10$, *update_med* = True and t = threshold
2. v = v_i, s = v_{i+k}
3. If *update_med* = True and s > t, set *update_med* = False
4. If *update_med* = False and v < t, set *update_med* = True
5. $i+ = 1$, go back to 2.

This algorithm thus allows the updating of the median to be stopped slightly in advance of the time when a TP is encountered and then resumed from the time when the current sample is below the threshold. Improvements to this algorithm can be imagined, especially in the case (the most encountered) where the pedestal value is very stable over time. An accumulator can receive the variations of m and update m only when a certain value of this cumulative variation reaches a predefined number (5 or 10 for example). This results in more stability on the value of the median.

Several methods exist to select the threshold, it can be chosen according to the noise of the channels or arbitrarily if a fixed threshold in energy is desired. In all cases, the threshold will be specific to each channel.

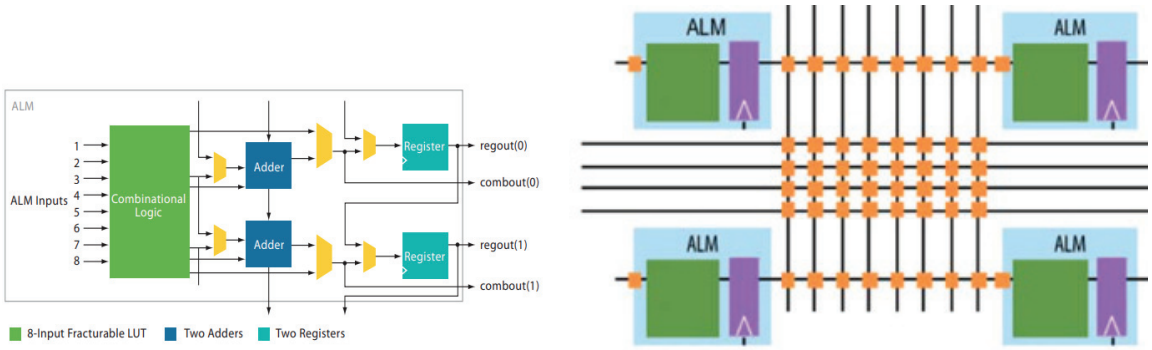
In the following, different versions of these algorithms will be implemented as a block receiving data in real time on FPGA. A first section will describe important notions about the FPGA as well as the elements necessary for the design of the architectures. Then a second and third sections will describe implemented architectures.

6.2 Introductory materials on FPGA and its development process

6.2.1 General description of FPGA

An FPGA (Field Programmable Gate Array) is an integrated circuit containing a matrix of interconnected elementary logic blocks whose particularity is to be re-programmable. These blocks implement a Look-Up Table (LUT), which is a programmable memory table with basic logical functions (adder, mux, register). Most of the FPGAs on the market, and in particular those which will be used for the processing explained below, are based on memory cells of the SRAM (Static Random Area Memory) type. These elementary blocks make it possible to carry out complex logical functions. At Xilinx, these

blocks are called CLB (Configurable Logic Block) and at Intel, ALM (Adaptive Logic Module) [46]. Fig. 6.5a shows the schematic of an ALM, which contains an 8-input LUT capable of implementing, for example, 2 independent 4-input functions. These ALMs are then linked together by programmable interconnects as shown in Fig. 6.5b.



(A) Adaptive Logic Module (ALM) schematic, source: Intel FPGA

(B) ALMs Interconnections schematic, source: Intel FPGA

FIGURE 6.5: Internal FPGA ressources.

The Hardware Design Language (HDL) is used to describe these complex logic functions and define their inputs / outputs, similarly to what is used for the design of ASICs (Application Specific Integrated Circuit) which, unlike FPGAs, cannot be reprogrammed. Using a design tool, this HDL description is converted into an initialization file of the elementary blocks of the FPGA as well as the configuration of their interconnections, also called firmware.

The FPGA design flow also requires the use of simulation tools to activate the logic functions in HDL as well as their timing. Indeed, these logic functions have a sequential operation punctuated by a clock. Simulations also make it possible to verify the interaction in an architecture between several logical functions, which may be different.

The constraints of an HDL architecture are characterized by the necessary resources (elementary building blocks), and its maximum operating clock frequency. This information is obtained after the generation of the firmware.

FPGAs (Fig. 6.6 shows an image of an Arria 10 FPGA as an example) have been existing since the mid-1980s and if at the beginning they were only used in the telecommunications sector, their fields of application are now widely diversified. Indeed, several sectors requiring digital electronics are now integrating FPGAs into their solutions, such as aeronautics, automotive or even space, to name but a few. They are also used for ASIC prototyping and even more recently to speed up intensive computation in data centers at companies like Microsoft or Amazon.

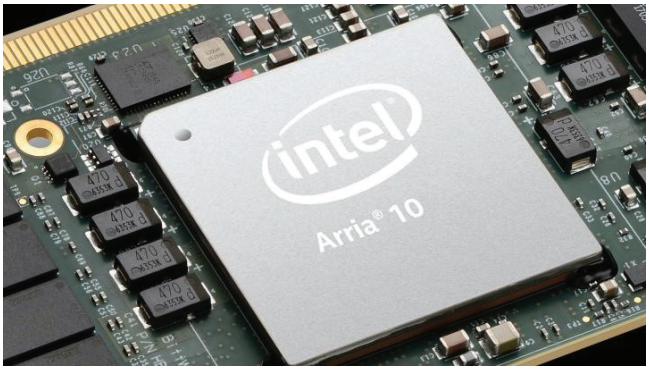


FIGURE 6.6: An Arria 10 FPGA chip.

6.2.2 FPGA IP

An IP (Intellectual Property) is, in the FPGA world, a logic block with a well-defined function. It is a module with inputs and outputs that can itself be connected to other modules or directly to the pins of the FPGA. These IPs are either created by the user for their application, or provided by the manufacturer for a number of standard functions to be implemented. (An introduction of how IPs in Intel FPGAs are handled is given in [47].)

The following sections will detail some classic IPs that will be used in the architectures described in parts 3 and 4.

6.2.2.1 FIFO

A FIFO (First In First Out) is a type of buffer that works on write and read requests. A write request adds a binary word to the FIFO and a read request brings out the first stored word.

Fig 6.7 illustrates the inputs and outputs required to achieve this behavior with signals for requests and data buses for input and output. Control signals can also be added such as *empty*, which goes logic high if the FIFO is not empty, and *full* which goes high if the FIFO is full.

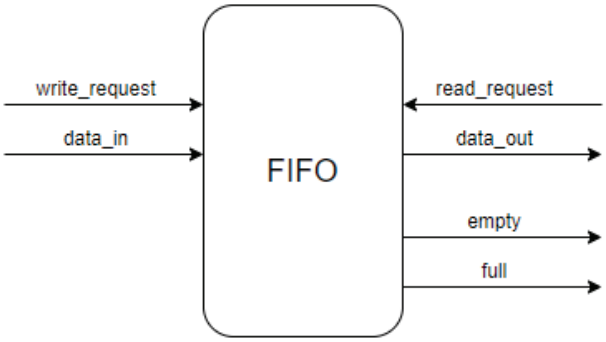


FIGURE 6.7: FIFO Schematic.

The timing diagram Fig 6.8 illustrates the behavior of a FIFO with a depth of 4. When D0 arrives, the empty signal goes low. Then, once 4 data are buffered, the full signal goes to 1 and the FIFO cannot accept any more values before it is flushed.

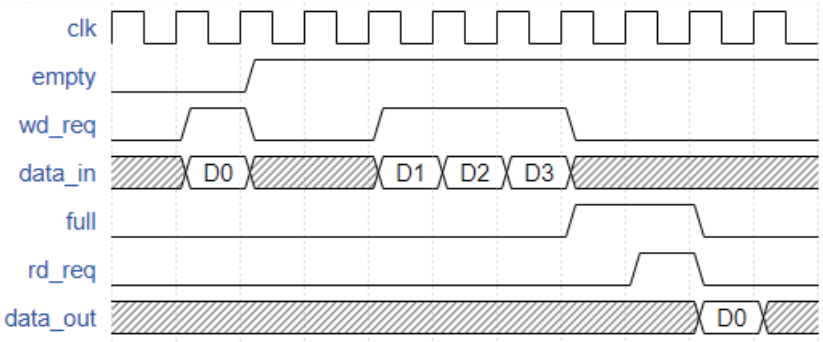


FIGURE 6.8: FIFO behavior chronogram.

6.2.2.2 Memory

A memory is an area for storing data more permanently than a FIFO. The memory therefore has cells each referenced by a unique address. These memory cells are accessible in reading and writing. Fig 6.9 illustrates the main signals needed for the functioning of a memory.

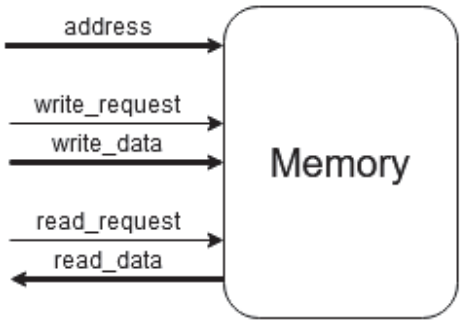


FIGURE 6.9: Memory schematic.

The timing diagram Fig 6.10 illustrates an example of writing to a memory. At the front of the 2nd clock cycle (2), the value 0xF is written at address 00 and at front 4, the value 0x9 is written at address 01. Then at fronts 7 and 8, addresses 00 and 01 are read. One clock period later the values read from memory are available on the *read_data* bus where the two previously stored values are found no matter how many times they are read.

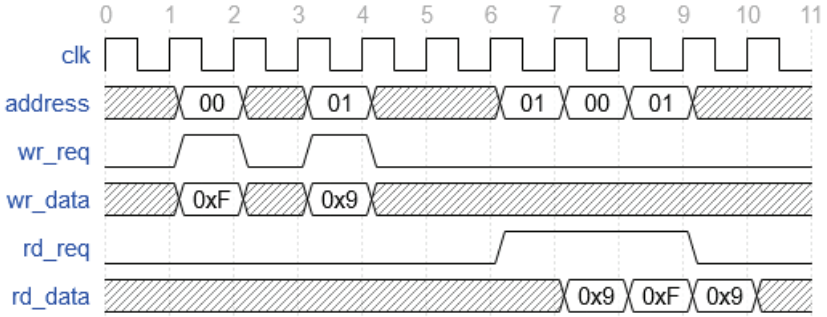


FIGURE 6.10: Memory behavior time diagram

6.2.2.3 Nios II

The Nios II is a specific IP performing the function of a 32-bit RISC-type processor. This is a so-called softcore processor because it uses the elementary resources of the FPGA and it is a logic block described in HDL. It can also be configured according to the type of functions and the desired operating clock frequency.

This IP integrates different communication buses to communicate with:

- A memory interface required for operation (memory instruction) made for an on-board system;
- HDL or classic IP devices.

The implementation of this IP Nios II requires the use of a dedicated design software tool called Platform Designer. This makes it possible to instantiate and interconnect Nios II with the IP devices required for the architecture under development.

Nios II is used like a conventional processor that executes C code for software applications. The performance is by no means equivalent to a current real computer processor, but is sufficient to achieve a system control communication interface. Fig 6.11 illustrates an application of the use of a Nios in an FPGA.

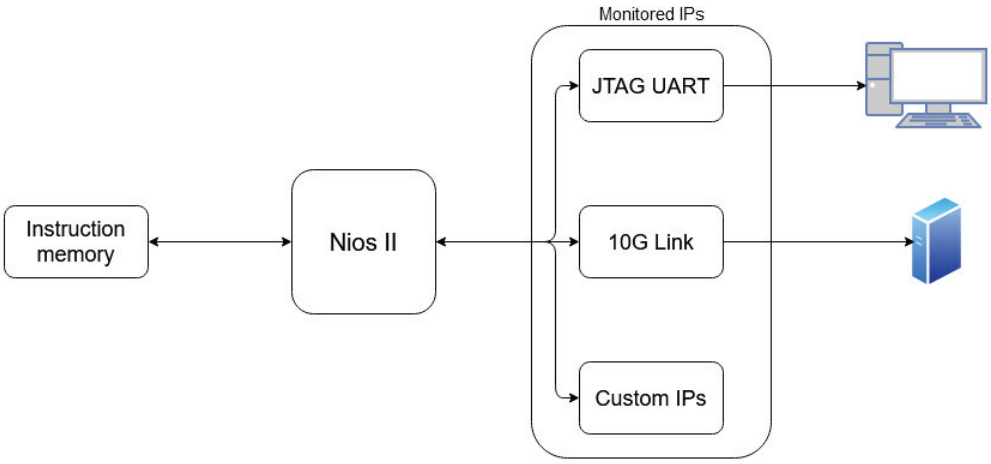


FIGURE 6.11: Nios II classic implementation in a FPGA architecture.

6.2.2.4 10GbE

A 10 Gigabit Ethernet link is a standard high speed network interface allowing Ethernet frame transfer at a transfer speed of 10Gb/s. It is an interesting protocol to implement in an FPGA because it allows direct communication between the FPGA card and a PC/server with a standard network card.

To be able to use this protocol, it is necessary to implement in FPGA the first two layers of the OSI model (Open Systems Interconnection), which are the MAC layer (Media Access Controller) and the PHY layer (Physical layer).

There is therefore the need for 2 IPs:

- IP MAC, allowing, among other things, to:
  - Manage access control to the transmission channel, including detecting collisions and re-transmitting packets if necessary
  - Give a physical address to the FPGA
  - Filter incoming packets based on their destination address
- IP PHY, used to prepare the frame sent by the MAC layer to be converted into an electrical or optical signal by serializing the bits in the case of transmission, or vice versa in the case of reception.

Fig 6.12 shows a typical case in which these 2 IPs are implemented to set up a 10GbE link in an FPGA.

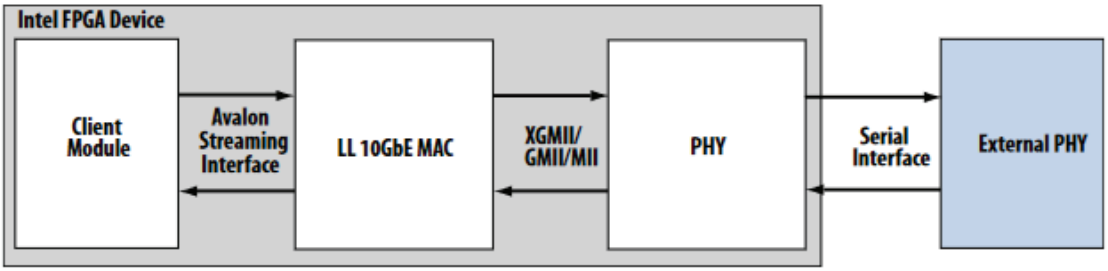


FIGURE 6.12: 10G Link implementation in a FPGA

6.2.3 Communication between modules in Intel FPGAs: the Avalon Interface

The Avalon interface simplifies system design by making it easy to interconnect components together in an Intel FPGA. This interface defines means of communication made for different types of applications such as:

- High speed data streams;
- Read and write accesses to memory.

