CERN-THESIS-2009-281

ALMA MATER STUDIORUM
UNIVERSITY OF BOLOGNA

FACULTY OF SCIENCE MM.FF.NN.

DOCTOR OF PHILOSOPHY DEGREE IN
Physics (FIS/01)

Applications of High Speed Configurable Logic Devices in Modern Particle Physics Experiments

Presented by

Dr. FILIPPO MARIA GIORGI

PhD Coordinator

Prof. FABIO ORTOLANI

Principal Adviser

Prof. ENZO GANDOLFI

Co-Adviser

Dr. GABRIELLI ALESSANDRO

Cycle XXI

Final exam, year 2009

To all those who made this possible.

Contents

Introduction

During my PhD activity I worked on electronics development and realization for different high-energy particle physics experiments. Formerly I took part in a collaboration involved in the realization of a mixed-signal front-end board for the NEMO telescope, an ambitious project for a huge underwater neutrino detector. Additionally, I have been involved in a technology research for the project, realization and testing of a Monolithic Active Pixel Sensor (MAPS) for vertex detectors.

These two activities brought both good results, and for this reason this thesis concerns generally the application of high-speed configurable electronics on different fields of physics such as neutrino astronomy and high-energy collisions.

Reliability and performance requirements about electronics for future particle detectors are scaling fast with the complexity of the physic we investigate. For this reason fast configurable-logic devices like Field Programmable Gate Array (FPGA) are taking a more and more growing role in this field of application. Moreover their extreme flexibility allows to reach the production phase faster, deferring the refinement of data acquisition policy to a later point, for example during the commissioning; sometimes the flexibility brought by these devices is so advantageous that it allows to use the same hardware architecture for different experiments.

Nowadays FPGA technology is improving fast and in the last ten years it has been adopted in many Data Acquisition (DAQ) systems of big detectors like those starting to operate at the Large Hadron Collider (LHC) of CERN, at the Collider Detector of Fermilab (CDF), and in many other experiments in the world.

Big steps have been done as well in the direction of lowering the power consumption, giving the opportunity to use FPGA devices very close to the detector's front-ends, especially in those situation where material budget and radiation hardness is not a critical issue. The application on a Km^3 scale telescope like NEMO is the proof: the front-end is deployed 3.5 Km beneath the sea level and it consists of four

thousand optical modules, each containing a Photo-Multiplier Tube (PMT) and its relative readout electronics. Under these conditions it is helpful to have a smart and configurable logic as close as possible to the front-end module without the unaffordable price of a high total power requirement.

On the other hand, configurable devices, used in high-energy collision experiments, usually find their field of application in the DAQ readout infrastructure. In this scenario the front-end is made up of full custom electronics in order to decrease the overall material budget, to lower the power consumption and to bear high radiation doses. The FPGA technology hence is used in the DAQ readout infrastructure, like that realized by the SLIM5 collaboration, to perform event building, first-order data analysis and to produce complex L0 triggers. In this situation is valuable the capability to improve the readout policy without a reworking of the DAQ boards.

During my PhD activities I worked with these devices in both the scenarios described above. The first application concerned the front-end board for the optical modules of the NEMO experiment, where a photo-multiplier signal is acquired, digitized and sent to shore on a dedicated communication protocol. In the second place I joined the SLIM5 collaboration where a Monolithic Active Pixel Sensor was submitted to a test-beam, where a dedicated DAQ infrastructure has been realized, tested and operated at CERN, Geneva.

outline

Due to the previous considerations this thesis is then subdivided into two main parts.

The first part starts with a digression on neutrino astronomy to introduce this new branch of science and to point out the target physic of the project NEMO km^3 telescope. In the second chapter the architecture of this detector is showed, describing its mechanical structure and then, the readout electronics. The following chapter focuses on the front-end electronics housed in the optical modules of the NEMO Phase-1 telescope, a small prototype already deployed and operated near in the Gulf of Catania; then an alternative mixed-signal solution, based on the analog acquisition chip LIRA06, is described as a possible front-end candidate for the whole km^3 project. The firmware digital logic is also briefly described. In the last chapter of this part are exposed some test results.

The second part of the thesis describes the work of the SLIM5 col-

laboration to project, realize and test the chip APSEL4D, a silicon pixel sensor. The first chapter introduces the common problem of vertex reconstruction in collider-physics, then it shows typical silicon technologies adopted in such experiments, trying to point out the limits of the current pixel detectors and proposing a heading towards better performances. The second chapter concerns the APSEL chip evolution from the first submissions to the last chip used in the test beam, provided with a fully integrated digital readout electronics. There follows a chapter about the test beam setup and the DAQ system, with a detailed description of the hardware architecture and of the firmware logic. The sequent chapter shows the results of data analysis performed on test-beam data, taken in September 2008.

Part I

An alternative mixed signal front-end solution for the optical modules of the Nemo Phase-1 experiment

Chapter 1

High-energy neutrino astronomy

Over the last 20 years, there has been a great expansion of the researches into neutrino physics. In this chapter we want to present a summary of neutrino discovery and history in order to explain the great importance that the scientific community is giving to it. The evidence of neutrino oscillations, for example, is one of the challenging aspects that lead beyond the standard model of particle physics.

Another aspect that is gaining more and more interest is the neutrino astronomy. Since the first observation of a neutrino flux from a supernova in the late 80's, it has been hypothesized the exploitation of this particle for astronomical research.

To summarize, a brief discussion on neutrino physics is shown and some astronomical Ultra High Energy (UHE) production models are presented. In the end we introduce the Cherenkov-based detection technique to examine the potentiality of a km^3 scale telescope for astronomical UHE neutrinos.

1.1 History of Neutrino

The existence of neutrino was first postulated around 1930 by Wolfgang Pauli.

The radioactive beta decay seemed to violate the known laws of linear and angular momentum conservation. Many hypotheses were made, between these Pauli theorized the existence of an electrically neutral particle, also involved into the beta decay. This particle was subsequently named *neutrino* in 1933 by Enrico Fermi, who proposed a theory on weak decay. It was the first time that an interaction with

no classical counterpart was proposed. This theory postulates a 0-range force for β decay and incorporates several new concepts: Pauli's neutrino hypothesis, Dirac's ideas about the creation of particle and Heisenberg's idea that the neutron and the proton were related each other.

One year later Bethe and Peiers calculate the cross section for the processes $\nu + n \rightarrow e^- + p$ and $\overline{\nu} + p \rightarrow e^+ + n$. Their article concludes: "...this meant that one obviously would never be able to see a neutrino". They were led to this consideration by the astonishing result of their calculations, in fact $\sigma \simeq 2.3 \times 10^{-44} cm^2 (\frac{p_e E_e}{m_e^2})$, which means for a 2.5 MeV neutrino an absorption length in water of $2.5 \times 10^{20} cm$, more or less the thickness of the disk of our galaxy (equivalent to one light-year of lead).

Nevertheless the advent of very intense sources of neutrino like fission bombs and fission reactors changed that prospect, and in 1956 neutrino was actually detected for the first time by Cowen, Reines et al. As Fermi's theory predicted also an inversion of the β decay, it is possible that an antineutrino will interact with a nucleus through the weak force and will induce the transformation of a proton into a neutron, leaving the nucleus with one less unit of positive charge:

$$\overline{\nu} + N(n, p) \rightarrow e^+ + N(n + 1, p - 1)$$

If the nucleus happens to be that of hydrogen then the interaction produces a neutron and a positron. This is the reaction chosen by Reines and Cowan to detect the neutrino. They realized a very large detector containing an organic scintillator liquid with a high proportion of hydrogen and a small fraction of cadmium, and they used the fission reactor at the Savannah River Plant as a source of $\overline{\nu}$. In the scintillating liquid the signature of an antineutrino interaction was the emission of two consecutive flashes of light. The first flash observed was caused by the two gamma photons outgoing at 180° produced by the annihilation of the generated positron. In the meantime the neutron wanders about following a random path and it is captured by a cadmium nucleus. The resulting nucleus de-excite itself releasing about 9 MeV of energy in gamma ray photons, causing the secondary flash in the liquid. See Fig. 1.1

The discovery of this new elusive particle was published in the article *Detection of the Free Neutrino a Confirmation* and the authors were rewarded in 1995 with the Nobel Prize.

In 1962, it was found by Leon M. Lederman et al. at the Brookhaven

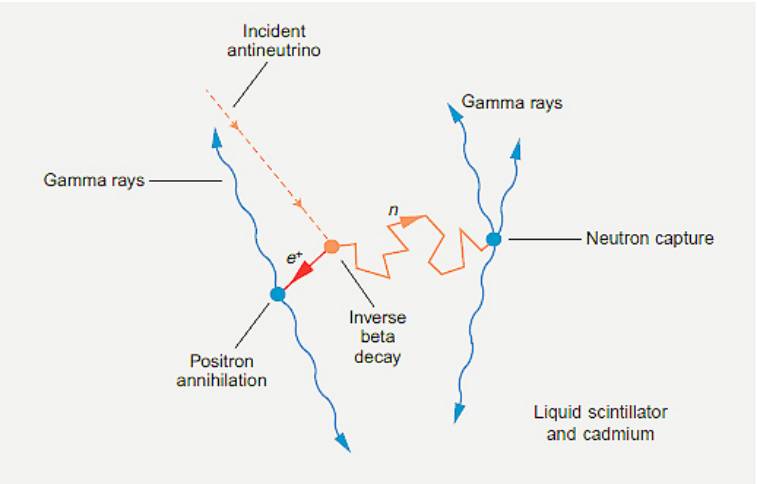

Figure 1.1: **Double signature of inverse beta decay**. Positron and neutron are created, both generating a flash of light one after the other.

laboratory that there were at least two types of neutrino. The first was the *electron neutrino* postulated by Pauli, the second was the partner of the other known lepton μ, hence it was named *muon neutrino*. When in 1975 was discovered the third lepton (τ) at the Stanford Linear Accelerator, it too was expected to have an associated neutrino, but only in 2000 the latest particle of the standard model was observed at Fermilab: the *tau neutrino*.

Since their first detection, neutrinos were detected from nuclear reactors, particle accelerators, from the Sun and earth atmosphere. Finally, in 1987 the first and only direct observation of neutrinos coming from a supernova started the era of *neutrino astronomy*. This discovery was announced by the Kamiokande and the IMB (Irvine Michigan Brookhaven) experiments.