These different standards are integrated into the components available in the classic IP catalog. However, it is also possible to design one's own components conforming to Avalon standards. This therefore allows greater flexibility in designs.

The two most used interfaces are the Avalon-MM (Memory-Mapped) interface and the Avalon-ST (Streaming) interface, so they will be discussed in more detail in the following.

6.2.3.1 Avalon Memory Mapped (MM) interface

The Avalon-MM interface is designed to implement read and write operations between master and slave components. This interface is often used with components such as a processor, memory, UART, etc.

This interface is synchronous and has different signals:

- A clock
- An address bus on a number of bits depending on the range to be addressed
- A byteenable bus which enables the transfer of the various bytes of the data bus (only active for transfers greater than one byte)
- A read bit set high during a read operation
- A write bit set high during a write operation
- A waitrequest bit is set high when the interface is unable to respond to a read or write request
- A writedata bus which is the data to be written from the master by the slave
- A readdata bus which is the data read by the slave for the master

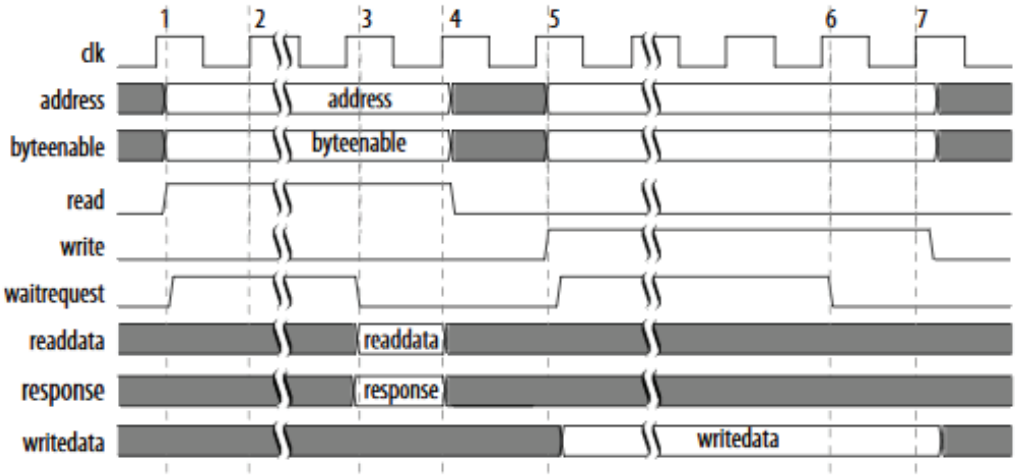


FIGURE 6.13: AMM Data transfer example

An example of data transfer is shown in Fig 6.13. The address and byteenable bus as well as the read bit change state on a rising front of the clock. The slave pauses the transfer by activating the waitrequest. Once it is activated, the cycle is on hold and the address bus, the byteenable bus, the read and write bits remain constant. The slave then disables the waitrequest once the request is accepted, and then sends the data over its readdata bus. The master only has to retrieve the data from the bus and the transfer is completed. Another cycle, write this time, then occurs in a manner very similar to the previous read cycle, with the exception that the slave retrieves the data at the end (clock front 7) on the writedata bus.

6.2.3.2 Avalon Streaming (ST) interface

The Avalon-ST interface is used by components handling a high unidirectional data throughput. The data comes from a component implementing the so-called "Source" part of the ST interface which sends the data to another component where the so-called "Sink" part is implemented. A data flow is established in between the Source and the Sink.

The interface implements various signals such as:

- A clock
- A ready bit that the Sink sends to the Source to indicate that it can receive data
- A valid bit sent by the Source so that the Sink knows when to receive the data
- A data bus that carries information
- An error bus for their detection
- A bus channel to identify the Source channel

Fig 6.14 shows a typical use case of the Avalon-ST interface. The Source interface sends valid, data, error, and channel signals to the Sink interface. The only signal that the Sink interface can send back to the Source is the ready signal.

It is possible to operate the interface in a packet transfer mode. This requires the implementation of two additional signals. These signals are startofpacket and endofpacket, and allow a Sink block to

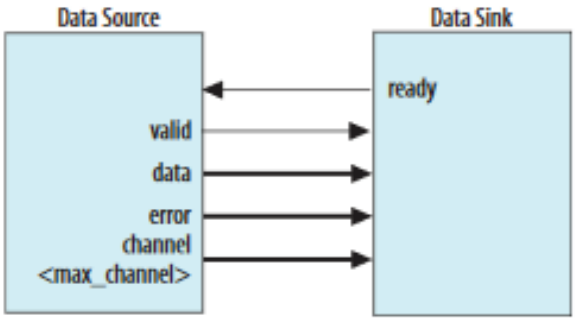


FIGURE 6.14: AST components connections

synchronize with the constant flow of incoming data. These signals are respectively set high at the start of the transfer and then at the end of the transfer as shown in the timing diagram Fig. 6.15.

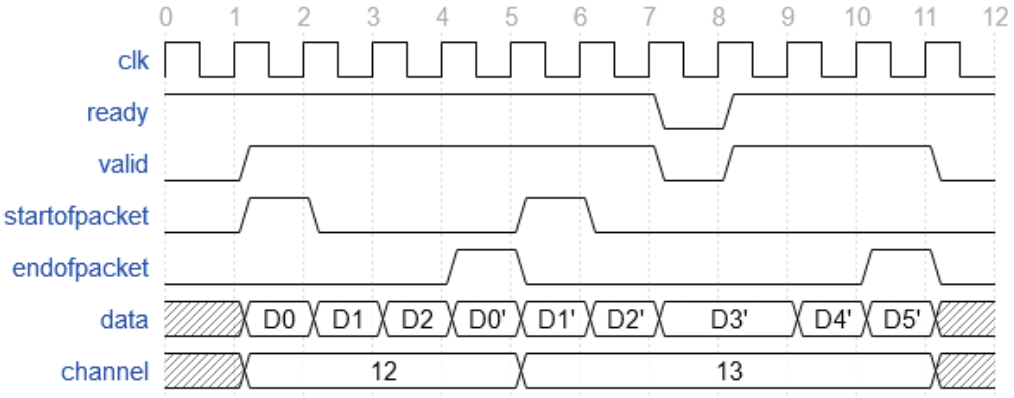


FIGURE 6.15: AST time diagram example.

6.3 Architecture for compressed data flow

In the same way as the data decompression, the TP search has been implemented on FPGA boards that receive the 10 GbE optical links from the μTCA crate of the acquisition. The architecture presented in this part will thus be the one implemented to treat the data of a whole crate and it will be necessary to duplicate this one on the same card to consider concentrating the data.

The study presented here was conducted on a compressed data stream performed at the level of AMCs. Due to the nature of the compression algorithm, the data are compressed channel by channel, which implies that the decompression must be performed in the same way.

In order to treat the data from an entire crate in real-time, the FPGA must be programmed to process 640 channels simultaneously. Thus one first needs to divide or demultiplex the stream into 640 distinct processing chains corresponding to each readout channel.

Fig. 6.16 shows the architecture implemented in the FPGA for the trigger primitive search in the compressed dataflow. The data from a given μTCA crate arrive to the analysis module via the 10GbE link. They are in the form of a 64-bit bus clocked at 156.25MHz.

The Primitive Offload Engine (POE) plays different roles depending on the type of acquisition. During normal acquisition managed by a central run control, data are duplicated at the level of the MCH by doing port mirroring of the 10GbE output link (spy mode for the FPGA board). POE then just

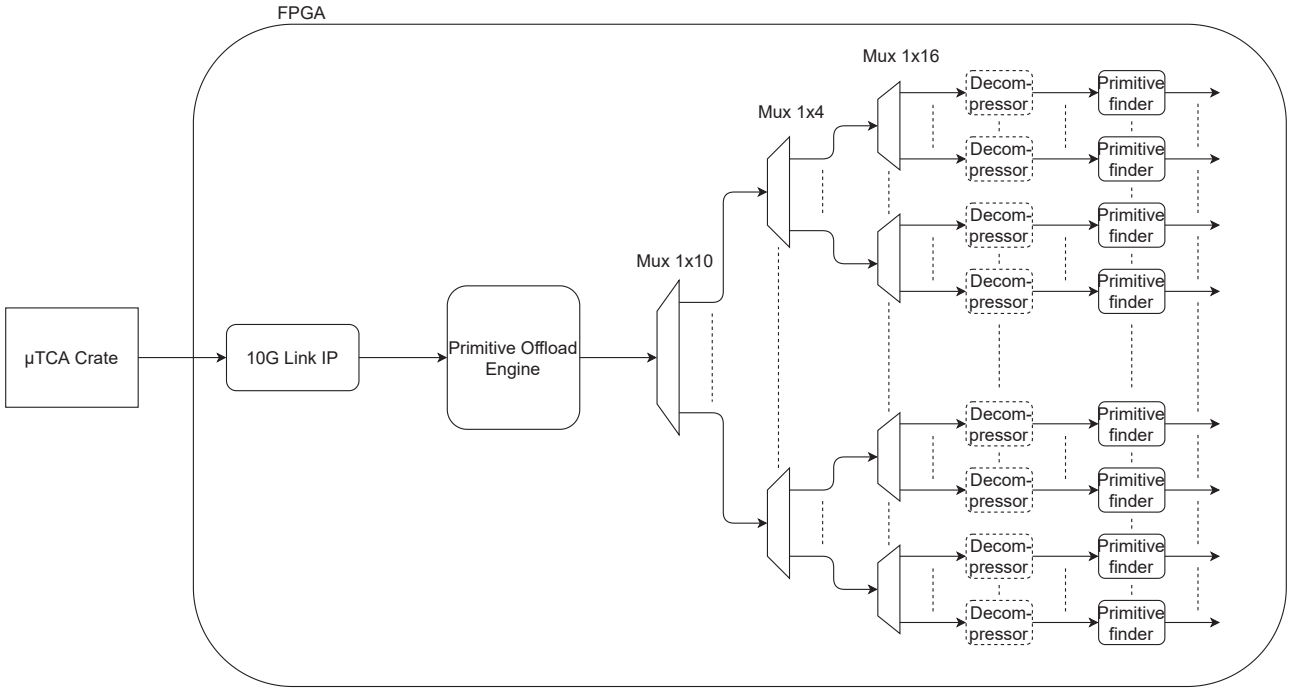


FIGURE 6.16: Compressed dataflow architecture

transmits the data packets to the downstream modules and keeps track of control packets exchanged between the AMCs and the data storage buffers. However, POE can also be configured to operate in a test mode, where it launches and controls the acquisition by establishing a network dialogue with a μTCA crate.

The data transmitted by POE then go through a cascade of three successive stages of demultiplexers, which allows to divide the incoming data flow into 640 sub-streams each carrying samples for a given readout channel. The number of stages is determined by the fact that the demultiplexers used in the implementation are the standard streaming components, which are limited to a maximum of 32 outputs. After demultiplexing the data for each readout channel, they are decompressed and analyzed by a Primitive Finder module. The architecture thus requires the implementation of 640 pairs of Decompressor and Primitive Finder modules.

The implementation and functionality of the Decompressor modules has been discussed already in section 5.3. In the following, implementations of POE, demultiplexing stages, and Primitive Finder modules will be described.

6.3.1 Primitive Offload Engine (POE)

The POE communicates with the μTCA crates through the 10 GbE link. This link takes care of the physical layer and the MAC layer of the 10 Gigabit Ethernet protocol, so the POE only has to work with Avalon ST interfaces to manage the flow.

As shown in Fig. 6.17, the data exchanged with the crate are sent through RX IN and TX OUT. An incoming packet first passes through a "Switch" module which determines in real time if it is a data packet or a control packet (e.g., an ARP request or an acknowledgement packet - ACK) by decoding the packet header. The data packets are transmitted without delay to the demultiplexer ("Demux Top") for the processing, while the control ones are redirected into a memory by passing through a FIFO and a DMA transfer. These packets are then analyzed by the Nios processor which acts according to their content. In the case of a test setup where the card itself initiates the data transfer,

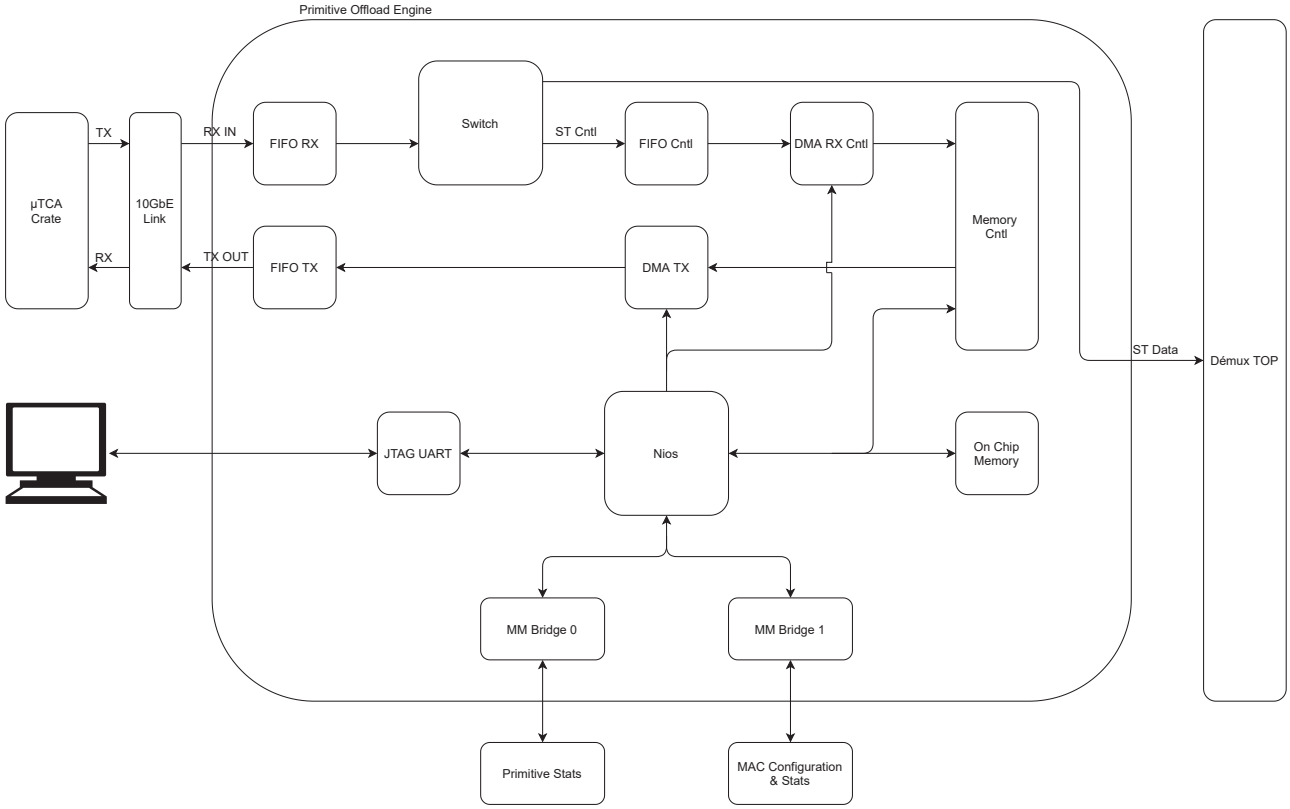


FIGURE 6.17: Primitive Offload Engine (POE) architecture

it responds to the chassis after the ARP frames have been sent, i.e., the Nios creates the response packet, places it in the control memory ("Memory Cntl") and initiates a DMA TX transfer to send the created packet to the crate. If the data transfer is initiated in another way and the card is in a spy mode, the Nios will wait for the ACK packets to know when a complete drift has been transferred and therefore analyzed.