The so called Standard Model of Particle Physics assumes massless neutrinos that can't change flavor; however nonzero neutrino mass and accompanying flavor oscillations remained a possibility. In the late 60's several experiments found that the number of electron neutrinos arriving from the Sun was between one third and one half the number predicted by the Standard Solar Model, a discrepancy that became known as the *solar neutrino problem* and remained unresolved for about thirty years. Starting in 1998, experiments like Super-Kamiokande and Sudbury Neutrino Observatory, began to show that solar and atmospheric neutrinos change flavor, resolving the solar neutrino problem. The elec-

tron neutrinos emitted by the Sun had partly changed into other flavors which the experiments could not detect.

1.1.1 Neutrino interaction properties

In the well known Standard Model of Particle Physics, which describes the elementary particles and their interaction, there are 3 flavor of neutrinos, the neutral partners of the massive leptons.

The neutrino is a fermion and has half integer spin ($\frac{1}{2}\hbar$), is neutral and can interacts only through the weak force. The weak interaction with matter can be classified in two types. There are the neutral current interaction, which involves the exchange of a Z^0 boson, or the charged current interaction, which involves the exchange of a W^+ or W^- boson.

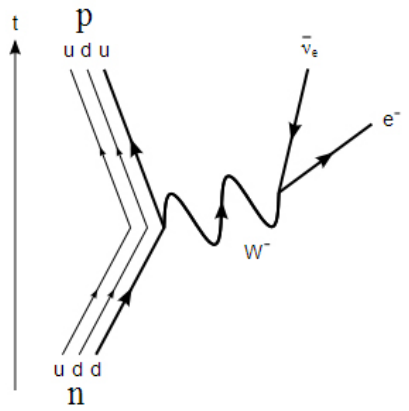

Figure 1.2: **Beta decay**. A neutron transform into proton with the emission of an antineutrino and an electron. The weak charged interaction is mediated by the W^- boson.

The number of existing neutrino flavors can be evaluated observing the decay of the Z^0 boson. This particle can decay into any light neutrino and anti-neutrino types, with *light* meaning of less than half the Z mass. The measurement of the Z^0 lifetime at LEP accelerator of CERN, has shown that the number of these light neutrino types is three, giving a good correspondence to the three flavors of quarks. Some hypothesis exists about the existence of a sterile neutrino, non-interacting via the Z^0 boson but which could appear in a neutrino oscillation.

The β decay is an example of a weak interaction involving the production of an electron neutrino, see Fig. 1.2.

1.2 Neutrino Astronomy

To introduce this branch of astronomy, before reporting a list of the known sources of neutrinos, a brief description of the cosmic radiation is given. In the end some extra-galactic proposed source of UHE neutrinos will be exposed.

1.2.1 Cosmic rays

Almost a century ago, Victor Hess performed experiments with electrometers suspended in balloons. His studies of ionizing radiation at different altitudes led to the conclusions that these "cosmic rays" must have an extra-terrestrial origin. Since then, the phenomenon has been studied by a broad range of different instruments. This list of instruments includes satellite detectors and very large air shower arrays. Although the phenomenon has been studied for almost one hundred years, the origin of some of these particles remains unclear.

The cosmic radiation incident at the top of the terrestrial atmosphere includes all stable charged particles and nuclei with lifetimes of order 10^6 y or longer. *Primary* cosmic rays are those particles accelerated by astrophysical sources and *secondaries* are those particles produced in interaction of the primaries with interstellar gas. Thus electrons, protons, helium, carbon, oxygen, iron and others elements synthesized in stars are primaries. Nuclei such as lithium, beryllium and boron which are not abundant end-products of stellar nucleosynthesis, are secondary, generated by primary interaction with interstellar matter.

The intensity of primary nucleons in the energy range from several GeV to somewhat beyond 100 TeV is given approximately by

$$I_N(E) \approx 1.8E^{-\alpha}\frac{\text{nucleons}}{\text{cm}^2 \text{ s sr GeV}}$$

were E is the energy-per-nucleon (including rest mass energy) and $\alpha = 2.7$ is the differential spectral index of the cosmic ray flux. About 90% of the primary cosmic rays are protons, 9% are helium nuclei and about 1% are electrons. The fraction of of the primary nuclei are nearly constant over this energy range.

Up to energies of at least 10^{15} eV, the composition and energy spectra of nuclei are typically interpreted in the context of *diffusion* or *leaky box* models, in which the sources of the primary cosmic radiation are located within the galaxy [30].

When a cosmic ray hit the atmosphere it can generate a so called *air shower* of secondary particles if it has enough energy. The shower has a hadronic core, which acts as a collimated source of electromagnetic sub-showers generated mostly from $\pi^0 \rightarrow \gamma\gamma$. The resulting electrons and positrons are the most abundant particles in the shower. The number of muons, produced by decays of charged mesons is an order of magnitude lower. Air showers spread over a large area on the ground, and array of detectors operated for long time are useful for studying cosmic rays with primary energy $E_0 > 100$ TeV.

In Fig.1.3 [34] is shown the spectrum of primary cosmic rays, the differential energy spectrum has been multiplied by $E^{2.5}$ in order to display the features of the steep spectrum that are otherwise difficult to discern.

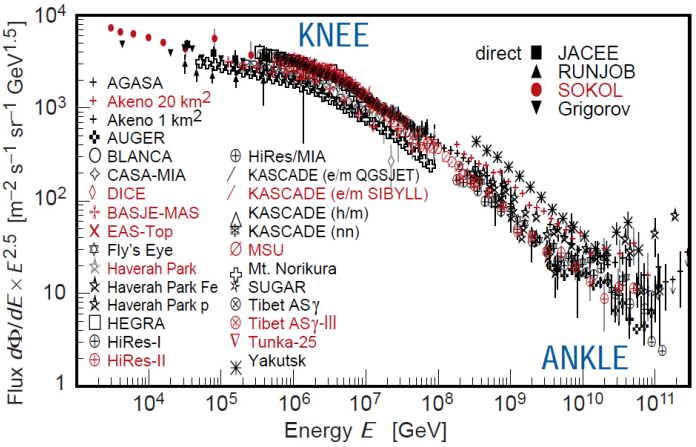

Figure 1.3: **All particle spectrum**. Here is visible the steepening at $10^{15} - 10^{16}$ eV known as the *knee* and the *ankle* at energies around 10^{19} eV.

If the cosmic ray spectrum below 10^{18} eV is of galactic origin, the knee could reflect the fact that some (but not all) cosmic accelerators have reached their maximum acceleration energy potential. Some types of expanding supernova remnants, for example, are estimated not to be able to accelerate particles above energies in the range of 10^{15} eV total energy per particle.

In 1966 Kenneth Greisen, Vadim Kuzmin and Georgiy Zatsepin independently calculated an upper limit in the primary energies based on interactions between the cosmic ray and the photons of the cosmic microwave background radiation. They predicted that cosmic rays with

energies over the threshold energy of 6×10^{19} eV would interact with cosmic microwave background photons to produce pions. This would continue until their energy fell below the pion production threshold.

$$\gamma + p \rightarrow \Delta^{+} \rightarrow p + \pi^{0}$$

or

$$\gamma + p \rightarrow \Delta^{+} \rightarrow n + \pi^{+}$$

Because of the mean path associated with the interaction, extragalactic cosmic rays with distances more than 50 Mpc (163 Mly) from the Earth with energies greater than this threshold energy should never be observed on Earth, and there are no known sources within this distance that could produce them. This effect is called GZK cutoff from the name of its discoverers.

While several experiments have reported events that have been assigned energies above 10^{20} eV [36] [46], more recent experiments such as HiRes [29] have failed to confirm this; results are consistent with the expected cutoff.

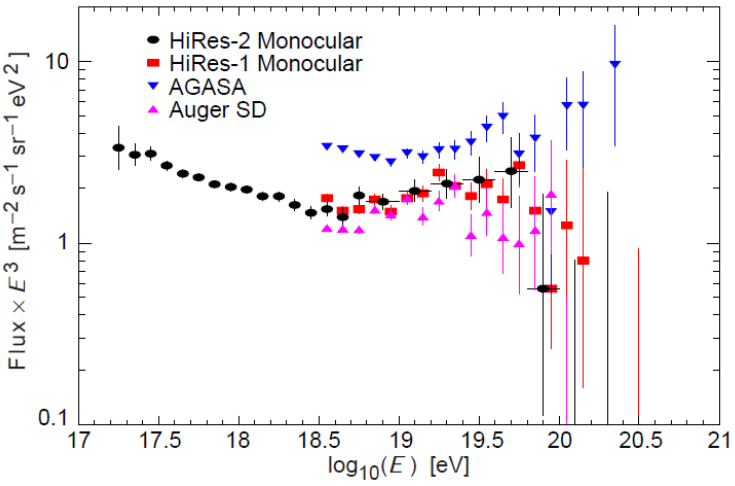

Figure 1.4: **High energy spectrum**. In this graph are shown the results of different experiments. In the higher part of the spectrum we can see the discrepancy between the Auger and the AGASA results.

Fig. 1.4 gives an expanded view of the high energy end of the spectrum, showing only the more recent experiments. This figure and the previous one have shown the differential flux multiplied by a power

of the energy, a procedure that enables one to see structure in the spectrum more clearly but amplifies small systematic differences in energy assignments into sizable normalization differences. All existing experiments are actually consistent in normalization if one takes quoted systematic errors in the energy scales into account. However, the continued power law type of flux beyond the GZK cutoff claimed by the AGASA experiment is contradicted by the HiRes data. The high-statistic amount of data from the more recent Pierre Auger experiment [42], basically confirmed the HiRes results and definitely contradicted the rising trend in energy of the AGASA experiment.

1.2.2 Neutrino sources

Now the main sources of incoming neutrinos are discussed.

Atmospheric neutrinos

As discussed above, Earth is constantly bombarded by cosmic rays, mainly protons but also some neutrons and nuclei. These particles interact with atmospheric nuclei at heights of approximately 12-20 km, creating pions, gamma ray, muons and muon neutrinos, see Fig. 1.5. Some of the muons are ultra relativistic and can reach directly the Earth surface, other less energetic muons instead decay in flight producing both electron and muon type neutrinos. These neutrinos, together with those created in the first hadronic interaction, are termed *atmospheric neutrinos*.

Solar neutrinos

The Sun is a natural nuclear fusion reactor, powered by a proton-proton chain reaction which converts four hydrogen nuclei (protons) into helium. This nucleo-synthesis take place is several steps, using another couple of protons to catalyze the reaction, during which two electron neutrino are released.