The user can dialogue with the Nios via the JTAG UART interface. From this interface, the user has access to the memory spaces, via Avalon MM (Memory-Maped) interfaces, on the one hand, to the statistics of the primitives and, on the other hand, to the configuration and statistics of the MAC of the 10G link thanks to Bridge 0 and 1. The processor therefore allows global control of the application.

6.3.2 Demux Top (Demux 1x10)

Once the data has passed the POE stage, it arrives in the "Demux Top" module (Fig. 6.18) which makes a first division of the flow. More precisely, the incoming stream is divided into 10 sub-streams such that each output corresponds to the data generated by a single AMC.

The incoming data packets first go through the "Extract Board Number" sub-module. It analyses the first words of a packet containing the AMC board number from which they come. This number is then used to fill the channel signal of the Avalon ST bus which enters the 1x10 demultiplexer. This signal then allows to know on which output the incoming flow will be redirected. Once the first demultiplexing stage has been passed, the stream enters the "Demux Board" sub-module where the rest of the demultiplexing and processing are carried out.

In addition, an Avalon MM Bridge is present to be able to read the statistics available for further processing.

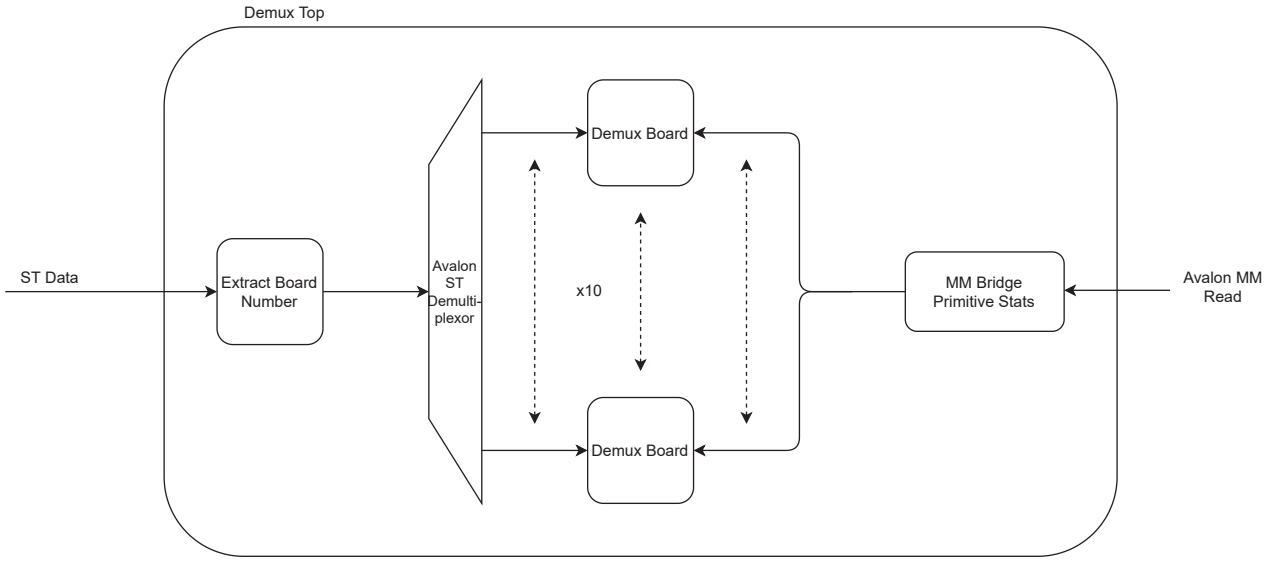


FIGURE 6.18: Demux Top module

6.3.2.1 Demux Board (Demux 1x4)

In a similar way to the division of the flow that takes place in the "Demux Top", the flow is divided once again, by 4, by the "Demux Board" module (Fig. 6.19).

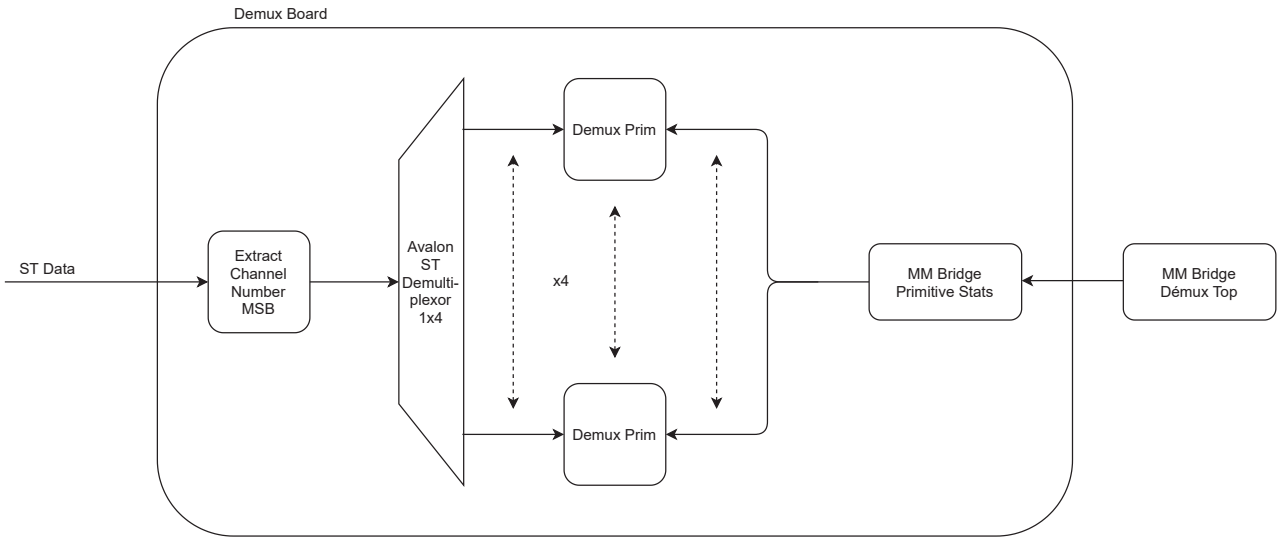


FIGURE 6.19: Demux Board module

The main difference with the "Demux Top" module is that the sub-module which will determine the channel signal extracts this time the 2 MSB (Most Significant Bits) in the channel number which is also included in the data header in order to be able to redirect the flow on 4 outputs.

The data are demultiplexed into 4 outputs based on the 2 MSB (Most Significant Bits) of the AMC channel number also contained in the data packet header. The first output corresponds to the flow of channels 0 to 15, the second to channels 16 to 31, the third to channels 32 to 47, and the fourth to channels 48 to 63. The statistics of the further processing can then be read by the successive MM Bridge.

6.3.2.2 Demux Prim (Demux 1x16)

The last demultiplexing stage is carried out in the "Demux Prim" module (Fig. 6.20), which realizes the division of flow at the level of each AMC channel.

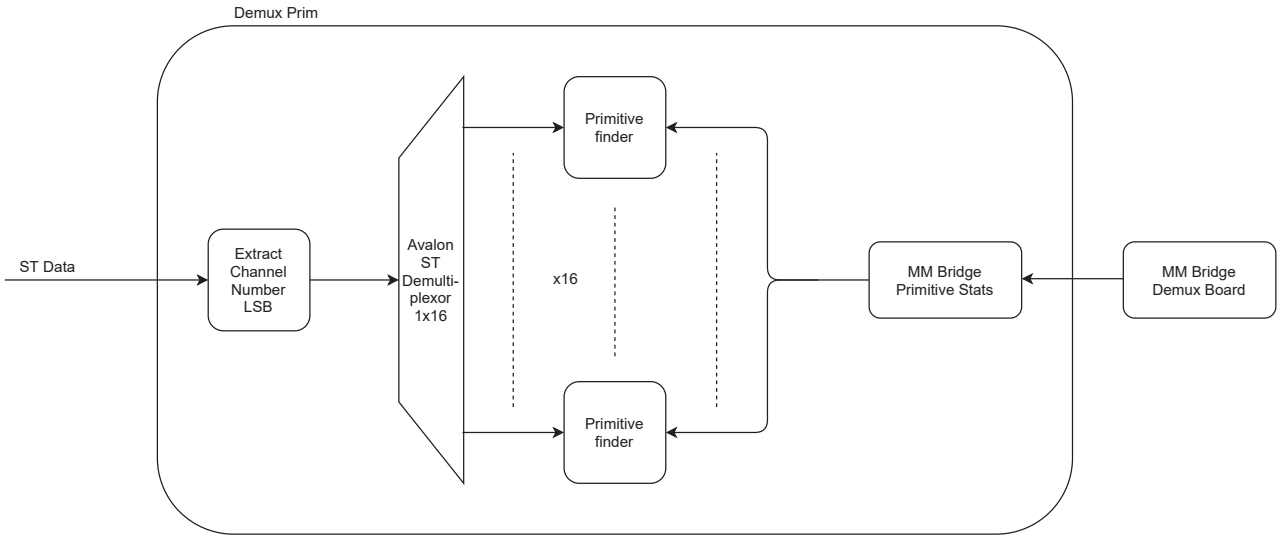


FIGURE 6.20: Demux Prim module

The beginning of this module is similar to the two previous ones, except that the number extracted to fill the channel signal is the 4 LSB (Less Significant Bit) of the AMC channel number. The resultant divided flow then either goes to:

- the "Primitive Finder" modules in the case of the uncompressed data flow,
- the "Decompressor" modules and then "Primitive Finder" modules in the case of the compressed data flow.

The MM bridges present successively in the different stages of the architecture are used to read primary statistics which allow a functional validation of the model. Indeed, the final goal is to be able to transmit all the primitives that the architecture detects, however, for certain reasons that will be detailed in the following (lack of resources), the choice was to keep only certain information. This information is available through an Avalon MM interface dedicated to each of the "Primitive Finder" modules which has a memory space of 4 addresses that can be read as follows:

- In "00": A counter of the number of detected primitives.
- In "01": The starting sample number of the last detected primitive.
- In "10": The end sample number of the last detected primitive.
- In "11": The sum of the samples contained in the last detected primitive.

This information is then accessible by the various successive Bridges which are present at all the division stages of the flow.

6.3.3 Primitive Finder module

6.3.3.1 Architecture

The (decompressed) channel data are processed within the Primitive Finder module, which looks for the potential trigger primitives. The search is performed by comparing the data samples to a given threshold taken relative to the pedestal baseline of a given channel. Each Primitive Finder module can therefore be configured with a threshold and a pedestal of a readout channel.

The only problem with this processing is its implementation in real time, and not in real time of the acquisition (12bits @ 2.5MHz), but in real time of the speed at which the data arrives, i.e. at the speed of the 10 GbE link (64bits @ 156.25MHz). Several solutions are therefore possible to achieve real-time processing:

- the processing rate is accelerated to process the samples one after the other, but it would be necessary to go up to a frequency of 625MHz (4×156.25MHz because 4 samples of 16bits in a word of 64bits)
- the data are first stored and then processed by the architecture at a lower frequency for each channel. However, for example, in the case of a protoDUNE drift that sends 10 000 samples, 640 memories of $10\,000 \times 16$bits will be needed, which corresponds to 97Mb, whereas FPGAs such as the Arria 10 or the Stratix V have only 50Mb of internal memory. Even if we adapt the packet size, this would be a very expensive solution in memory.
- the data are parallelized again in the Primitive Finder module so that the processing rate is not too high. This solution has been chosen to be as close as possible to real time and will be explained in the following.

The principle of the treatment carried out by this module is as follows:

- The stream is divided into 4, each 64-bit word corresponds to 4 words of 16 bits, i.e. 4 ADC samples.
- Each sample is evaluated by a module ("Comparator" in Fig. 6.21) which evaluates its position with regards to the median and the threshold (to detect a TP),
- Each module sends resulting signals such as:
  - The position relative to the median (2 bits: 0b11 = -1, 0b00 = 0, 0b01 = 1, 0b10 = no median update).
  - The position relative to the threshold (1 bit: 0b0 = below threshold, 0b1 = above threshold).
- The resulting signals are sent to a referee ("Referee" in Fig. 6.21) who updates the median and manages the information relative to the primitives (start, end, integrated value of the signal),
- The information about the detected primitives is then sent through an Avalon-ST interface to a memory and the median is looped back to the input of the sample evaluation module.

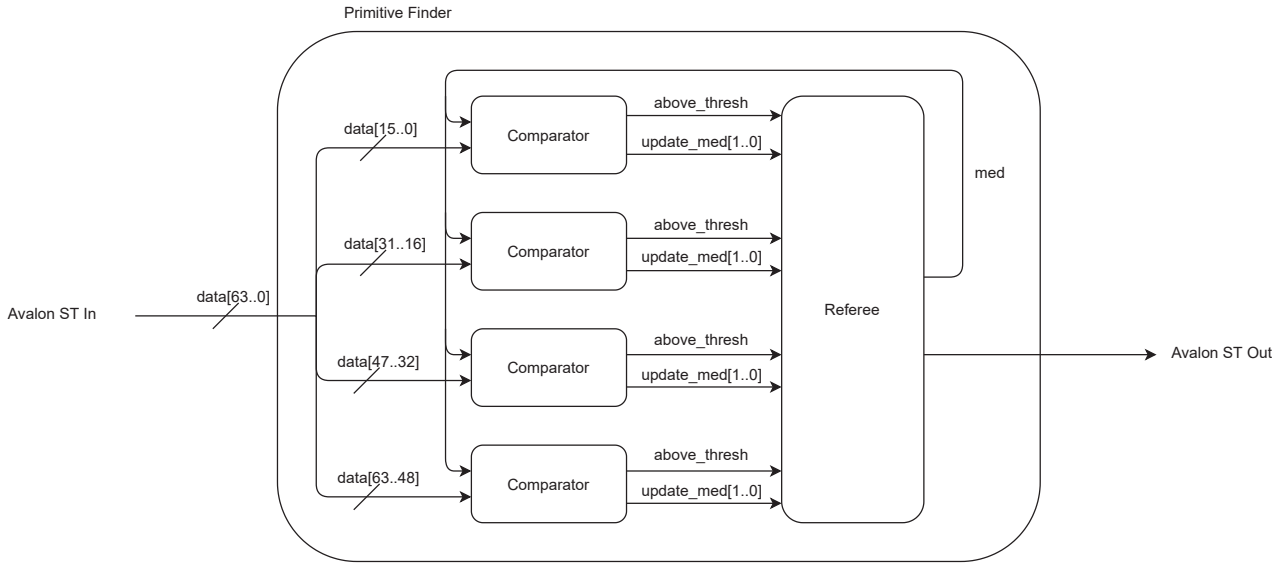


FIGURE 6.21: Primitive Finder module

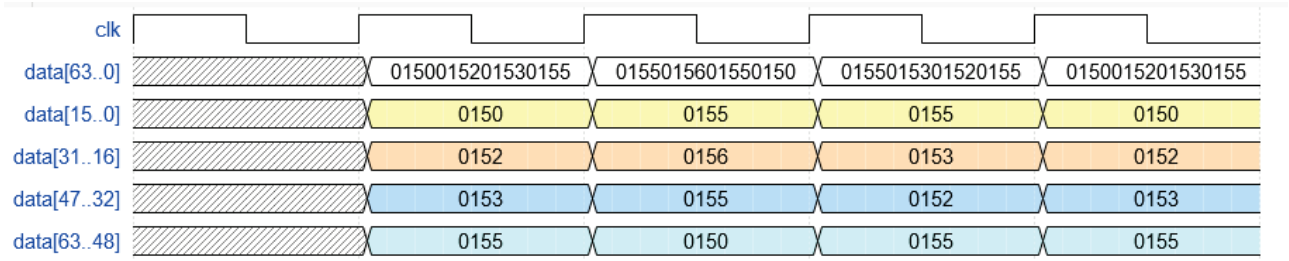


FIGURE 6.22: Splitting incoming data flow

6.3.3.2 Model simulation

The first step of data manipulation is to split the 64-bit stream into 4 buses of 16 bits. Fig. 6.22 shows this step:

The second step is the calculation of the median which works as such:

- At the very beginning, the median is initialized with the pedestal value.
- A treatment is done in the "Comparator" modules.
- The median is updated in the "Referee" module.
- The median is looped back to the input of the comparators.

This explains why the value of the median has a delay of 2 clock strokes, i.e. 8 samples, compared to the values on which the "Comparator" modules work. Fig. 6.23 illustrates this behavior.

When the first data arrive, the *update_med* signals are at the value 0b10 (which is the value that tells the "Referee" module not to update the median). Then once the first burst of samples is taken into account (`0x0151`, `0x0153`, and `0x0155`), the *update_med* signals change to the value 0b01, which means that the value of the median was lower than the values of the inputs so each sample sent back the code to increment the median, which, in this example, was initialized to the value `0x12C`. With all 4 samples being above the median, 2 clock periods later it is increased by 4 units (`0x130`). In this

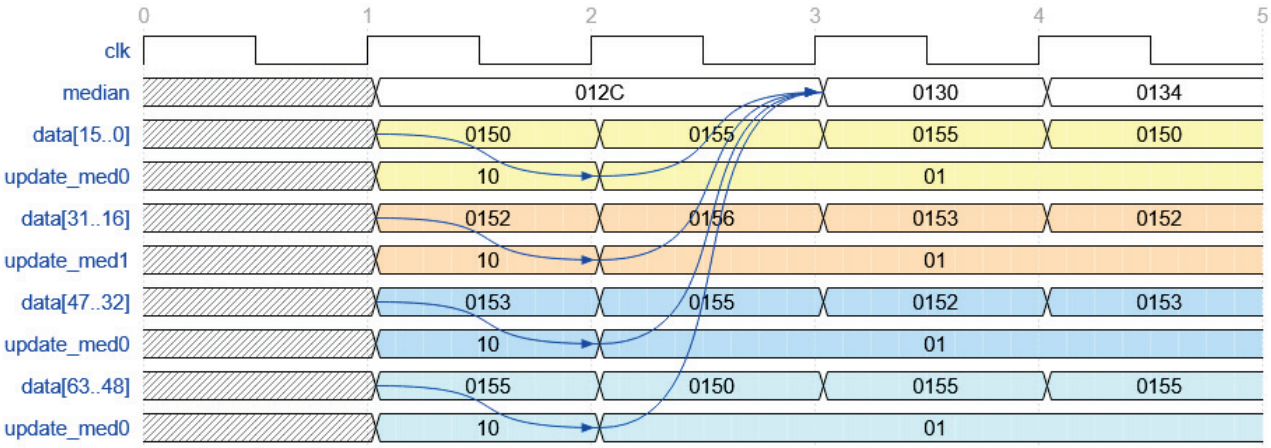


FIGURE 6.23: Median update

example, the median is updated at each sample. For more stability in the following, it will often be either directly fixed to the pedestal or updated every 10 samples above or below its current value.

In parallel to the median update, the "Comparator" modules evaluate whether the input values exceed a certain threshold (relative to the median). Each module has an *above_threshold* output that changes to 0b1 when the values become above the threshold. Also, when *above_threshold* is 0b1, the input value is passed to the "Referee" module so that the information about the primitives is stored (Fig. 6.24).

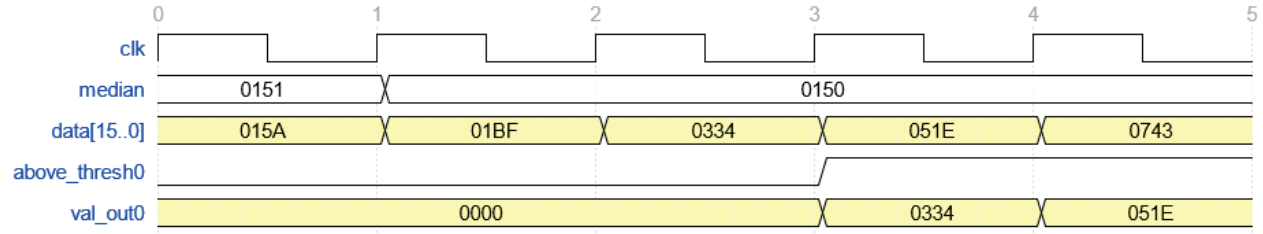


FIGURE 6.24: TP detection

Finally, after a primitive has been detected, i.e. after all *above_threshold* signals have dropped to 0b0, the information retrieved on the detected primitive is sent through the output Avalon ST interface. Fig. 6.25 shows the output bus when sending the detected primitive in the simulation data.

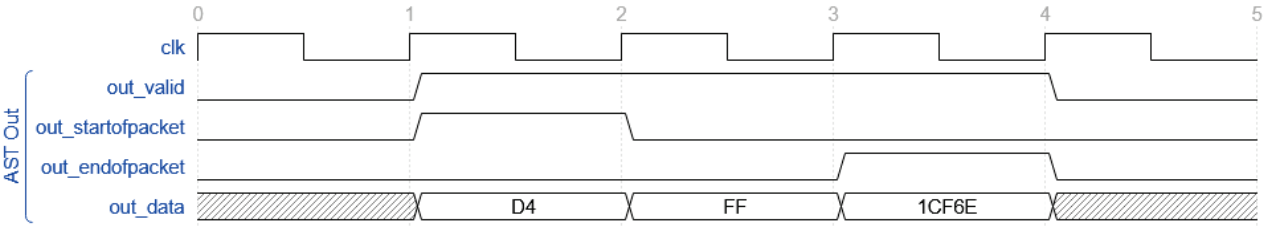


FIGURE 6.25: Avalon ST TP Info Out

In this example, a TP appears at a sample `0xD4` (212) and lasts `0xFF` (255) samples with the sample sum within this window being `0x1CF6E` (118638). This TP is an artificial example and in the detector the signals will generally be smaller.

6.3.4 Resources evaluation and conclusions

Parallelizing the processing of 640 channels on an FPGA board presents a challenge for the design synthesis especially in terms of timing constraints. Table 6.1 shows the resource allocation report for Stratix V FPGA.

Device family	Stratix V
Logic Utilization (in ALMs)	249,229/262,400 (95%)
Total block memory bits	1,652,350/52,572,160 (3%)

Table 6.1: FPGA resources used

As the report shows, the Stratix V is almost full in terms of logical resources. In addition, in order to make the architecture running for a test, the decompressors upstream of the Primitive Finder modules were removed and the data stream sent was adapted. Indeed, as the whole architecture was not able to be implemented on the FPGA board, the data was sent directly uncompressed in order to be able to validate the behavior of the design.

It is also for this reason that in the part detailing the global architecture, the results of the Primitive Finder modules can only be read in a degraded way through an Avalon MM interface. The board was too full to add a 10GbE or PCIe interface to send all the primitives directly to another location.

The architecture is therefore very resource-intensive and it is unrealistic to duplicate the processing of several crates on the same FPGA board.

Improvements to this version of the architecture were nevertheless made to assess the real limitations that this architecture may have once it has been optimized. To achieve the optimization of this architecture, one must work to reduce the time when the resources allocated to the various modules implemented in the FPGA are idle. That is, each module has as little downtime as possible. In order to understand this point, it is interesting to look at the way and the timings in which data packets from a chassis arrive and are processed in the architecture.

Figure 6.26 illustrates the scheduling of data packets to be processed. Each packet contains data from a single channel across the 640 channels of a crate.

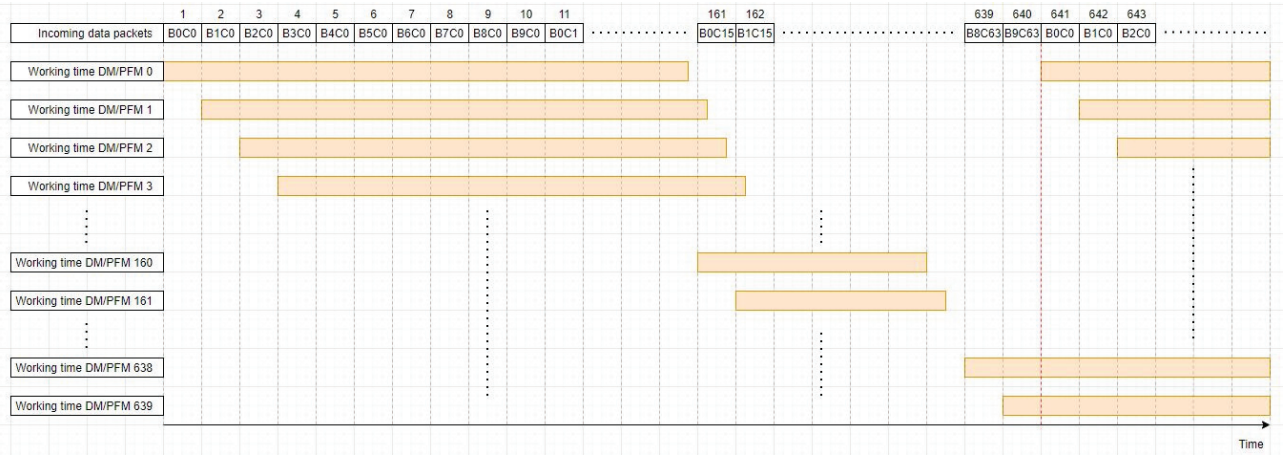


Figure 6.26: Scheduling of data packets to be processed for decompression

The structure of these 640 packets is called a frame. Each DM/PFM (Decompressor Module / Primitive Finder Module) pair processes only one of these packets per frame, the processing time of which is shorter than the total duration of a frame. The processing time corresponds to a buffering of the packet, its decompression and a search for primitives. This duration is longer than the reception

time of a packet but remains shorter than the duration of a frame. This implies a dead time of the DM/PFM modules until the arrival of the data of the next frame.

One optimization envisaged is therefore to use this dead time of the DM/PFM modules to make them process other packets of the same frame. This would reduce the number of DM/PFMs and save in terms of logic resources and memory on the FPGA.

It should also be noted that the decompression time, due to the nature of the processing algorithm, is not fixed for all packets. Indeed, this type of decompression is dependent on the data, the higher the compression rate, the longer the decompression will take. However, these times are still all relatively close and after measurements carried out in the laboratory, it was demonstrated that only 160 pairs of DM/ PFM were sufficient to process a frame in real time and therefore a continuous stream of compressed data coming from a crate. This new optimized architecture has been implemented on the FPGA board based on the Stratix V.

Table 6.2 shows the total resources that this new architecture takes. The FPGA is, this time, about 50% occupied in terms of logic as well as in terms of memory.

Device family	Stratix V
Logic Utilization (in ALMs)	142,350/262,400 (54%)
Total block memory bits	28,526,592/52,572,160 (54%)

Table 6.2: FPGA resources used for optimized architecture

If the goal is to process multiple crates on the same FPGA board, it would still be necessary to use a larger FPGA (like the Arria 10). However, even if the processing of two crates could be implemented on the Arria 10, this solution seems difficult to implement (number of cards, cost, data concentrator ...) for the DUNE far detector which will include several hundred crates. Decompression takes time (larger than the duration of a packet) and lots of resources. Another approach would be to perform the trigger primitives search on an uncompressed stream to overcome the problems of decompression and save a lot of resources.

The last notable point of these analyses is that the architecture working without the decompressors works entirely in streaming. The internal memory of the FPGA being only used at 3% (see table 6.1). This means that there is no need to buffer the data so that they can be processed and the parallelization at the level of a 4-samples module works. The following section will therefore explore the possibility of overcoming the decompression phase and checking if other solutions are possible on a raw data stream.

6.4 Implementation on uncompressed data-flow

In the previous section it has been shown that it is possible to set up an online primitives search associated to a decompression stage. However, this design is in practice difficult to implement when considering the resources available in the current class of powerful FPGAs (Arria 10, Stratix V).

The following figure (6.27) shows that much shorter processing times may be achieved by removing the decompression modules from the architecture presented in the previous section. This processing is done in real time (at the speed of data reception) and without latency.

The working time of each of the **PFM** does not overlap, so the idea was to develop a single **PFM** capable of handling the entire input data stream continuously. This is the architecture that will be presented in this part.

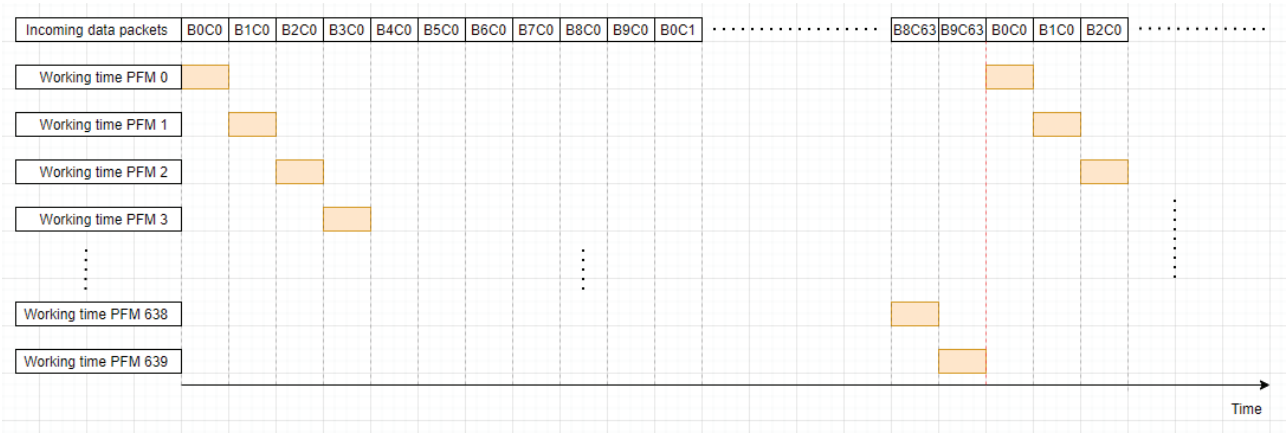


FIGURE 6.27: Operation time of the PFM modules

6.4.1 Architecture

The architecture working on the uncompressed data stream can be simplified as shown in figure 6.28.