The field of solar neutrino research had its birth in the BNL Chemistry Department, where Raymond Davis and colleagues developed a radiochemical method to separate and detect the few radioactive atoms formed by capture of solar neutrinos in a huge target. This first solar neutrino experiment, in the Homestake Mine in South Dakota, used the isotope, ^{37}Cl, as the target in 680 tons of an organic liquid, perchloroethylene. Neutrino capture on the ^{37}Cl, with an energy threshold of 0.814 MeV, produces radioactive ^{37}Ar, a gas, which is removed from

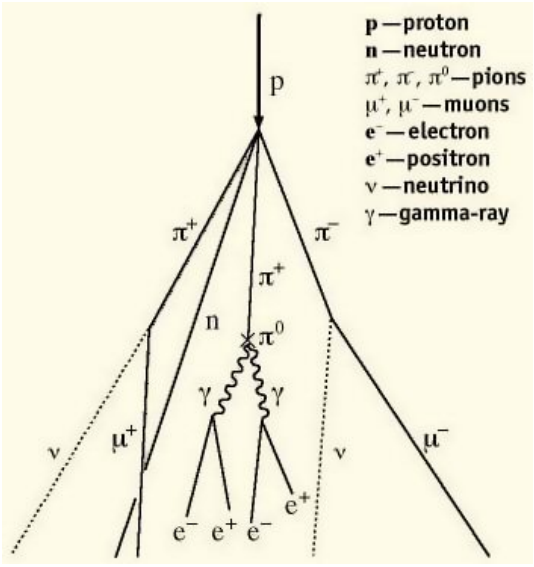

Figure 1.5: **Induced particle shower from a primary proton**. The proton interacts with the atmosphere generating an hadronic shower which decay into gamma, muons, electrons and relative neutrinos.

the target, purified, and counted. The results of this experiment revealed the "solar neutrino problem" mentioned before: The number of measured solar neutrinos was only about one-third of the value predicted from solar theory.

The BNL Solar Neutrino Group participated in GALLEX at the underground Gran Sasso National Laboratory in Italy, where 30 tons of gallium in the form of a 100-ton aqueous solution of gallium trichloride served as the target; SAGE at the Baksan Neutrino Observatory in Russia instead used 57 tons of liquid gallium metal. The results from both gallium experiments confirmed the deficit of solar neutrinos, by observing only 30% of the expected neutrino flux. The GALLEX experiment ended in 1998.

From these experiments, and the Kamiokande and Super-Kamiokande neutrino detectors in Japan, the consensus has developed in the scientific community that the reason for the observed deficit of solar neutrinos is that the neutrinos "oscillate". In other words, the electron-flavor neutrinos that are produced in beta-decay processes in nuclear reactions in the solar interior can be transformed into the other two known neutrino flavors, those of the muon-neutrino and the tau-neutrino. These neutrinos are not produced in the sun's nuclear reac-

tions. In this scenario, the measured solar neutrino flux is artificially low since these other neutrino flavors are not readily observed by most neutrino detectors, and certainly not at all by the radiochemical neutrino detectors. Note that for this process to occur requires that at least one of the neutrino types must have non-zero rest mass. Since the current Standard Electroweak Model carries the assumption of massless neutrinos, the existence of neutrino mass is a major new discovery, leading to changes in the theory, what has been dubbed *New Physics*.

Galactic neutrinos

Cosmic rays also propagate through the interstellar medium of our galaxy, thereby producing secondary particles such as neutrinos in hadronic interactions similar to those reactions giving rise to the terrestrial atmospheric neutrino flux. Unlike Earth's atmosphere, the interstellar medium has a much lower density of about 1 particle per cm^3 which leads to far greater interactions lengths as compared to the decay lengths of the secondary particles. Therefore, in contrast to the atmospheric scenario, mesons and muons are more likely to decay on the way rather than to interact.

As a results, the flux of galactic neutrinos can exceed the flux of atmospheric neutrinos at very high energies, at which the latter is typically suppressed by the energy loss of mesons in high-energy collisions. However, at the low energy end of the spectrum the atmospheric neutrino flux clearly dominates the galactic flux due to the increased rate of reaction by the cosmic rays with the dense atmosphere.

On the assumption that the cosmic ray flux on Earth is uniformly distributed and isotropically distributed through the galaxy, the neutrino flux from the galactic disk has also been calculated on the basis of density, with the interstellar medium in a column length of 1 kpc and a density of 1 nucleon/cm^3.

Extra-galactic neutrinos

Very high-energy cosmic rays propagating in the extragalactic medium interact with the cosmic microwave background and infra-red, optical and ultra-violet background photons. These interactions produce features in the ultra-high-energy cosmic ray spectrum such as the GZK cutoff and their decay products generate the extra-galactic neutrino flux, also referred to as GZK neutrinos.

The total cosmogenic flux was calculated for instance by [12] and it

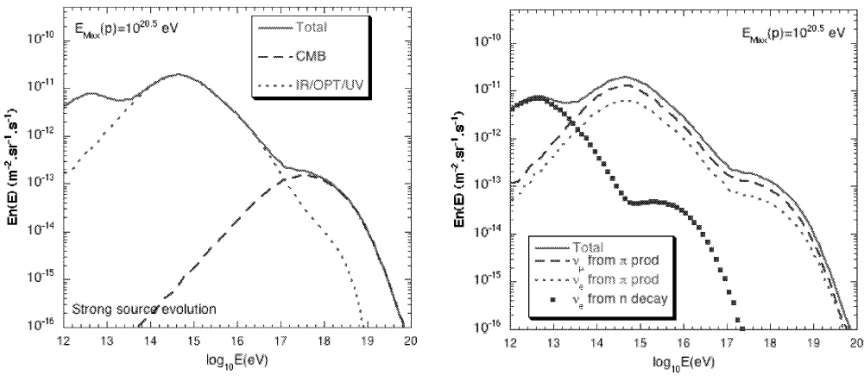

Figure 1.6: **Total cosmogenic neutrino flux from pure proton sources**. Left: separate contributions from the interactions with Cosmic Background Radiation and light (extended to IR and UV). Right: separate contributions of different neutrino flavors (without oscillations).

is displayed in Fig 1.6. Three distinct peaks appear in the total comsogenic neutrino flux. In the TeV energy region the flux is generated purely by neutron decay, which gives rise to electron neutrinos only. At higher energies, in the PeV region, the contribution due to hadronic interaction with photons dominates, while at the highest energies prior to the GZK-cutoff the neutrinos are produced via interaction of photons and nuclei with the CMB radiation. In these models, pions are generated via photo-production and neutrinos emerge from the decay of π^+.

However, despite comprehensive knowledge of the particle physics behind the GZK-cutoff, the phenomenon itself has not been conclusively measured. It is still unclear whether the cosmic ray spectra are truncated above 10^{20} eV, mainly due to lack of data, which is understandable at the extremely low fluxes involved.

1.3 Neutrino telescopes

New frontiers in astronomical research were opened with the discover of neutrinos as they are good candidate for the observation of the cosmological sources mentioned before.

A number of possible techniques exists for detecting high energy neutrinos from space. The most widely exploited method for the core energy range of interest (10^{11} to 10^{16} eV) is the detection of neutri-

nos in large volumes of water or ice, using the Cherenkov light from the muons and hadrons produced by neutrino interactions with matter around the detector. So far, water Cherenkov detectors like IMB and Kamiokande/SuperKamiokande are the only detectors that have observed neutrinos produced beyond the solar system, detecting neutrinos from supernova 1987a.

The first to propose water as a cheap and useful target was Markov in the early sixties [2]. Given the need of a kilometer-scale detector, only designs incorporating large naturally occurring volumes of water or ice can be viable. A deep seawater telescope has significant advantages over ice and lake-water experiments due to the better optical properties of the medium.

The pioneering project, DUMAND, attempted to deploy a detector off the coast of Hawaii in the years between 1980 and 1995. At that time technology was not sufficiently advanced to overcome these challenges and the project was canceled. In contrast, AMANDA [15] and BAIKAL [25] where the equipment is deployed from the surface of the solid glacial ice or the frozen surface lake ice have developed workable detector systems. After the completion of their detector in 2002, the AMANDA collaboration proceeded with the construction of a much larger detector, IceCube. Completion of this detector is expected in 2010.

The work of DUMAND to build a deep sea neutrino telescope is being continued in the Mediterranean Sea by ANTARES [3], NEMO [16] and NESTOR [21]. A more detailed description of these will be given in next chapter.

The fundamental effect, on which the detection is based, is the emission of Cherenkov light by the muon created after a UHE neutrino collision with the target. In the special case of in-water Cherenkov detectors, the directional correlation of the muon and the parent neutrino trajectories is within 0.3° for $E_\nu > 10$ TeV. Neutrino events can be easily recognized as they are the only possible source of upgoing muons, their absorption length is in fact too small to let them travel along all the Earth's core.

1.3.1 The Cherenkov radiation

A charged particle, traveling through a medium at a velocity exceeding the speed of light in that medium, emits Cherenkov radiation. This electro-magnetic radiation is emitted at a characteristic angle θ_c with respect to the direction of the charged particle, thus forming a conical

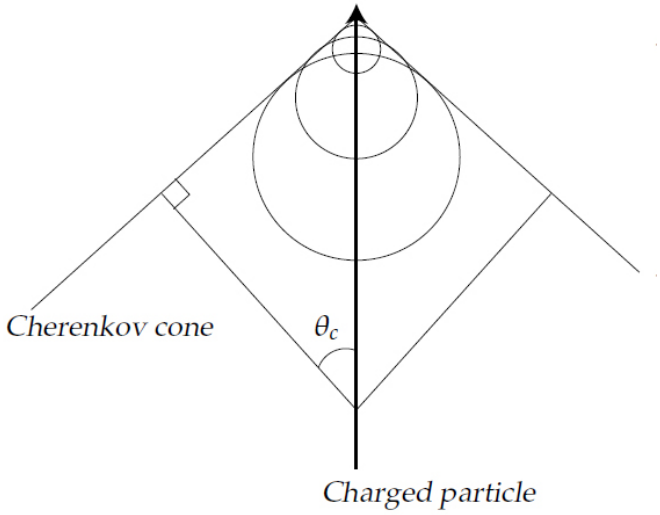

Figure 1.7: **Cherenkov light cone**.

light-front see Fig. 1.7.

The angle θ_c can be expressed as

$$cos(\theta_c) = 1/\beta n \tag{1.1}$$

where β is the ratio v/c of the velocity of the particle to the speed of light and n is the index of refraction for the medium. The water refraction index is about 1.35. Thus, for relativistic particles ($\beta \approx 1$) the value of θ_c is about 42.2°. The number of Cherenkov photons emitted by a particle with unit charge (e.g. a muon) per unit wavelength ($\delta\lambda$) and per unit track length (δx) is given by

$$\frac{dN}{dxd\lambda} = 2\pi\alpha\frac{1}{\lambda^2}\left(1 - \frac{1}{\beta^2 n^2}\right) \tag{1.2}$$

where λ is the wavelength of the emitted photon and α the fine-structure constant. Considering the typical wavelength range of efficiency of a PMT (300 - 600 nm), the number of detectable photons emitted per meter is about 35000.