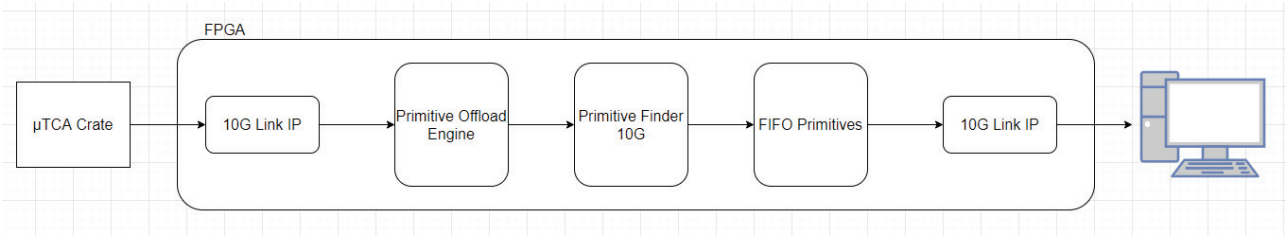


FIGURE 6.28: Trigger Primitives architecture on uncompressed data stream

This architecture will always need the POE module which can communicate with the crate and redirect the packets according to the protocol. On the other hand, all the data packets will go directly to a unique Primitives Finder 10G module which will have the task to process them all in real time. The outcome of the **PFM10G** will be stored in a FIFO (**FIFO Primitives**) and then sent to a dedicated DAQ machine by another 10G Ethernet link in order to centralize the results.

6.4.1.1 Architecture of the 10G Primitives Finder

This implementation, although schematically simpler, remains complicated to implement. Indeed, due to the algorithms used, each channel has a different "**context**" which has to be taken into account. The previous architecture did not have this problem because having one module per channel allowed for a hardwired storage of the **context** and data continuity. In this version, where the packets from different channels will arrive one after the other, the context must be loaded before starting to receive the data. The **context** consists of:

- The median value for each channel;
- The information if a primitive finding was in progress at the end of the previous packet corresponding to the same channel:
  1. The sample number where the signal goes above the threshold;
  2. The sum of the samples so far.

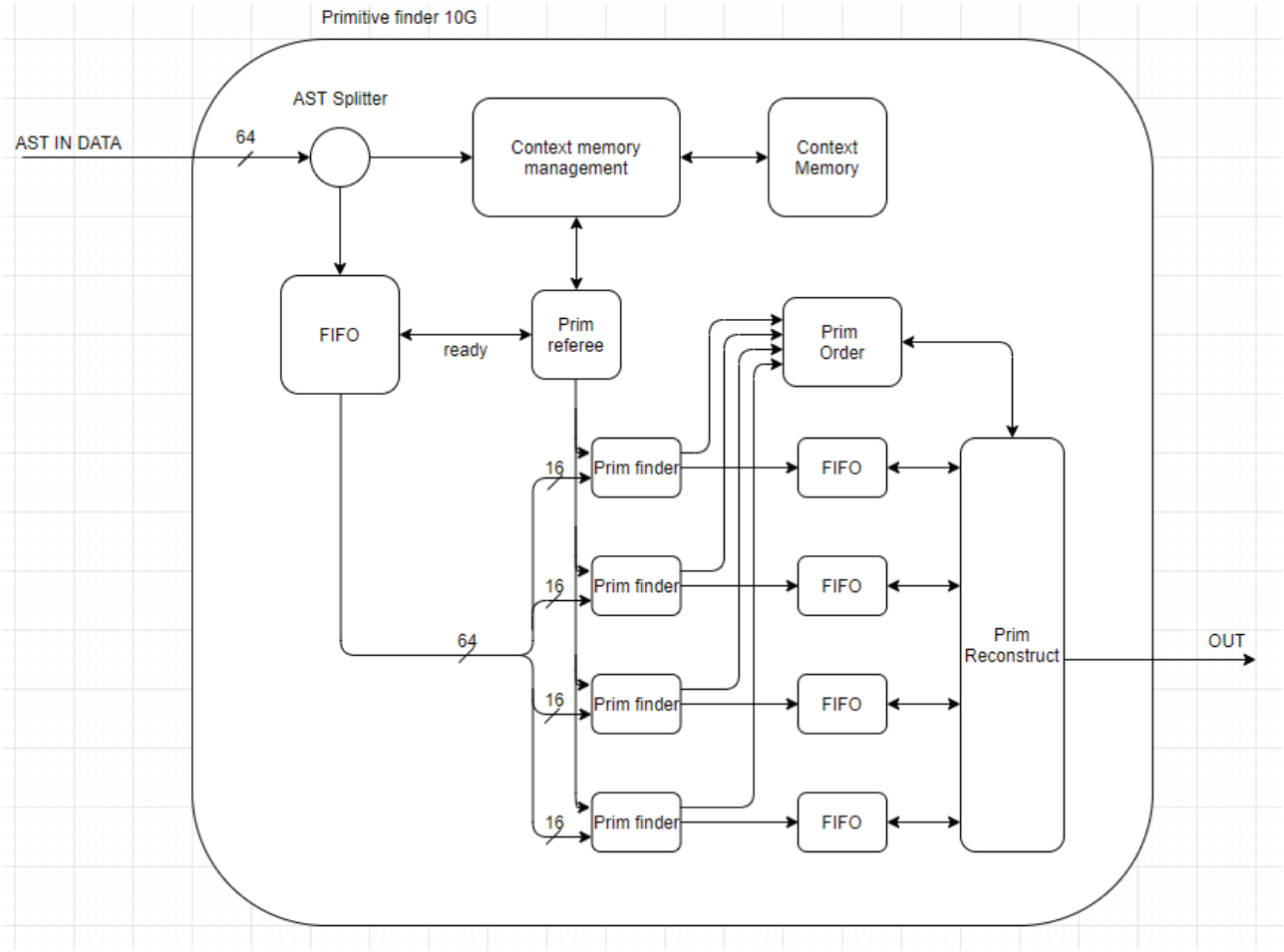


FIGURE 6.29: Architecture of the 10G Primitives Finder

Figure 6.29 then illustrates the architecture that has been put in place to meet these needs.

Data arrive from the 10G link in the form of an Avalon Streaming bus, then they are duplicated in an **AST Splitter** module. This module will send the data to a FIFO and to a module called **Context Memory Management (CMM)**. It is thanks to these blocks that the problem of the context for the processing will be solved.

The principle is the following:

- Data arrive at the same time in the FIFO and to the **Context Memory Management module**
- The "**ready**" signal of the AST output bus from the FIFO is at 0, data are stored without going out for the moment
- The **CMM** module identifies from which AMC card and which channel the packet comes from
- Once this information has been acquired, the **CMM** recovers the corresponding context in memory (memory that has been initialized with the values of the channel pedestals)
- The "**Referee**" module retrieves the context and sends this information to the Comparators and to the **Prim Reconstruct** and validates the reading of the data from the FIFO with the **ready** signal
- The FIFO can therefore send the packet already stored and the detection of the primitives can begin

These steps take about ten clock periods and therefore introduce a very low latency of around 64ns. This latency is only introduced at the start of data acquisition, namely when the very first data packet is received. This fixed latency therefore ensures that the input FIFO will not be saturated during the acquisition. The architecture then processes continuously samples at the rate of the 10G link.

Once the processing of a packet begins, 4 samples are evaluated at the same time by the Comparators as in the previous part. In case that these samples constitute a TP (Trigger Primitive), they are in a first time stored in dedicated FIFOs as well as the order in which the samples are stored in these FIFOs using the **Prim Order** module. The **Prim Reconstruct** module then interrogates the memory present in **Prim Order** to reorder the samples and builds the primitives with the good information.

Once the TPs are created, they are stored in a dedicated **TP FIFO** memory and wait to be transferred to another architecture which will take care of the processing of all the TPs. All of these steps are necessary to be as robust as possible to the different scenarios that may arise, as it will be detailed below.

6.4.1.2 Simulation of the PFM10G module

This section will detail the internal machinery of the **PFM10G** module. The first step carried out by this module is a duplication of the input data arriving with the **AST In** bus. This bus is duplicated using a standard Intel submodule named **AST Splitter** which produces in output two **AST Splitter** buses: **Out 0 and Out 1**. One of these buses is sent to the **CMM** module which will take care of fetching the context and the other will be used to temporarily store the data in a FIFO. These data will be stored for a short time which will allow subsequent modules of the architecture to be ready to process them. The fact that the other modules are ready to begin processing will be notified by the "**ready**" signal of the **AST Data** output bus of the FIFO which will be controlled by the **Referee** module (see figure 6.30).

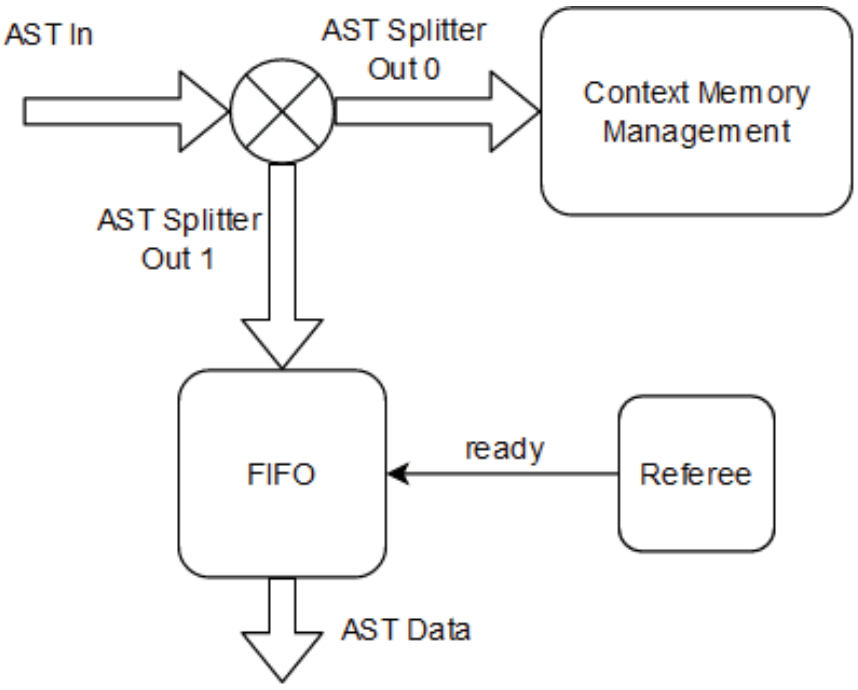


FIGURE 6.30: Data flow in the PFM10G module

This translates at the level of the signals as shown in the timing diagram of Figure 6.31: the data arrive in correspondence to the front 1 on the **AST bus In**, then they pass into the **AST Splitter** where

they are duplicated and shifted by one clock period on the outputs since this module is synchronous. Finally, when the "ready" goes to 1, data are output on the **AST Data** bus in the same order as they arrived, just shifted in time by a few clock periods.

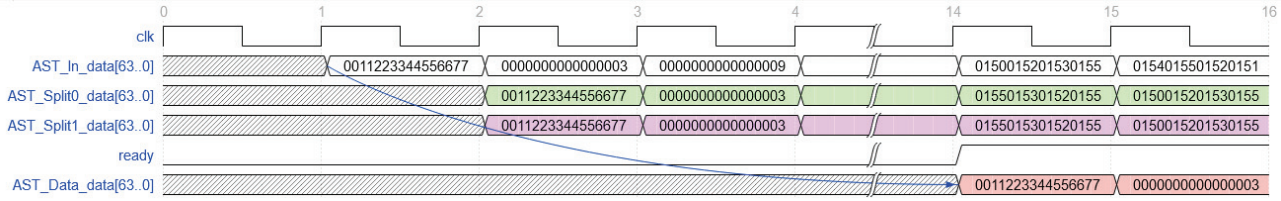


FIGURE 6.31: Architecture of the 10G Primitives Finder

It is then interesting to look at the interactions between the CMM, the Context Memory and the Referee to understand how the "ready" signal is generated (see Figure 6.32). The principle here is that the CMM will receive the packets "ahead" of the rest of the architecture. Indeed, as the packets will be stored first, the CMM will see the packets before the rest of the processing architecture (comparators, prim reconstruct, etc ...). During this time, it will look at the header of the data packet and retrieve information based on which channel that packet is coming. This information corresponds to the number of the AMC card in the μTCA crate (ranging from 0 to 9) as well as the number of the channel on the card (ranging from 0 to 63). These two fields are respectively contained in the second and third 64-bit words of the packet, the first word (0x0011223344556677) being the code identifying that the packet is a data packet.

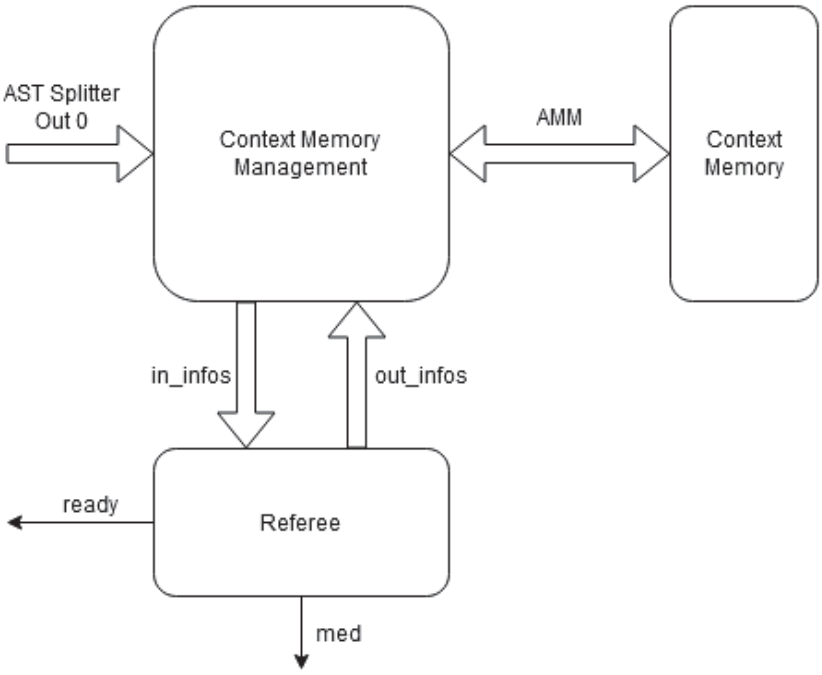


FIGURE 6.32: Interactions among the **CMM**, the **Context Memory** and the **Referee** and generation of the "**ready**" signal

Once this information is retrieved, the **CMM** can match the context address corresponding to the correct channel in the **Context Memory**. As explained previously, the **context** consists of 3 pieces of information: the value of the channel baseline, the start value of a potentially unfinished primitive and the sum of its values above the threshold up to now. Each channel therefore needs to have 3 fields to be retrieved from memory. It was chosen to represent each of these values with 32-bit words. As

a consequence, the memory needs $640 \times 3 = 1920$ 32-bit words. Figure 6.33 shows a diagram of the address and channel correspondence table.