Photons traveling through the water are subject to several processes. They can be absorbed and scattered by molecules and particles in the water. The effects of photon absorption and scattering can be quantified by the absorption length (λ_{abs}) and the scattering length

(λ_{scat}) which are both wavelength dependent. The intensity of the light emitted by a muon (I_0) is related to the intensity (I) at distance r from the muon track by

$$I \propto I_0 \frac{1}{r} \exp\left(\frac{-r}{\lambda_{abs}}\right) \tag{1.3}$$

The factor $1/r$ comes from the geometrical spread of the Cherenkov cone. The scattering length λ_{scat} is the length at which on average a fraction of e^{-1} of the photons is unscattered.

Chapter 2

The Project NEMO Km^3 telescope

In this chapter is shown the proposed structure of a deep underwater Km^3 scale Cherenkov detector by the Nemo collaboration. As was said in the previous chapter this huge dimension is imposed by a lower limit in the acceptance of the detector of at least some events per year in the energy range above 10^{20}. The main goals of the collaboration are to project an innovative detector with an acceptance as high as possible mediating with deployment feasibility and low overall cost. Several key features as the shape of the detector, the distance and the orientation of the phototubes have been studied by many simulations to find out the best acceptance configuration, and are still under investigation.

The realization of a Km^3 telescope in the Mediterranean Sea will be the Boreal counterpart of the American Project AMANDA-ICECUBE [15], another neutrino telescope which is under construction and which will be deployed under the Antarctic ice. With these two operating stations will be possible to investigate neutrino point sources from all the celestial map.

The realization of such a huge detector in the Mediterranean Sea is possible only with the economical contribution of several countries and for this reason an European collaboration was started up. This collaboration is called Km3NET and its goal is to present to the European Union a technical proposal of the telescope to be financed. This collaboration includes several countries and institutes of research, like NIKHEF[1], CEA-SACLAY[2] and the INFN[3], gathering all the know-how of the pre-existing projects like ANTARES, NEMO and NESTOR.

[1]National Institute for Subatomic Physics of Amsterdam
[2]Atomic Energy Commission of Saclay
[3]Italian National Institute of Nuclear Physics

At the moment the main tasks of this collaboration are to point out which is the physical problem the telescope is meant to investigate, to give a description of the technique for the neutrino detection and to write a technical design report of the proposed telescope. In this document will be described in details the key elements of the facility:

- For instance the so called *Detection Unit*, a vertical structure to support the optical sensors like the NEMO tower or the ANTARES string. The layout of this part is critical both for deployment optimization and for the global acceptance of the telescope.
- The *Optical Module* element which is the Cherenkov light sensitive part. It should be designed to bear high pressures, to grant the higher sensitive surface and, if possible, to be able to give a directional information of the detected photons. The spatial and temporal resolution that the collaboration intends to achieve is few *cm* and less than a *ns*.
- The global *Read-Out System*, from the front-end DAQ electronics to the back-end infrastructure. Another very challenging point, which should mediate between the highest possible bandwidth, the much information as possible and the overall cost.

Like the other collaborations the NEMO project is investigating on the elements mentioned above and it has realized, deployed and operated during the NEMO Phase-1 a demonstrative detection unit. The NEMO Phase-1 experiment consisted of only 16 Optical Modules deployed in the Gulf of Catania but it has been very useful to validate some characteristics and to point out the limits of some others. For details see section § 2.2.2. At the moment the Italian collaboration NEMO has been re-financed for the realization of the Phase-2 experiment, another step forward in the characterization of the key elements described above for an underwater neutrino detector.

A brief description of the pilot projects for the Km^3 telescope is then formerly given. Hence the architectures of the two main phases of the NEMO project are described, and in the end the whole detector proposed by the NEMO collaboration is presented.

2.1 The other European pilot projects

2.1.1 Antares

Antares is an international collaboration which realized a telescope at a depth of 2.4 km, approximately 40 km off the south coast of Toulon -

France. The experiment consists of lines of PMTs which are anchored to the seabed. Buoys on top of these lines keep them approximately vertical. The PMTs are grouped in threes around a string and look down towards the seabed at an angle of 45°. Three PMTs make up one storey. Each of these lines houses 90 PMTs over 30 storeys and the distance between each storey is 12 m. Overall each line is 384 m high and each of them is connected to a junction box which in turn is connected to the shore station by an electro-optical cable. On May 30^{th} 2008 the Antares detector has been completed with the deploying of the last two lines, thus bringing the total number of detecting lines to twelve 2.1. The total instrumented volume of the telescope is now 0.1 Km^3.

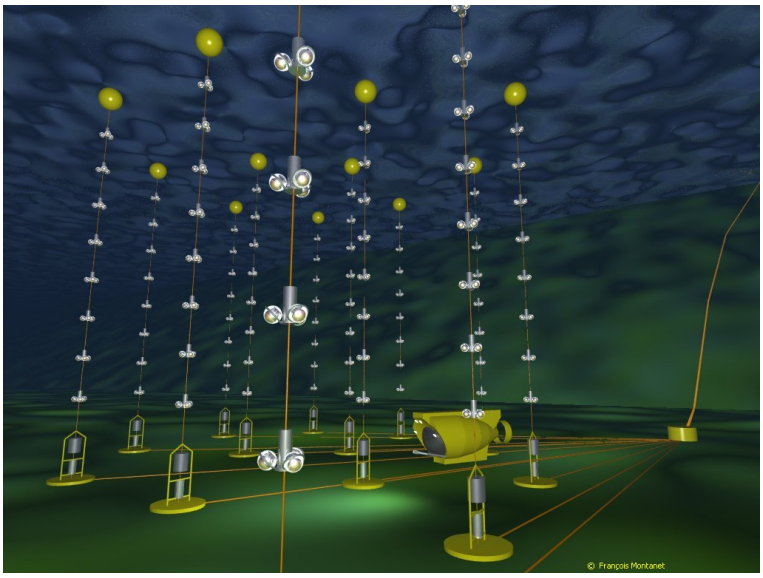

Figure 2.1: **3D virtual picture of the Antares telescope**. 12 lines of storeys instrument a volume of about 0.1 Km^3 of water, making the Antares telescope the biggest underwater neutrino detector up to now.

The default readout mode of ANTARES is the transmission of the time and the amplitude of any photomultiplier signal above a threshold corresponding to 1/3 of a photo-electron signal for each OM. Time measurements are referenced to a master clock signal sent from shore. All photomultiplier signals are digitized and sent to shore where they are processed in a computer farm to find hit patterns corresponding to muon tracks or other physic events producing light. The grouping of

three OM in a storey allows local coincidence to be used for this pattern finding. In addition the front-end electronics can acquire a 128 samples pattern at 500 MHz. Sampling and digitization of the signal is made through an ASIC chip, the *Analog Ring Sampler* [5]. A data acquisition card in each storey, containing an FPGA and a micro-processor, outputs the multiplexed signals of the three local OMs on an Ethernet optical link. DWDM (*Dense Wavelength Division Multiplexing*) is then used to transmit through optical fibers the whole data of each line.

2.1.2 NESTOR

NESTOR is another international collaboration, also involved in the Km3NET project, its acronym stands for Neutrino Extended Submarine Telescope with Oceanographic Research. NESTOR is a deep-sea neutrino telescope under construction in the southern Ionian Sea, off the coast of Greece.

The NESTOR collaboration has developed an approach to operating a deep-sea station, permanently connected to shore by an in-situ bidirectional cable, for multi-disciplinary scientific research.

Construction and deployment of such a multidisciplinary deep-sea station, at a depth of 4100 m was achieved in January 2002 [4]. This deep-sea station, developed in the project LAERTIS, also serves the purpose of being the bottom platform for a deep-sea neutrino telescope. It has been operated via an electro-optical cable for the power supply of the structure and for the data transfer to shore. Recovery and redeployment operations with payload exchange were performed. The structure was equipped with several sensors like thermometers, barometers, compasses, a water current meter and an ocean bottom seismometer.

An important feature of the deployment and recovery procedure developed by NESTOR lies in allowing the instrument package once deployed at the seafloor, to be recovered, modified or serviced at the surface and be deployed again, without having recourse to manned submersibles or remotely operated vehicles. The feasibility of this procedure has been demonstrated in repeated redeployments.

In March 2003, the NESTOR collaboration successfully deployed a test floor of the detector tower, fully equipped with 12 optical modules, final electronics and associated environmental sensors [22]. In this operation the electro-optical cable and the deep-sea station, previously deployed at 3850 m, were brought to the surface, the floor was attached and cabled and redeployed to 3800 m. The basic element of the NESTOR detector is a hexagonal floor or star. Six arms, built with

titanium tubes to form a lightweight lattice girder, are attached to a central casing. Two optical modules are attached at the end of each arm, one facing upwards and the other downwards. The electronics of the floor are housed in a 1 m diameter titanium sphere within the central casing. The diameter of the floor deployed in 2003 was 12 m. The optical module consists of a 15" diameter photomultiplier tube enclosed in a spherical glass housing which can withstand hydrostatic pressures up to 630 bar. To reduce the effect of the terrestrial magnetic field, the photomultiplier is surrounded by a high magnetic permeability cage. Its optical coupling to the glass sphere is made with glycerine, sealed by a transparent silicone gel gasket. Other modules, above and below each floor, house LED flasher units that are used for the calibration of the detector, controlled and triggered from the floor electronics.

On the base of these studies the structure proposed by the NESTOR collaboration is a tower of 12 hexagonal floors of 32 m diameter with large PMTs at the corner points. Each floor is located above the next at vertical intervals of 20 m as shown in Fig 2.2

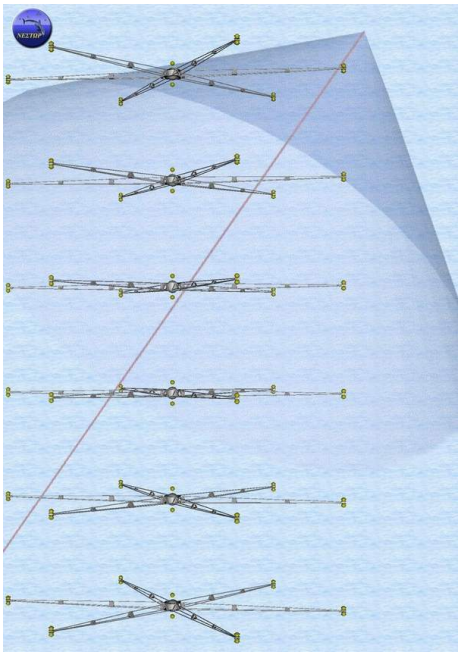

Figure 2.2: **The NESTOR proposed tower**. The star shape of each floor houses two PMTs on each tip.

2.2 The NEMO project

Starting from 1998 the NEMO collaboration has carried out R&D activities aimed at developing and validating key technologies for a cubic kilometer scale underwater neutrino telescope [32]. A first phase focussed on site investigation and characterization studies as well as the development of a suitable detector concept. The site characterization is described in more details in the next section.

Hence the R&D activities on the detector characterization were organized in two successive phases. During Phase-1 a demonstrator tower was installed at a test site close to Catania at a depth of 2000 m and verified the technologies. The Phase-2 project, which is currently under construction, aims at installing an infrastructure, comprising a 100 km electro-optical cable, a shore station and a full scale tower, at the Capo Passero site at a depth of 3500 m. The activity of these two phases is described in section 2.2.2 and 2.2.3.

Finally the NEMO km^3 detector concept is discussed in section 2.2.4.

2.2.1 Site investigations

A key point for the NEMO collaboration is the definition of an optimum site for the deployment of the apparatus. The site of the telescope should satisfy several environmental requirements:

- First of all the site should be deep enough for the removal of the atmospheric muon background, in order to increase the signal over noise ratio.
- Closeness to the coast is essential to reduce the expense of the power and signal cable connection to shore.
- The optical properties of water are fundamental to optimize the detection range of each optical module because the less are the absorption and scattering lengths, the more is the detection range of a module. Sea water should have a low absorption length in a wavelength rage from 350 nm to 550 nm, which optimize the transmission of Cherenkov blue light.
- The seafloor should be as smooth as possible for the whole large area that will be occupied by the detector in order to make easier the positioning procedure and to have a homogeneous spread of the detectable units.

- The site should present a low marine life activity involving bioluminescent bacteria and film accretion on the optical surfaces of the telescope. This will reduce the background noise and preserve the sensitivity of OMs during their lifetime.
- Low sedimentation rate is also important to increase the lifetime of the whole detector.
- Stable and low undersea currents would ease the deployment of the apparatus and would give it a still configuration during operation time. This means an accurate knowledge of the OMs position with a consequent more accurate reconstruction of the tracks. The overall mechanical stress of the structure would also be reduced.

Several years of investigations in different off shore areas of the Tyrrhenian and Ionian Sea managed to point out the deployment site showed in Fig. 2.3.

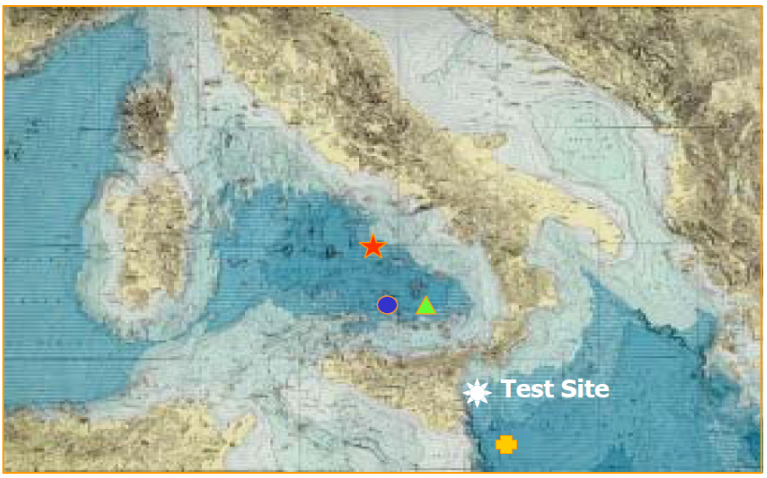

Figure 2.3: **Sites investigated by the NEMO collaboration**. In the Capo Passero off shore site were found the best overall properties. In the Test Site, during NEMO Phase-1, the demo tower was deployed.

A more accurate and continuous observation at the Capo Passero site of the optical properties of deep water has been taken, and the results [10] are shown in Fig. 2.4 and Fig. 2.5.

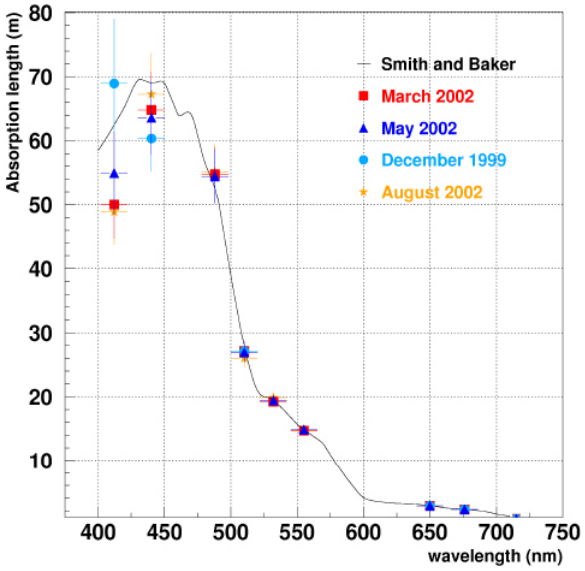

Figure 2.4: **Average absorption length at the Capo Passero site**. The Absorption length is plotted as a function of wavelength for 4 seasons.

2.2.2 NEMO Phase-1

The Phase-1 project started in 2002 and was completed in December 2006 with the deployment and connection of two components: the junction box and a prototype NEMO tower [16]. All key components of an underwater neutrino detector are included: optical and environmental sensors, power supply, front-end electronics and data acquisition, time and position calibration, slow control systems and onshore data processing.

The junction box provides connection between the main electro-optical cable and the tower. It has been designed following an innovative concept to withstand pressure and corrosion. The two issues have been decoupled by placing the electronics inside a pressure resistant steel vessel housed inside a fibreglass container filled with silicone oil, that is pressure compensated. Moreover, all electronics components

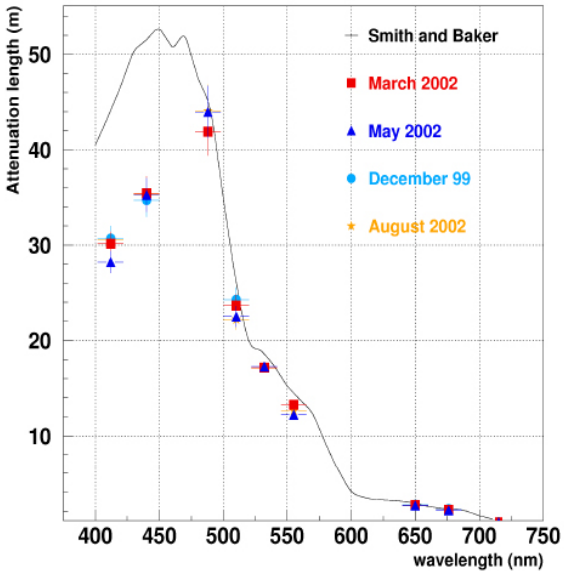

Figure 2.5: **Average attenuation length at the Capo Passero site**.

proven to withstand pressure in laboratory tests, have been placed directly in the oil bath. The double containment technology has the further advantage of preventing water ingress in case of failure of an internal connector or penetrator. A 3D reconstruction of the JB deployed for the NEMO Phase-1 experiment is shown in Fig. 2.6.

The prototype tower-like detection unit is a three dimensional flexible structure composed by a sequence of floors (that host the instrumentation) interlinked by cables and anchored on the seabed. The structure is kept vertical by appropriate buoyancy on the top.

While the design of a complete tower for the km^3 foresees more floors, the prototype under realization for the Phase-1 project is a *mini-tower* of four floors only, each made with a 15 m long structure hosting two OM (one down-looking and one horizontally-looking) at each end (in total 4 OM per storey). The floors are vertically spaced by 40 m. Each floor is connected to the following one by means of four ropes that are fastened in a way that forces each floor to take an orientation perpendicular with respect to the adjacent (top and bottom) ones. An additional spacing of 150 m is added at the base of the tower, between the tower base and the lowermost floor to allow for a sufficient water volume below the detector. A schematic of the prototype tower is

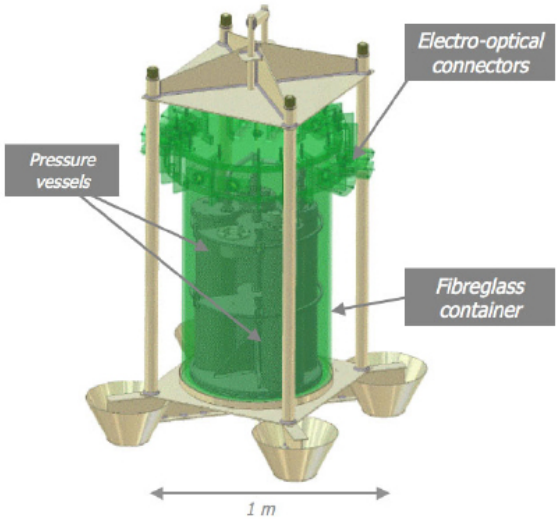

Figure 2.6: **3D image of the NEMO Phase-1 junction box.**

shown in Fig. 2.7.

In addition to the 16 optical modules the instrumentation includes several sensors for calibration and environmental monitoring. In particular two hydrophones are mounted on the tower base and at the ends of each bar. These, together with an acoustic beacon placed on the tower base and other beacons installed on the seabed, are used for precise determination of the tower position. The other environmental probes are: a Conductivity-Temperature-Depth (CTD) probe used for monitoring water temperature and salinity; a light attenuation probe (C*, pronounced C-star) and an Acoustic Doppler Current Profiler (ADCP) that provides continuous monitoring of the deep sea currents along the whole tower height.

The NEMO Phase 1 apparatus was successfully operated for several months after the installation. The installation operation allowed for full validation of the underwater connection concept and the "unfurling" technique. The power supply, data transmission and time and position calibration procedures were also validated. Power is distributed by means of a three phase AC system to each Floor Power Module, where a conversion to DC is made. The system has been designed to have most of its components working under pressure inside an oil bath [16].