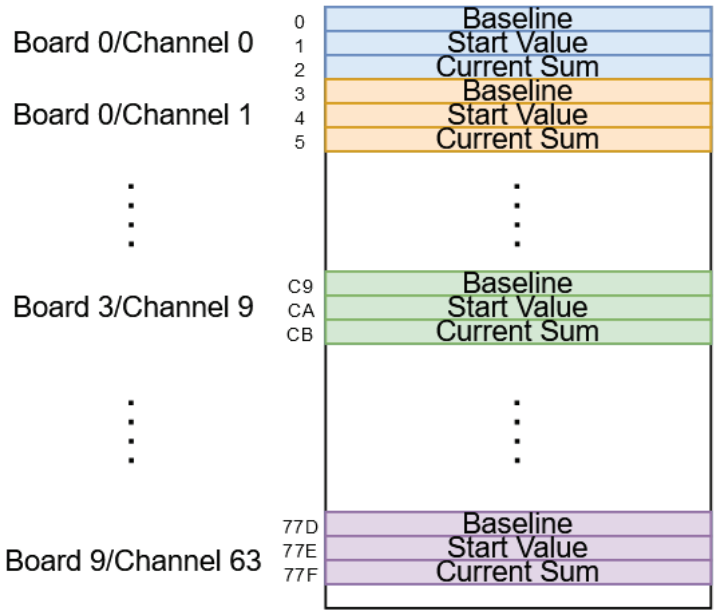


FIGURE 6.33: Organization of the context memory

The following time diagrams detail this behavior at two key points. The first one (Fig. 6.34) shows when the very first data acquisition package arrives in the processing architecture. In the example this package comes from the AMC 3 card and from channel 9, which would not be the case in reality. The first packet would a priori be the one coming from card 0 and channel 0. However, it is more interesting for illustrative purposes to consider another channel to clearly explain the address/channel correspondence in the memory.

The data begins to arrive in the **CMM** module via the *AST Split Out 0* bus starting synchronized at clock front 2 (with the signal *SOP* - Start Of Packet at 1 on the first 64-bit word of the packet). The first three 64-bit words of the packet contain the header:

- The identifier of a data packet `0x0011223344556677`
- The number of the AMC card `0x3`
- Channel number `0x9`

The address corresponding to this channel in the memory is therefore $3 \times 64 + 9 = 201$ or `0xC9`. From the moment these data are passed to the **CMM**, the read address of the context is reconstructed. Then a request is sent to memory at this address and the next two ones. Here the baseline value is `0x151` and the other two values are zero. These values are then transmitted to the Referee module using the bus *in_infos* containing two signals: valid and data [31..0].

Finally, when the Referee receives the information, it updates the value of the baseline (med) that will be used by the comparators and notifies that the TP search architecture can start data processing. This operation introduces latency just when receiving the first packet. As shown in Fig. 6.34, there are 6 clock periods between the moment when data arrive and when they leave the FIFO.

The same mechanism is repeated every time a new data packet arrives. The following timing diagram (see figure 6.35) illustrates the behavior at the end of the first packet and at the start of the next

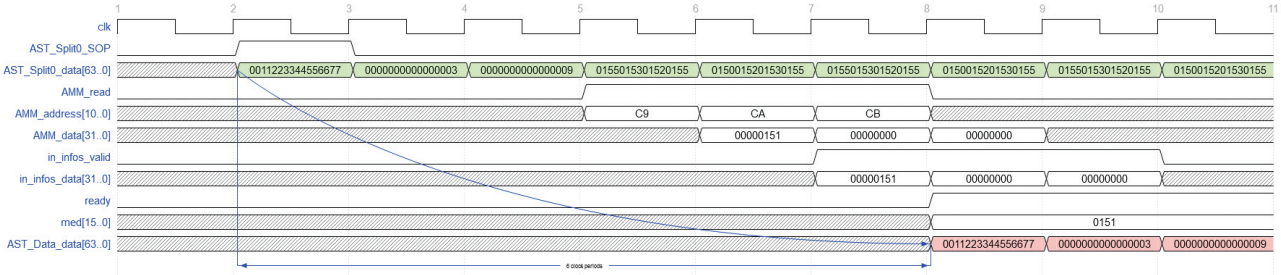


FIGURE 6.34: Time diagram corresponding to the treatment of the first DAQ packet

one. At the front 1004, the first packet ends its transfer (*EOP* - End Of Packet signal at 1 on the last 64-bit word). Then the new packet starts, in the same way as the previous one except for the channel number which is different.

Information relating to this channel is therefore retrieved from the memory and sent to the Referee to update the *ready* signal. It should be noted that when receiving consecutive packets, performing as well the search for the corresponding contexts does not cause an accumulation of latency for processing the data.

Interestingly, the latency here is always 6 clock periods. This is because the latency introduced in the very first packet is maintained but does not increase because the context of the next packet is loaded while the last words of the previous packet are being pulled out of the FIFO.

This ensures that the FIFO will never be saturated by this operation. In addition, this FIFO does not need to be very deep to be able to buffer the samples for the time needed to retrieve the context, 6 words are always sufficient in principle. However, to avoid locks in other parts of the architecture, this memory, has, for the time being, been oversized (128 words).

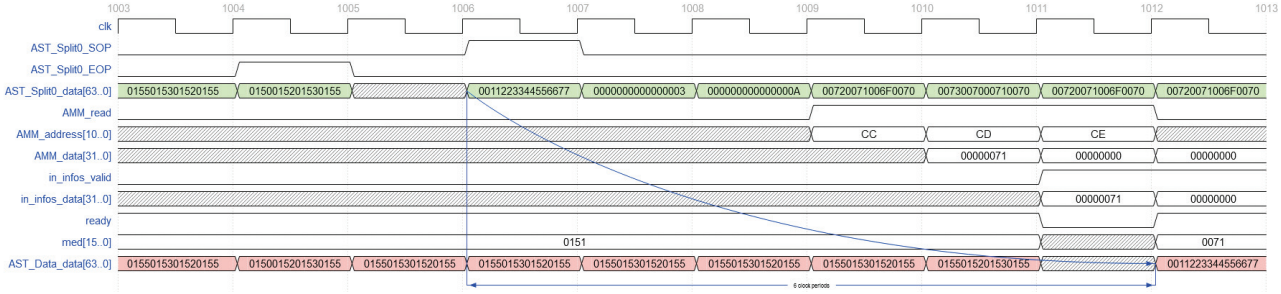


FIGURE 6.35: Time diagram corresponding to the treatment of the packets following the first one

In this part, the principle will be similar to what presented in the previous section on the **PFM**. Namely, the 64-bit words will be split at the 16-bit sample level. These 4 samples will then be compared, in parallel, to a threshold in their respective comparators. Then a module named Prim Reconstruct will take care of recovering the samples that will have been detected above the threshold and transmitting them in a previously defined **TP** format.

For more robustness, small buffers (see figure 6.36) have been installed in different places of the processing in order to allow time for reconstruction (with the **FIFO AT** - Above Threshold) and for time for transmission (with the **Primitives FIFO**). Another FIFO is also present in the **Prim Order module**, its interest is detailed below.

The following timing diagram shows the example of a packet where, on the data arriving on the fronts 4 and 5, a TP is detected with 4 consecutive values above a threshold of 10 ADC counts above the baseline of `0x151`. This corresponds to a value above the threshold on each of the comparators. These values will be stored in their respective **FIFOs AT** and the order in which they arrived will be stored

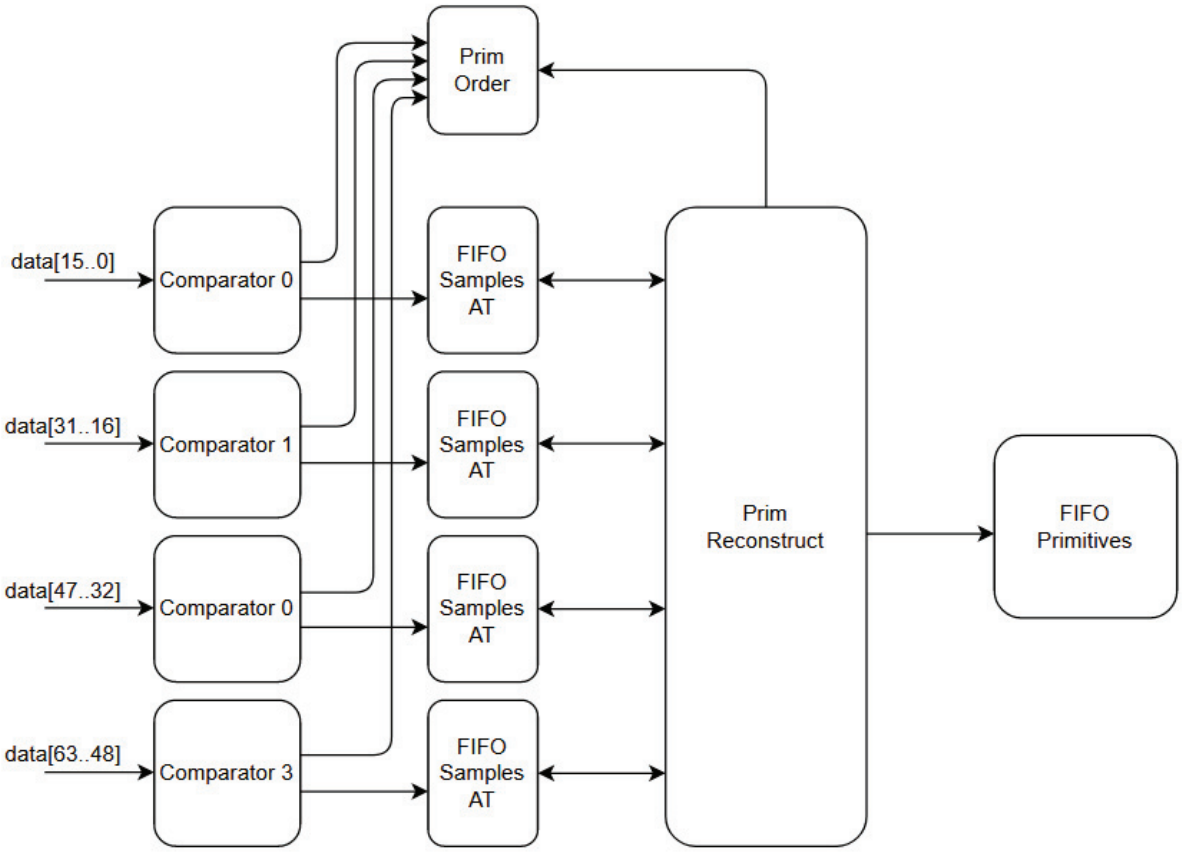


FIGURE 6.36: Implementation of FIFOs

in the Prim Order module. Indeed, storing the order is essential so that the **PRM** can correctly sequence the different reads of the **FIFOs AT** in order to reconstruct the TP in the proper order.

It is important to note that the TP processing algorithm, as presented, is supposed to work on consecutive samples. The fact of using this architecture introduces a parallelization of the processing at the level of 4 samples which therefore prevents, a priori, a sequential approach. Storing values and the order in which they arrived helps reproducing this sequential behavior in the **PRM**. This is because if the **PRM** sees only values above the threshold in the order in which they arrived, it can process them the way foreseen by processing algorithm, as previously described.

A code was therefore defined to maintain the order in which samples above the threshold arrive. Each comparator has a unique identifier on 2bits (`0b00`, `0b01`, `0b10`, and `0b11`). The idea is to store this identifier as well as a flag in the **Prim Order** to know which, among the 4 samples, was detected above the threshold. A 3-bit code is therefore assigned to each comparator (0 & ID if <threshold, 1 & ID if> threshold) and all 4 codes are inserted into the **Prim Order** if at least one of the four samples is above the threshold.

In the example illustrated by the timing diagram in Fig 6.37, two 64-bit words contain samples above the threshold; those located at fronts 3 and 4. In this first word, 2 samples are above the threshold: samples 2 and 3. Some samples are above the threshold, a code will therefore be assigned to each one, which will be:

- 000: 0 -> Sample < Threshold & 00 -> ID Comp 0

- 001: 0 -> Sample < Threshold & 01 -> ID Comp 1
- 110: 1 -> Sample > Threshold & 10 -> ID Comp 2
- 111: 1 -> Sample > Threshold & 11 -> ID Comp 3

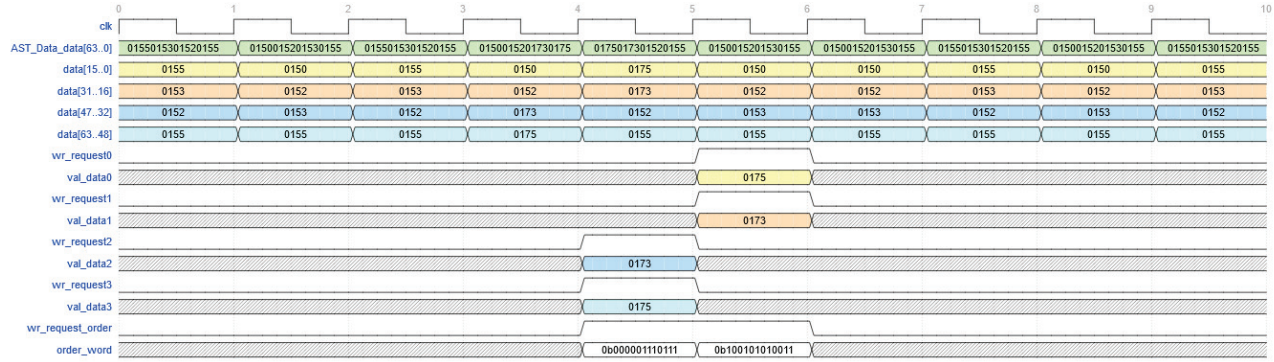


FIGURE 6.37: Example of processing of a packet

These 4 codes of 3 bits each are then concatenated into a single word of 12 bits which is inserted into the FIFO present in the **Prim Order** module, which corresponds to the word `0b000001110111` in this case. Similarly, in the next 64-bit word, 2 samples are also above the threshold. This time, they are samples 0 and 1. This corresponds to the word `0b100101010011` inserted in the **Prim Order**.

Once the samples detected above the threshold are stored as well as the order in which they appeared, the PRM reads the **Prim Order** to know in which order to poll the **FIFOs AT**. It detects when to perform these readings using the empty signal of the **Prim Order**. In addition, this FIFO has a peculiarity which is that the size of its input data is different from that of its output data. The input bus is on 12bits while the output bus is on 3bits. This makes it possible to manage the codes of each comparator one by one and therefore to process the **AT samples** one by one.

Continuing with the previous example, as shown in Fig 6.38, the **PRM** reads the values present in the **Prim Order**. The first 2 codes are `0b000` and `0b001`, which means that there are no samples yet detected by comparators 0 and 1. Then the third code is `0b110`, which means that comparator 2 has detected a value above threshold, the **Prim Reconstruct** module therefore reads the corresponding **FIFO AT**. It then continues with the 4th, 5th and 6th codes which notify to it to read in turn the **FIFOs AT** 3, 0, and 1. Finally, it arrives to the 7th and 8th codes where as for the first two, it has no **FIFOs AT** to read.