Data transmission is based on a synchronous communication pro-

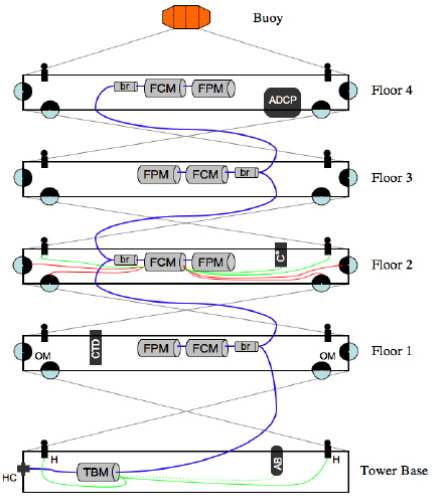

Figure 2.7: **Schematic of the prototype tower of the NEMO phase 1.** The NEMO Phase 1 experiment is deployed off the Catania shore and it is made up of only 4 floors. It started to be operated at the beginning of 2007.

tocol, which embeds data and synchronization and clock signals in the same serial bit stream [18]. The technology relies on Wavelength Division Multiplexing techniques, using only passive components with the exception of the electro-optical transceivers. The architecture of the data transmission system is based on a Floor Control Module located at the center of each bar, that collects data and streams them to shore. In the opposite direction, the Floor Control Module receives slow control data, commands and auxiliary information, as well as the clock and synchronization signals needed for timing. A picture of the FCM board is showed in Fig. 2.8

Time calibration is performed by measuring time delays from shore to each Floor Control Module from the propagation times of signals that are distributed via a network of optical fibres [28]. This system was demonstrated to provide an accuracy of 1 ns.

Since the tower structure can flex under the influence of sea currents, a determination of sensor positions is necessary. This is achieved by means of acoustic triangulation measurements using acoustic beacons placed on the seabed and a couple of hydrophones on each bar. Distances are calculated by converting the “time of flight” of acoustic

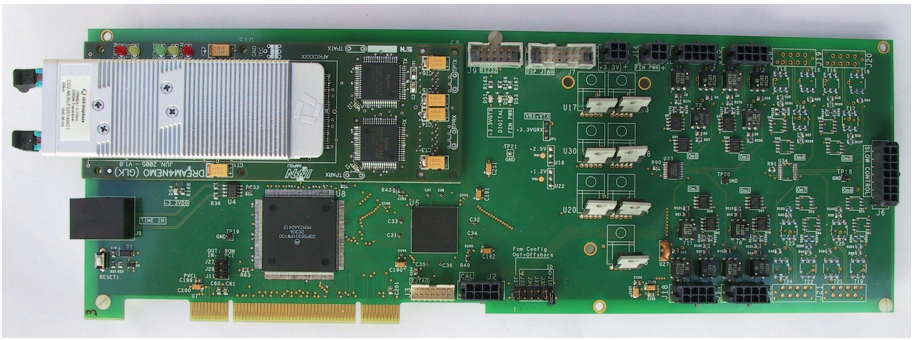

Figure 2.8: **Floor Control Module board.** It concentrates data from the 4 OMs of a floor on the optical link. The link is a point-to-point connection towards its on-shore counterpart, an equivalent board housed in a DAQ PC (the FCM-Interface).

pulses into lengths knowing the sound velocity in water. Time of flight is the difference between the time of arrival on the receiver hydrophone and the time of emission on the beacon. To achieve the requested accuracy of 15 cm the time of flight has been measured with accuracy better than $10^{-4}s$. In addition the inclination and orientation of each bar is measured by a tiltmeter and a compass.

Using the NEMO Phase-1 installation down-going muon tracks have been reconstructed; an example is shown in Fig. 2.9

On the NEMO phase-1 test site in the bay of Catania a geo-seismic station has been deployed and connected to the electro-optical cable of the NEMO underwater infrastructure by the INGV[4] in January 2005. This station, which includes a seismometer, a magnetometer and several water environmental probes, is the first working node of the ESONET[5] network.

On the same site a set of hydrophones has been installed to test the feasibility of acoustic detection of high energy neutrinos. They were operated from January 2005 to December 2006 and provided a large set of deep-sea acoustic data. Analysis of the acoustic data allowed the detection of sperm-whales at a distance of more than 40 km, revealing a population larger than previously estimated.

[4]Istituto Nazionale di Geofisica e Vulcanologia
[5]European Seafloor Observatory Network

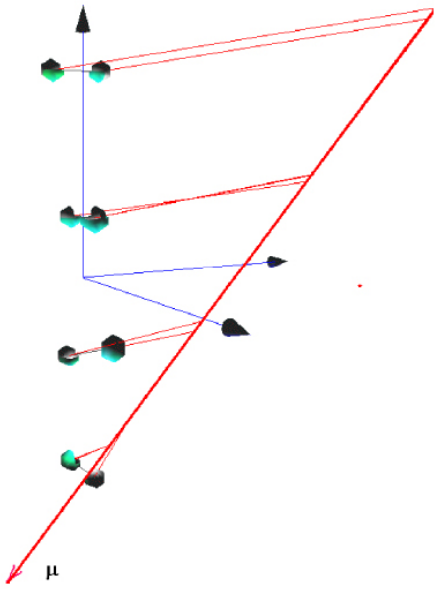

Figure 2.9: **Reconstruction of a down-going atmospheric muon track.** Event reconstructed with real data acquired during the operation of the NEMO Phase 1 detector.

Data transmission system

Timing performance is a key feature for an accurate reconstruction of an event track, which means keep the whole kilometric cube structure in synchronous for a long period and with an accuracy of less than a nanosecond. By means of that, communications are all synchronous, and a master clock at 4.86 MHz is delivered to all the OMs. All the others clocks, such as the 100 MHz sampling frequency and the 19.44 MHz uplink clock, are derived by the master clock and kept in phase by sharp PLLs (Phase Locked Loop).

The interface between the FCM and an OM is a proprietary protocol based on 8b/10b modulation [47]. A more precise description of all the layers of this protocol will be given in section 3.3.

On the other hand, for the main data transfer over the seabed dorsal a standard synchronous protocol, developed for telecommunications, has been adopted. The architecture of the Phase-1 transmission system is in fact a point-to-point connection (OM → onshore buffer), where the OM data streams of a floor are encapsuled in a unique data flux: a concept very similar to a set of independent phone-calls running on

a single backbone and de-multiplexed to the destination telephones.

The standard adopted is the SONET/SDH protocol (Synchronous Over Network/Synchronous Digital Hierarchy). This standard is thoroughly defined by ITU-T recommendations and represents the de facto protocol adopted by all modern telecommunication systems, supporting data-rates over 10Gbps. Beyond the physical layer, a protocol layer is defined as well: user data are merged with an overhead stream packet which manages, controls and implements the details of the communication over the selected link.

One specific transmission scheme, namely STM-1, was again selected among the many offered by SDH format; this provides a total raw data rate of 155.52 Mbps. The basic aggregate data unit is called frame and lasts 125 ms; therefore, changing data rate means modifying the number of transmitted bytes per frame: the STM-1 frame consists of 2430 bytes. Actually the useful data available to the user (payload) consists of 2340 bytes per frame, which yields 18.72 MB/s, or about 150 Mbps [23]. This capacity allows the static allocation of up to eight logical channels (each holding a maximum of about 2 MB/s), one for each FEM board. This means that an FCM board can theoretically manage up to 8 OMs but for the Phase-1 project only four channel were used.

As previously mentioned, each FCM is connected to its counterpart on shore. Each of the on-shore FCM is plugged one into a dedicated PC (FCM-Interface) and it is accessible via a 32-bit, 33 MHz PCI bus.

The OM data streams are then unpacked and transmitted by the PCI bus to the computer central memory where they are stored in a corespondent set of FEB (*Front End Buffers*). Data in each FEB can be accessed by forthcoming processes like time-wise alignment, direct muon track event-triggering, or just data displacement into a new concentrating machine.

Even if the physical link which connects two FCM is bi-directional, the stream of data flows only towards the shore. At the onshore laboratory data are extracted and stored in a buffer of the FCM-I and sent over an Ethernet connection to the Data Manager server. Fig 2.10 show a diagram of the dataflow related to one floor in the NEMO Phase-1 mini tower.

At the Junction Box level, data coming from the 4 floors are dropped into a single multi-modal fiber, exploiting standard DWDM (Dense Wavelength Division Multiplexing). In the adopted scheme, carrier frequencies fall in the range between 192.1 and 196.1 THz with 100 GHz spacing. Each DWDM channel hosts a STM-1 FCM-to-FCM link. In

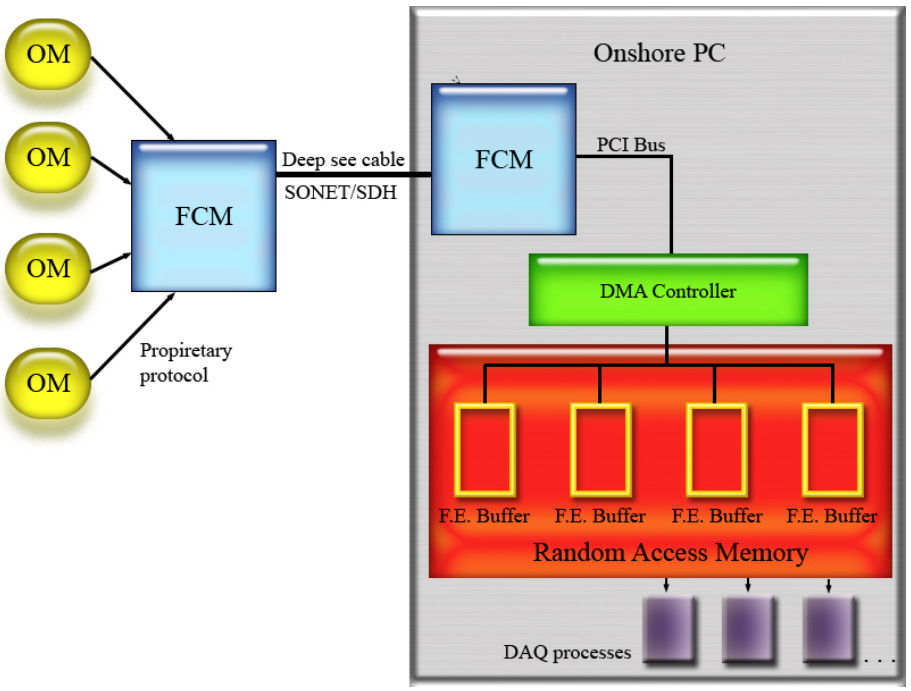

Figure 2.10: **Data-flow diagram for the NEMO Phase-1 telescope.** This picture show the key elements and protocols involved in the data transmission concerning one floor. The Onshore PC is also called FCM Interface.

this configuration the four data flows, traveling towards the onshore FCMs through four dedicated STM-1 logical links, are DWD Multiplexed onto a single optical fiber. The data flow towards the offshore FCM is hosted by a different fiber. Some Bit Error Rate measurements have been done: considering the two furthest network entries (i.e. the uppermost floor and its onshore counterpart), an additional attenuation level of more than 24 dB still guarantees an estimated bit error probability less or equal to 10^{-9} [23].

The Slow Control System

Environmental sensors, infrastructure diagnostics and positioning system are parts of the so called Slow Control System, as well as everything which not concerns the mere data transfer. Even if this system shares the same optical-link with data, it travels on a different application protocol. In this case the communication needs to be bi-directional because all the configuration parameters are sent to the tower via the Slow Control Management System installed onshore.

Hence the main application fields of the SC system are monitoring and configuring, purposes which require a much more narrow bandwidth if compared to that allocated for data transmission. For this reason it is called *slow* control.

Several sensors are housed inside the OM to make possible the monitoring of temperature, humidity, current flow etc. In addition, at each floor there is also a dedicated SC Board which connects via RS232 to the FCM. This board controls several instruments of the floor such as the Hydrophones, the CTD, the ADCP etc [41].

In figure 2.11 there is a scheme of the whole on shore infrastructure including the Data Manager server and the Slow Control System Manager [31]. As the SC server is located at the Laboratori Nazionali del Sud, an INFN laboratory, a radio link was installed for the communication towards the shore station. A web server made possible to monitor the whole apparatus from any node of the Internet.

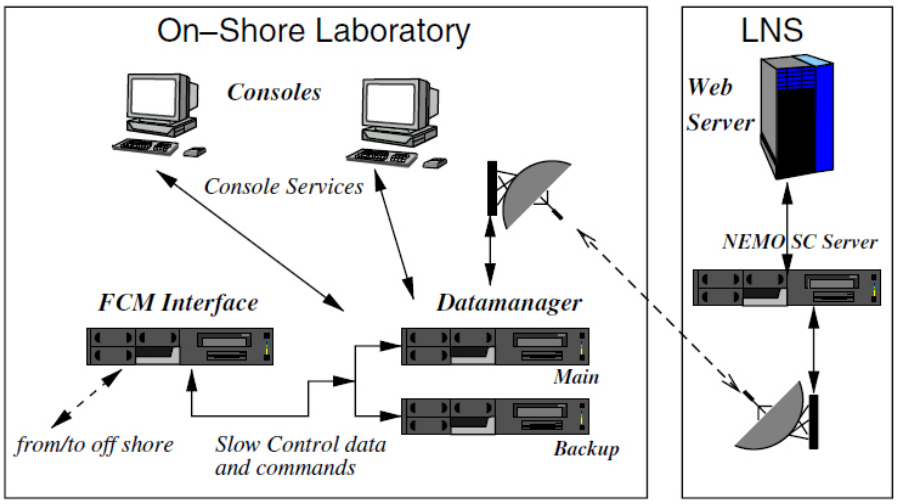

Figure 2.11: **Onshore hardware architecture.**

2.2.3 NEMO Phase-2

In the NEMO Phase-2 project, a complete tower structure with 16 storeys will be constructed at a depth of 3500 m at the Capo Passero site.

For the integration of this full-size tower is foreseen to dedicate 2 floors to new R&D projects. The effort of this thesis is collocated in

this working-area as it concerns a new hybrid approach to the PMT signal acquisition, details in section 3.3.

The NEMO Phase-2 deep-sea infrastructure includes a 100 km long electro-optical cable, laid in July 2007, which links a shore station, located in the harbor area of Portopalo di Capo Passero to an underwater infrastructure needed to connect detector prototypes. The shore station, hosting power supply and data acquisition systems, will also include integration and test facilities for the detector structures.

A DC power system with sea return was chosen. The main electro-optical cable, manufactured by Alcatel, carries a single electrical conductor, that can be operated at 10 kV DC allowing for a power load of more than 50 kW, as well as 20 single-mode optical fibres for data transmission. The Phase-2 infrastructure will include a cable termination assembly with a 10 kW DC/DC converter to 400 V [43]. This system will be deployed at the end of 2008.

The experience gained with the Phase-1 tower has led to a revision of the design aimed at simplifying the tower integration and reducing construction costs. The major changes concern: the electro-optical backbone, with a new segmented structure that allows an easier integration; the integration of all the electronics, power systems and fibre breakouts in a single pressure vessel; a revision of the time calibration system to eliminate fibres along the storey. The power system has also been modified to comply with the new DC design. The completion of Phase-2 at the end of 2008 is essential for a full validation of the deployment and connection techniques and of the functionality of the system at a depth of 3500 m. At the same time it will permit a continuous long term monitoring of the site properties.

2.2.4 The NEMO Km^3 telescope

The NEMO detector concept is based on a 9×9 square grid of uniformly spaced Detection Units. The spacing between these is 200 m in both directions to cover a surface of 2,5 km^2. The detection unit elements proposed is the *tower* as described in the previous section and which is still under investigation in recent NEMO Phase-2 experiment.

Each of these DUs has two optical modules at both ends like the Phase-1 tower (§ 2.2.2), and contains instrumentation for positioning and monitoring of environmental parameters. A tower consists of 16 bars of marine grade aluminum, 12 m long, interlinked by a system of ropes. The whole structure is anchored to the seabed and kept vertical by appropriate buoyancy on the top, see Fig. 2.12. The spacing between floors is 40 m, while an additional spacing of 150 m is added

between the anchor and the lowermost storey. The structure is designed to be assembled and deployed in a compact configuration, and unfurled on the sea bottom under the pull provided by the buoy. Once unfurled the bars assume an orthogonal orientation with respect to their vertical neighbors.

The power and readout is provided by light-weight electro-optical cable that is kept separated from the system of tensioning ropes in order to reduce interference with the mechanical structure. Optical fibre technology is used for data transfer. The towers are connected through a network of undersea cables and junction boxes and a single main electro-optical cable to shore. The towers are connected to the junction boxes through underwater wet-mateable electro-optical connectors operated by a remotely operated vehicle (ROV).

A hierarchic tree of *Junction Boxes* collects data from the whole telescope and distributes DC-power to each tower. A junction box is essentially a pressurized vessel containing power electronics as DC-DC transformers to provide a down-step in output voltage and data-concentration electronics to aggregate information coming from several detection units or other JBs.

In the Km^3 scheme there is a net made up of 9 secondary JBs and 1 primary JB. The primary JB is then connected to the main electro-optical cable that is the seabed dorsal which brings power and transmit all the data to the shore. A schematic of this global layout is presented in Fig 2.13.

The next chapter will focus on the front-end electronics of the optical modules, and the NEMO solution will be presented.

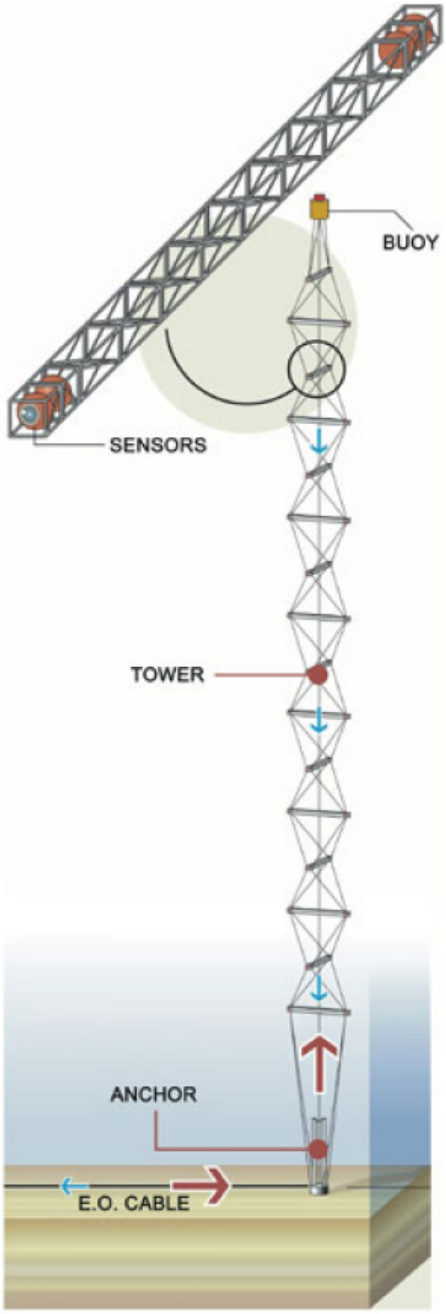

Figure 2.12: **The NEMO tower architecture**. 16-floors tower, the structure proposed by the NEMO collaboration.

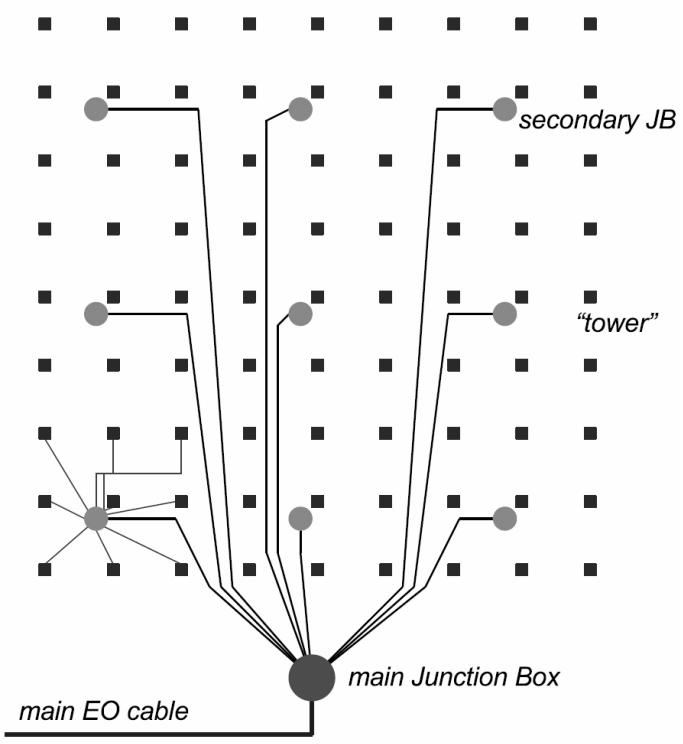

Figure 2.13: **Layout of the NEMO Km^3 telescope.** The telescope is composed by a square grid of towers. A redundant net of junction boxes is responsible for power distribution and data concentration. Elettro-optical cable brings power and data connectivity between the telescope and the shore.