This process may seem complicated but it allows to master well the stages of construction of TPs. It is important to note that this reconstruction can take longer than the filling of the **AT and Order FIFOs**, but this is compensated by the fact that the rate of primitives received on the different channels is very low compared to the quantity of data analyzed.

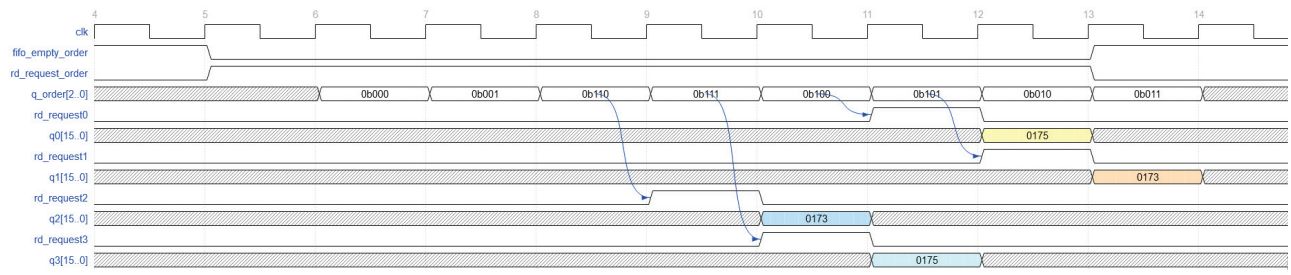


FIGURE 6.38: Example of processing of values over threshold.

The TPs then generated by the **PRM** are stored in a FIFO while awaiting their transfer via the 10G link to the processing/storage server. The format of the TPs differs slightly from that presented previously. Each TP is stored in a 64-bit word as follows (Fig. 6.39):

- 8bits card number
- 8bits channel number
- 16bits start value
- 16bits sum samples AT
- 16bits nb AT samples

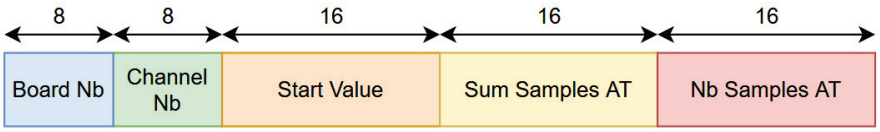


FIGURE 6.39: Trigger primitives data format

A DMA is then connected to this memory and at the end of a complete frame, a request is sent to empty the FIFO.

6.4.1.3 Resource evaluation and conclusions

This time, we chose to work on an FPGA board based on an Arria 10 FPGA. This FPGA is more powerful than the Stratix V used to implement the architecture presented in part 6.40. It has more logic and memory resources than the previous architecture, and more optical links. Indeed, this card contains 4 40Gb/s links (each link containing 4 lines at 10Gb/s) while the Stratix V-based card contains only 2. In the event of concentrating several crates on the same card, the one based on the Arria 10 will be more interesting.

In this part, to analyze the processing on interesting data, a test architecture has been implemented. An architecture processing a crate is implemented as shown in Fig 6.40. The main difference with the architecture that should really be implemented is that the data comes from a memory already present in the FPGA instead of data coming from a real crate. This memory is initialized at startup by the user, which allows complete control over the expected inputs and outputs of the architecture.

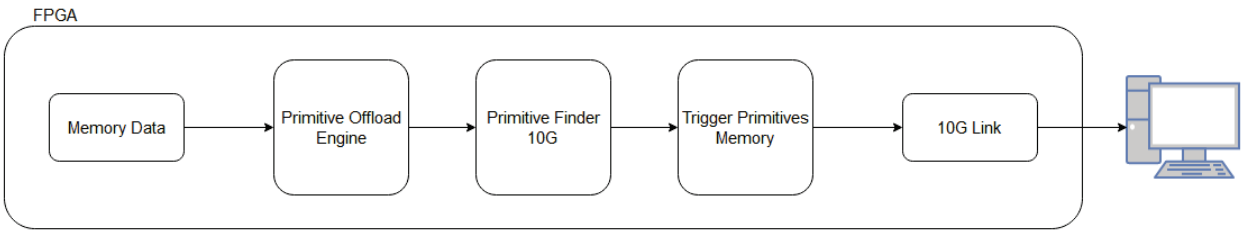


FIGURE 6.40: Test architecture

Table 6.3 shows the resource allocation in the FPGA for the architecture. A very clear improvement is to be noted compared to the resources that the architecture presented in the section 6.3 needed. Indeed, only 1 % of the logical resources and 10 % of the available memory are used. Duplicating this architecture and concentrating several crates on a single FPGA board therefore seems completely feasible.

It should be noted that in the case of the real architecture implementation, another **10G link** would take the place of the **Memory Data**. This means that a little more logic would be used by this link. The memory would remain more or less the same because the buffers of the **10G link** take about the same resources as the **Memory Data**.

Device family	Arria 10
Logic Utilization (in ALMs)	4,826 / 427,200 (1 %)
Total block memory bits	5,444,996 / 55,562,240 (10 %)

TABLE 6.3: FPGA resources used for the decompressed data flow test architecture

In terms of logical resources, the architecture consumes very few. The Fig 6.41 shows the evolution of the logical resources needed to implement several crates. With 5 % of the resources, it is already possible to implement 8 crates, so it would be possible to place a large number of crates (about 160) on a single board if one does not take into account any other limitations for a moment.

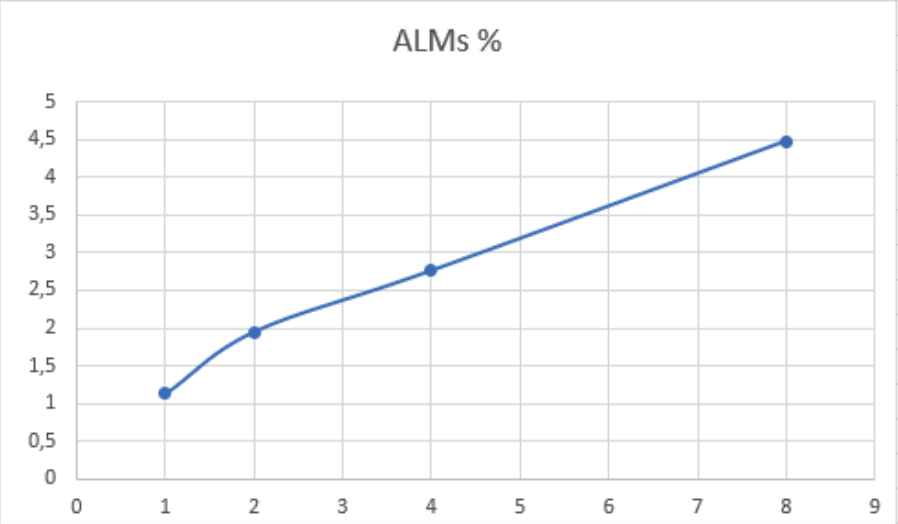


FIGURE 6.41: ALMs resource allocation evolution with regards of the number of treated crates

In terms of memory, the architecture takes up about 10% of the FPGA's internal memory. This architecture uses more memory than the version described in the section 6.3. A part of this memory is used to store the detected TPs contrary to the previous architecture. The amount of memory used by the architecture is therefore directly related to the number of TPs that the architecture can buffer.

The output memory of the architecture has for the moment been over-dimensioned in order to be sure to be able to store all the TPs that could be presented. This memory is 256kB (1 byte noted B = 8bits noted b) and its content is intended to be directly sent to the more global system of treatment of all the TPs. It is organized as shown in Fig 6.42.

In this memory, between each space where the TPs detected by the architecture are stored there is an Ethernet header that contains:

- Destination MAC address on 48bits
- Source MAC address on 48bits
- Protocole type on 16 bits
- Crate number on 16 bits

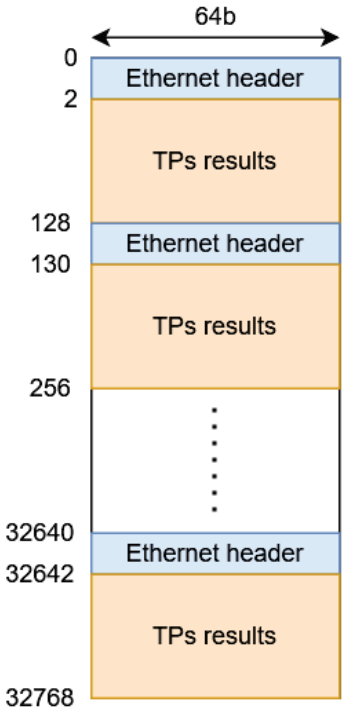


FIGURE 6.42: TP Memory

This makes a 128bits header or 2 words of 64bits in the memory. The Ethernet packets have been chosen with a size of 8192bits or 1024B/1kB or 128 words of 64bits. Reserving 2 words for the header, each packet can thus carry up to 126 TPs. Because of the total size of the memory which is 256kB, the memory, taking into account the headers, can store 256kB * 126 / 1kB = 32256 TPs. It is important to note that this number is very large compared to the number of primitives expected in the detector because it would correspond to an average of 50 primitives per channel per 640 channels.

With this amount of memory used to store the TPs, Fig 6.43 shows the evolution of the allocation of memory resources according to the number of crates concentrated on a single board. The architecture implementing 8 crates needs about 55% of the FPGA internal memory. Limits are therefore reached more quickly than with logical resources.

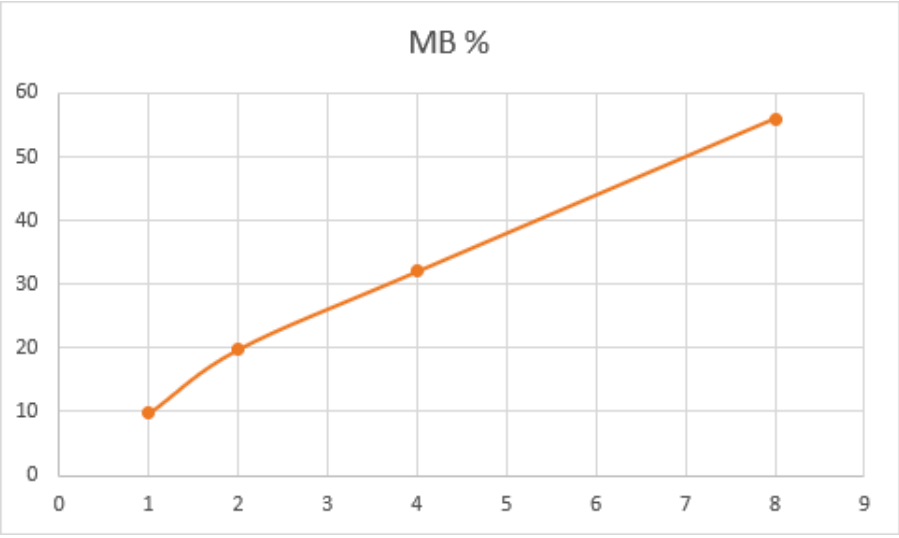


FIGURE 6.43: Memory blocks resource allocation evolution with regards of the number of treated crates

To get an idea of how to size these memories more finely, two parameters must be taken into account:

- The number of TPs that can be buffered locally in the FPGA board
- The frequency at which these TPs are sent to the global processing architecture

The determination of these parameters goes with the constraint of real-time processing. Indeed, the TPs must be stored for a certain time. They cannot be sent one after the other in a dedicated packet to avoid saturation of the 10G output link. However, it is also not possible to wait for a large number of TPs per crate due to the rarity of events. The real time aspect of the processing and analysis would not be respected in this case. We must therefore find a compromise that allows us to send a certain number of primitives in a time that allows us to maintain a real-time approach. The question is therefore to know if the packets containing the TPs fill up fast enough to be sent once they are full.

To answer this question, several elements are necessary:

- The size of a packet containing the TPs : 1024B
- The size of a TP : 8B
- The frequency with which TPs are expected

To evaluate this last element, it is possible to rely on the event causing TPs most often and at all times, the radiativity of the ^{39}Ar. The activity of the ^{39}Ar is 1Bq/kg [48], or 1 decay/second/liter, where each of these decays will generate at least one TP. A strip in the detector (corresponding to a channel) sees a volume of 3.12mm * 3m * 12m = 0.112 m^3 or 112 liters. This means that the ^{39}Ar generates 112 TPs per second per channel or 112 * 640 = 71680 TPs per second per crate.

Each packet contains (1024 - 16) / 8 = 126 TPs per packet (16B of non useful data, Ethernet header). Each crate generates a packet of TPs in 126 / 71680 = 1.8ms. There is thus enough data generated to fill the packets and to keep the real time of the treatment if the packets are sent at the time they are ready.

These packets represent 4.5Mb of TPs per second, so it is possible to process a large number of crates on the same card while using a single 10Gb/s output interface.

If memory is really a problem, it can be interesting to move the storage of the primitives on an external memory to the FPGA. Indeed, FGPA boards often integrate external memory such as DDR3-4 to allow storing larger volumes of data.

The disadvantage of these memories is that they can cause latency. Moreover, this solution would require the development of a particular controller and would have to be the object of a whole evaluation.

The last constraint to the concentration of multiple crates on the same FPGA board that seems to be more limiting for the moment is the number of 10Gb/s optical links available on the FPGAs used. Indeed, in terms of links, the A10P3S board has 16 10Gb/s links (4x40Gb/s with 4 lines at 10Gb/s per link). Counting at least one link for the output interface, this implies a maximum of 15 crates processed on this board

This study was carried out with the DUNE baseline in mind and a 10Gb/s communication at the output of the μTCA crates. However, thanks to the results obtained on compression and decompression, it was shown that an online decompression was too expensive in resources. The idea to get around this problem is to send the data using a 40Gb/s link to avoid compression and decompression. This

40Gb/s link technology is a technology that has been available only recently on the μTCA and is the solution towards which DUNE tends to go. This change would imply several things for the processing of TPs but does not invalidate the developments made.

A first thing to note is that each input link of a μTCA crate will take 4 lines at 10Gb/s. This means that on the A10P3S board, only 3 crates will be able to be processed by the board. A solution could be to switch to a more powerful FPGA board with more available links. At Bittware, for example, the 520R-MX board based on a Stratix 10 has 48 lines at 10Gb/s, which would correspond to 12 40Gb/s links. If one line is used to send the primitives, 11 links will be usable as input to receive the crates data. Moreover, because of the internal architecture of the **PFM10G** and the 40Gb/s link technology, the changes required to make the processing work are not huge. Indeed, the data rate would increase from 64b at 156.25MHz to 256b at 156.25MHz. This means that instead of processing 4 samples in parallel, we would have to process 16. A duplication of the Comparator modules would therefore be necessary within the architecture but the operating principle would remain entirely the same.

Chapter 7

Outlook and conclusions

DUNE being a project with important stakes for modern physics, it entails equally complex challenges on the technical side, in particular on the acquisition system. Indeed, to detect neutrino interactions with a sensitivity never reached before, DUNE plans to build a 40kton detection target. This target, composed of 4 modules each the size of a multi-storey building ($12m \times 12m \times 60m$), aims to have a spatial resolution of a little more than 3mm and a sampling of the signals every 400ns.

This corresponds to a set of more than 150,000 channels which generates a throughput of about 4Tb/s. Moreover, due to the unprecedented accuracy of the detection technology, some external low energy neutrino events could be detected. This is the case of neutrinos that could come from a Supernova in the Milky Way. Since the detector has a particular sensitivity to electron neutrinos as well as the ability to detect neutrinos at energies up to a few MeV, detecting this type of event - if it were to occur - is one of the major scientific objectives of the project.

This event poses several challenges on its detection:

- One of the challenges that this event poses is its rarity, a Supernova in the Milky Way occurs, on average, only about 3 times per century. The last closest supernova that was observed was Sna 1987A where only a few neutrinos were detected. Very useful information could be extracted from the analysis of these data to put better constraints on the supernova models, however it would now be very interesting to have more. In addition, DUNE predicts the number of neutrino interactions that could be observed in the detector to be in the order of a thousand, which would be a very interesting data source.
- The second challenge is related to the duration of these Supernova events. Current models predict a flux of detectable neutrinos lasting about ten seconds, which makes it a much longer event and much less localized in a precise place of the detector than any other type of event. Indeed, where only a part of the detector data is needed and for a duration of a few ms, a Supernova event implies a storage of at least the 40Tbits of data of the 10 seconds window from the moment the event arrives.

For this type of event to be detected, due to the two previous constraints, a system is needed that is capable of several things:

- To maximize the chances of such a rare event occurring, the detector must have a long lifetime. This is the case for DUNE which plans to take data for a few decades.

- To be sure not to miss this event if it were to occur, it is necessary to be able to analyze all the data of a sliding window of 10 seconds.

A continuous acquisition is therefore necessary to meet these constraints. To evaluate the feasibility of the project, two prototypes of the detector have been built at CERN (protoDUNE SP and protoDUNE DP). These prototypes correspond to 1/40th of the volume of a module of the far DUNE detector. The fact that the acquisition is continuous has not yet been implemented on the prototype of the protoDUNE DP detector, which is currently operating with an external trigger. The work presented in this manuscript is part of the evaluation of the resources and the material necessary for the good functioning of the acquisition of DUNE with, as a starting point, the progress already made by protoDUNE DP as well as the material already in place.

One of the major problems with such a detailed acquisition is the very high data rate generated by the detector. In addition, the vast majority of this data will be noise, due to the small number of events to be observed. However, some events being very close to the background and not necessarily very well known, it is not possible to roughly discard the data that does not seem useful. This is, for example, the case of supernova neutrino bursts which will be just above the background but which must be saved. The storage and analysis of these data is therefore a challenge to set up a hardware means able to process and store all the data without losing events that could be interesting.

Two studies were therefore conducted during this thesis seeking to reduce the data rate and to evaluate the hardware required to implement a large-scale real-time acquisition system.

The first study concerns the study of a lossless data compression of Huffman type at the level of the AMCs in order to reduce the data rate coming from the uTCA chassis. The principle of this compression is explained in the first part of chapter 5. It has two major advantages:

- The overall data rate is reduced by a factor of 5, which would save storage resources later on.
- This reduction allows to use exactly the same material of the protoDUNE acquisition for DUNE.

Indeed, to support this second point, a uTCA crate concentrates 640 channels and in view of a real time acquisition where each of these channels on 12bits is sampled at 2.5MHz, the data rate coming from a continuous crate is thus 19.2Gb/s. A problem arose on the hardware side in view of the evolution from protoDUNE to DUNE because the crate have a 10Gb/s data output link, which is about half the rate needed for continuous acquisition.

Compression was therefore implemented at the level of each channel, in the AMCs just after the signals had been digitized. This functional firmware was tested in protoDUNE, where several low-level operations were performed to optimize the compression factor that could be obtained. The key to this optimization was to use the fact that each sample is saved on 12bits while the ADCs work on 14bits. It was therefore possible to play with the internal offset of each of the ADC channels to reduce the RMS noise of the pedestals and at the same time increase the compression ratio. During the tests of empty detector, it was possible with these optimizations to reach compression ratios of 30. However, in real-life tests, the front-end electronics make too much noise for the optimizations to be relevant. The compression ratios reach a maximum of only 5 in this case.

To complete the study on compression, its hardware feasibility and the potential study of real-time processing of the data just after, it was necessary to evaluate to what extent it was possible to carry out a real-time data decompression. Indeed, if certain types of processing on the data stream, such as the search for trigger primitives, filtering, etc., are to be carried out further down the acquisition chain, data decompression is necessary upstream.

An architecture, explained in part 5.3 and developed during this thesis, has been implemented on an FPGA board based on a Arria 10. After optimizations of this architecture, this type of FPGA is able to decompress the flow coming from two uTCA crates in real time by exploiting at most its resources. A far detector module of DUNE that requires the use of more than 200 crates would therefore require the use of more than a hundred FPGA cards of this type if this solution were to be implemented. This solution, although functional, would however be very expensive and complicated to implement. Indeed, the decompression itself being supposed to be the first step of a real time processing chain, it would have to be a low hardware cost operation. Where a first step of concentration of several crates on a single FPGA board would have been desirable. The rest of the work was therefore focused on working on the raw data flow directly out of the uTCA crates to try to reduce the throughput and achieve a larger concentration of the flow of different crates on the same hardware.

The second study concerns a real-time analysis of trigger primitives. Trigger primitives, which are signal peaks distinguishable from noise, allow us to know if any activity has occurred on a channel. This is an intermediate information that is not a complete reconstruction of a particle trace but can be very useful to know if a noticeable event has happened in the detector.

The first step was to develop and experimentally validate a module that detects TPs on a channel level on raw data. This module could then be duplicated 640 times in order to process the entire flow coming from a uTCA crate after a decompression step. This architecture is costly in resources, about the same order as the decompression architecture alone. However, it has an advantage over the decompression architecture which is that the data can be processed without any buffering. This means that if there is no need to perform upstream decompression, there is not necessarily a need to duplicate the processing of the TPs.

So in a second step, without taking into account the compression and decompression that goes with it, a single module capable of processing the whole flow of a crate was realized. This module, being certainly bigger individually than one of the 640 previous modules, is not very expensive in resources and is able to treat a whole data flow coming from a uTCA crate. After functional validation, this module could be duplicated until the limits of the hardware resources were reached. With the most powerful hardware available in the lab - an Arria 10-based FPGA board - it was possible to place 15 of these modules on one board. This means the concentration of 15 uTCA crate outputs on the same board. These results are explained by the number of 10Gbps links available on the board. 16 links can be used, but 1 of these links has to transmit the results of the detected TPs to another architecture.

To summarize, decompression is very resource intensive for real-time processing. It is preferable to work directly on the raw data where some real-time processing is possible and where a realistic architecture in terms of hardware can be implemented. The processing of the TPs illustrates that a first concentration of the data can be achieved. This induces a change in the DUNE DAQ model at the level of the transmission links of the uTCA crate outputs. Indeed, to be able to transmit all the data produced by a crate, a 10Gbps link only works if a compression of at least a factor of 2 is implemented. Therefore, it is necessary to switch the crate output to the next standard which is a 40Gbps transmission. This does not lead to a great change in the results presented because the data transmission is carried out at the same speed internally on the links (256b at 156.25MHz for the 40Gbps and 64b at 156.25 for the 10Gbps). An internal duplication of the number of samples processed in parallel (16 instead of 4) is to be expected but the operation remains the same. However, this would cause an even greater bottleneck in the links, so a more powerful card would be necessary. This kind of boards exist, at the moment it is possible to find FPGA boards on the market with up to 12 40Gbps links, which would already allow a concentration of several crates on the same FPGA board.

Bibliography

[1] J Chadwick. Intensitätsverteilung im magnetischen Spectrum der β-Strahlen von radium B + C. *Verhandl. Dtsc. Phys. Ges.*, 16:383, 1914.

[2] Laurie M. Brown. The idea of the neutrino. *Phys. Today*, 31N9:23–28, 1978.

[3] J. Chadwick. The Existence of a Neutron. *Proc. Roy. Soc. Lond. A*, 136(830):692–708, 1932.

[4] J. Chadwick and M. Goldhaber. A Nuclear Photo-effect: Disintegration of the Diplon by Gamma Rays. *Nature*, 134:237–238, 1934.

[5] H. Bethe and R. Peierls. The 'neutrino'. *Nature*, 133:532, 1934.

[6] Frederick Reines and Clyde L. Cowan. The neutrino. *Nature*, 178:446–449, 1956.

[7] B. Pontecorvo. Inverse beta process. *Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol.*, 1:25–31, 1991.

[8] Raymond Davis, Jr. Attempt to detect the antineutrinos from a nuclear reactor by the Cl37(anti-nu,e-) A37 reaction. *Phys. Rev.*, 97:766–769, 1955.

[9] B. Pontecorvo. Electron and Muon Neutrinos. *Zh. Eksp. Teor. Fiz.*, 37:1751–1757, 1959.

[10] M. Schwartz. Feasibility of using high-energy neutrinos to study the weak interactions. *Phys. Rev. Lett.*, 4:306–307, 1960.

[11] G. Danby, J. M. Gaillard, Konstantin A. Goulianos, L. M. Lederman, Nari B. Mistry, M. Schwartz, and J. Steinberger. Observation of High-Energy Neutrino Reactions and the Existence of Two Kinds of Neutrinos. *Phys. Rev. Lett.*, 9:36–44, 1962.

[12] Martin L. Perl et al. Evidence for Anomalous Lepton Production in e+ - e- Annihilation. *Phys. Rev. Lett.*, 35:1489–1492, 1975.

[13] R. Davis. Solar neutrinos. II: Experimental. *Phys. Rev. Lett.*, 12:303–305, 1964.

[14] Raymond Davis, Don S. Harmer, and Kenneth C. Hoffman. Search for neutrinos from the sun. *Phys. Rev. Lett.*, 20:1205–1209, May 1968.

[15] John N. Bahcall. Solar neutrinos. i. theoretical. *Phys. Rev. Lett.*, 12:300–302, Mar 1964.

[16] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, 1998.

[17] M. H. Ahn et al. Measurement of Neutrino Oscillation by the K2K Experiment. *Phys. Rev. D*, 74:072003, 2006.

[18] P. Adamson et al. Measurement of the Neutrino Mass Splitting and Flavor Mixing by MINOS. *Phys. Rev. Lett.*, 106:181801, 2011.

[19] K. Abe et al. Improved constraints on neutrino mixing from the T2K experiment with $\mathbf{3.13 \times 10^{21}}$ protons on target. *Phys. Rev. D*, 103(11):112008, 2021.

[20] M. A. Acero et al. An Improved Measurement of Neutrino Oscillation Parameters by the NOvA Experiment. 8 2021.

[21] B. Aharmim et al. Electron energy spectra, fluxes, and day-night asymmetries of B-8 solar neutrinos from measurements with NaCl dissolved in the heavy-water detector at the Sudbury Neutrino Observatory. *Phys. Rev. C*, 72:055502, 2005.

[22] S. Abe et al. Precision Measurement of Neutrino Oscillation Parameters with KamLAND. *Phys. Rev. Lett.*, 100:221803, 2008.

[23] Ivan Esteban, M.C. Gonzalez-Garcia, Michele Maltoni, Thomas Schwetz, and Albert Zhou. The fate of hints: updated global analysis of three-flavor neutrino oscillations. *Journal of High Energy Physics*, 2020(9), Sep 2020.

[24] NuFIT 5.1 2021. `http://www.nu-fit.org`.

[25] P.F. De Salas, S. Gariazzo, O. Mena, C.A. Ternes, and M. Tórtola. Neutrino Mass Ordering from Oscillations and Beyond: 2018 Status and Future Prospects. *Front. Astron. Space Sci.*, 5:36, 2018.

[26] S. P. Mikheev and A. Yu. Smirnov. Resonant amplification of neutrino oscillations in matter and solar neutrino spectroscopy. *Nuovo Cim. C*, 9:17–26, 1986.

[27] L. Wolfenstein. Neutrino Oscillations in Matter. *Phys. Rev. D*, 17:2369–2374, 1978.

[28] Babak Abi et al. Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II: DUNE Physics. 2 2020.

[29] B. Müller. Neutrino Emission as Diagnostics of Core-Collapse Supernovae. *Ann. Rev. Nucl. Part. Sci.*, 69:253–278, 2019.

[30] B. Abi et al. Supernova neutrino burst detection with the Deep Underground Neutrino Experiment. *Eur. Phys. J. C*, 81(5):423, 2021.

[31] Irene Tamborra, Bernhard Muller, Lorenz Hudepohl, Hans-Thomas Janka, and Georg Raffelt. High-resolution supernova neutrino spectra represented by a simple fit. *Phys. Rev. D*, 86:125031, 2012.

[32] B. Abi et al. Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I: Introduction to DUNE, 2020.

[33] David R. Nygren. Proposal to investigate the feasibility of a novel concept in particle detection. 1974.

[34] Davide Caiulo. *Charge readout analysis in Liquid Argon Time Projection Chambers for neutrino and astro-particle physics*. Theses, Université de Lyon, April 2017.

[35] Carlo Rubbia. The liquid-argon time projection chamber: a new concept for neutrino detectors. Technical report, CERN, Geneva, 1977.

[36] S. Amerio et al. Design, construction and tests of the ICARUS T600 detector. *Nucl. Instrum. Meth. A*, 527:329–410, 2004.

[37] C Anderson et al. The ArgoNeuT detector in the NuMI low-energy beam line at Fermilab. *Journal of Instrumentation*, 7(10):P10019–P10019, Oct 2012.

[38] R. Acciarri et al. Design and Construction of the MicroBooNE Detector. *JINST*, 12(02):P02017, 2017.

[39] R. Acciarri et al. The Liquid Argon In A Testbeam (LArIAT) experiment. *Journal of Instrumentation*, 15(04):P04026–P04026, Apr 2020.

[40] B. Abi et al. The Single-Phase ProtoDUNE Technical Design Report, 2017.

[41] B. Abi et al. Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume IV: Far Detector Single-phase Technology, 2020.

[42] DUNE Collaboration, B. Abi, et al. The DUNE Far Detector Interim Design Report, Volume 3: Dual-Phase Module, 2018.

[43] David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[44] P. Abratenko et al. The continuous readout stream of the MicroBooNE liquid argon time projection chamber for detection of supernova burst neutrinos. *JINST*, 16(02):P02008, 2021.

[45] BittWare FPGA boards (legacy products). `https://www.bittware.com/fpga/legacy-obsolete/`.

[46] FPGA architecture - white paper. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01003.pdf`.

[47] Introduction to Intel FPGA IP cores. `https://www.intel.com/content/www/us/en/docs/programmable/683102/21-3/introduction-to-cores.html`.

[48] P. Benetti et al. Measurement of the specific activity of ar-39 in natural argon. *Nucl. Instrum. Meth. A*, 574:83–88, 2007.