Chapter 3

Optical module front-end electronics

The *Optical Modules* (OM) are the sensitive part of the experiment, and for this reason they take an important part in the R&D of the detector. This chapter will focus on the architecture of the OMs developed by the NEMO collaboration and especially on their read-out electronics. The optical module concept, that will be described in section 3.1, has been practically validated by the Phase-1 experiment and it will be integrated again on the full-size tower that is going be deployed at the Capo Passero site during NEMO Phase-2.

The OM data acquisition electronics, used in the Phase-1 mini-tower, are characterized mainly by a fast ADC which samples the PMT anodic signal after a logarithmic compression performed by a calibrated diode. It will be discussed in section 3.2.

An alternative hybrid solution has been studied, exploiting an analog delay line as a sampler of the anodic waveform. It will be shown that it is a possible solution which can grant less power consumptions and an improved linear dynamic range. This work was took on by the collaboration between the INFN sections of Bologna and Catania, and it led to the production of a mixed signal board fully compliant with the specification of the NEMO phase-1 OM requirements. This board remains a case-study applied to the latest version of the analog sampling chip LIRA (Italian Acronym for *Analog Delay Line*) developed in Catania, but it has been a milestone in the design of a more complex chip (SAS [14]), whose architecture logic is meant to be deployed and tested during the Phase-2 experiment.

3.1 Architecture of the Optical Module

The OM is essentially composed by a PMT enclosed in a 17" pressure resistant sphere of thick glass called Benthos-sphere. The PMT used in Phase-1 is a 10" Hamamatsu R7081Sel with 10 dynode stages. In spite of its large photocathode area, the Hamamatsu PMT R7081Sel has a good time resolution of about 3 ns FWHM (*Full Width Half Maximum*) for single photoelectron pulses with a charge resolution of 35%. Mechanical and optical contact between the PMT and the internal glass surface is ensured by an optical silicone gel. This gel has very good light transmission properties and a refractive index close to that of the sea water, of the glass sphere and of the photomultiplier's glass window. It is also sufficiently elastic to absorb shocks and vibrations during transport and deployment and to support the deformations of the glass sphere under pressure.

The PMT is shielded from the Earth magnetic field by a μ-metal cage. The terrestrial magnetic field affects the trajectories of the electrons in the photomultiplier, especially between the photocathode and the first dynode. The wire cage made of μ-metal, which is a nickel-iron alloy with very high magnetic permeability, can reduce this effect. The shape and the size of this cage has been designed to minimize shadowing effects on the photocathode (2%).

The base card circuit for the high voltage distribution (Iseg PHQ 7081SEL) requires only a low voltage supply (+5 V) and generates all necessary voltages for cathode, grid and dynodes with a power consumption of less than 150 mW. A 3D reconstruction of the Benthos-sphere is showed in Fig. 3.1.

3.2 Front End Module Board

For the typical spectrum of a PMT Single Photo-Electron (SPE) signal, it has been chosen to sample the anodic output at a rate of 200 MSample/s. Due to Nyquist theorem the PMT signal spectrum has been limited with a 100MHz low-pass anti-aliasing filter. The expected timing error on the samples due to clock jitter and quantization noise is expected to be of the order of 300 ps [17].

Concerning the signal dynamics, assuming that a resolution of 4 bit (16 converter's channels) is sufficient for the acquisition of a SPE event, and assuming that the system must be able to acquire as much as 300 contemporary photo-electron, it comes out that the necessary conversion dynamics is about 75 dB, which would require at least a

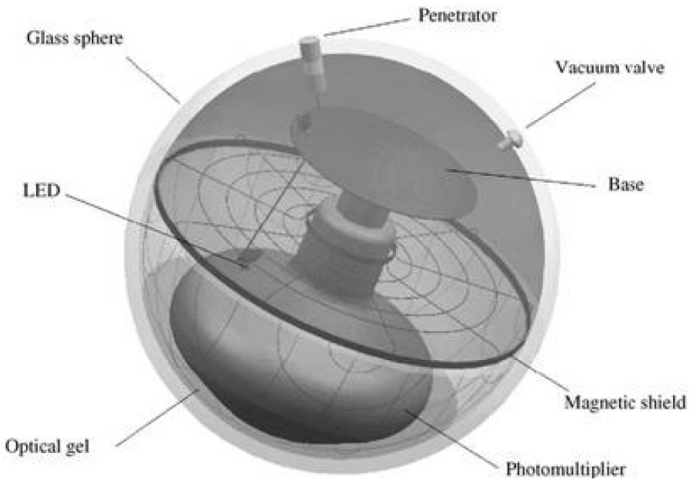

Figure 3.1: **3D image of the optical module**.

13-bit analog to digital converter. In order to keep power requirements as low as possible an 8-bit ADC has been chosen and, in order to match the expected signal dynamics, a passive analog compression stage has been interposed between the PMT signal and the digitizers. This quasi-logarithmic compression is realized with a calibrated diode and the circuit is shown in Fig. 3.2 [38].

Instead of using a single 200 MHz analog to digital converter, a couple of 100 MHz ADCs have been adopted to lower the power consumption. The samplers are driven by the same clock staggered of half a period thus yielding an actual sampling rate of 200 MHz. The 100 MHz clock is generated by a zero-phase PLL which multiplies by a factor of 20 the 5 MHz signal received from the FCM.

A continuous sampling would produce a total data rate of 1.6 Gbps, nevertheless such a huge bandwidth would be wasted: the interesting events are SPE peaks, which have a mean time duration of about 50 ns and whose mean rate, dominated by ^{40}K decay, is expected to be of the order of 50 KHz. This would mean an average waste of 99.75 % of the bandwidth (10 8-bit samples every 20 μs occupy a bandwidth of 4 Mbps only).

In order to reduce the bandwidth waste, a zero-skipping programmable digital threshold has been implemented. This feature is realized by an FPGA, a Xilinx Spartan2 device. The samples which come out

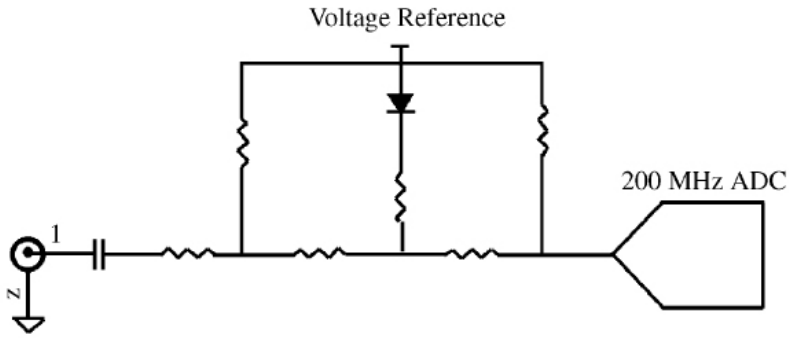

Figure 3.2: **Compression stage circuit**. The analog compression in the Front End Module of the NEMO Phase-1 mini-tower is performed by a diode.

of the digitizers are stored inside a FIFO (First In First Out) memory. When a sample over threshold is detected an *event* is pushed into another FIFO. An event consists of a 16-bit time stamp, two pretrigger samples, the over-threshold following samples, and a user programmable (from 0 to 15) number of the following under-threshold samples. In this architecture the acquisition is always on, as the digitizers are working continuously, and data discrimination is performed inside the FPGA by the digital logic.

The slow control subsystem is managed by a DSP, which boots the FPGA at power-up loading the bitstream from a Flash EEPROM. It is directly connected to the FPGA, the monitoring sensors, the calibration DAC, the High Voltage control DAC and the High Voltage ADC monitor. It also integrates an RS232 UART for bench debugging.

Communication towards and from the shore is managed by the FPGA which interacts with the Floor Concentrator Module (see 2.2.2). The link consists of three shielded twisted-pair cables. One is dedicated to the 5 MHz clock signal, distributed with a *Low Voltage Differential Signaling* standard (LVDS). The remainders are a down-link at 5 Mbps and an up-link at 20 Mbps that use RS485 differential standard, re-engineered to carry a DC component for the board powering.

Being this connection the present NEMO FCM-OM digital interface, it has been implemented as well in our LIRA DAQ-Board and so will be discussed in more detail in section 3.3.

In Fig. 3.3 there is a picture of the NEMO Phase-1 Front End Module Board[38].

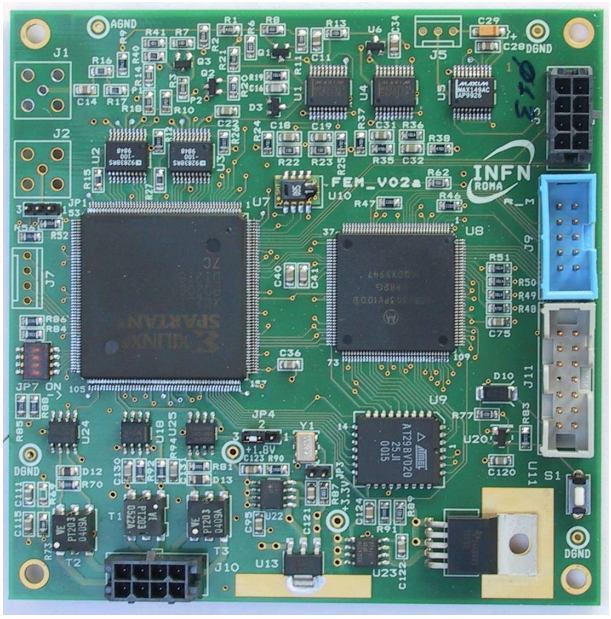

Figure 3.3: **The NEMO Phase-1 Front End Module Board**.

3.3 An alternative solution: the LIRA DAQ-board

The INFN microelectronic group of Catania started with the INFN section of Bologna the study of a different front-end architecture based on the analog sampling of the signal. It was proposed, and successively realized, a full-custom ASIC (*Application Specific Integrated Circuit*) for a multichannel analog acquisition of PMT signals.

This hybrid solution is meant to bring considerable improvements in the acquisition performance such as a wider range of linearity and a lower dead-time. Furthermore, the high level of integration of the front-end chip suits more the scaling towards a km^3 detector.

The first step of this project has foreseen a series of chips which integrates almost only the analog acquisition pipeline and the discriminator of the events, relaying its control to external digital programmable logic.

The Catania group started a collaboration with the INFN section of Bologna for the realization of a demonstrative board integrating the analog acquisition technology and a digital control environment. The horizon of this research is a low power System On Chip (SOC) that integrates both the analog acquisition device, the data converter and

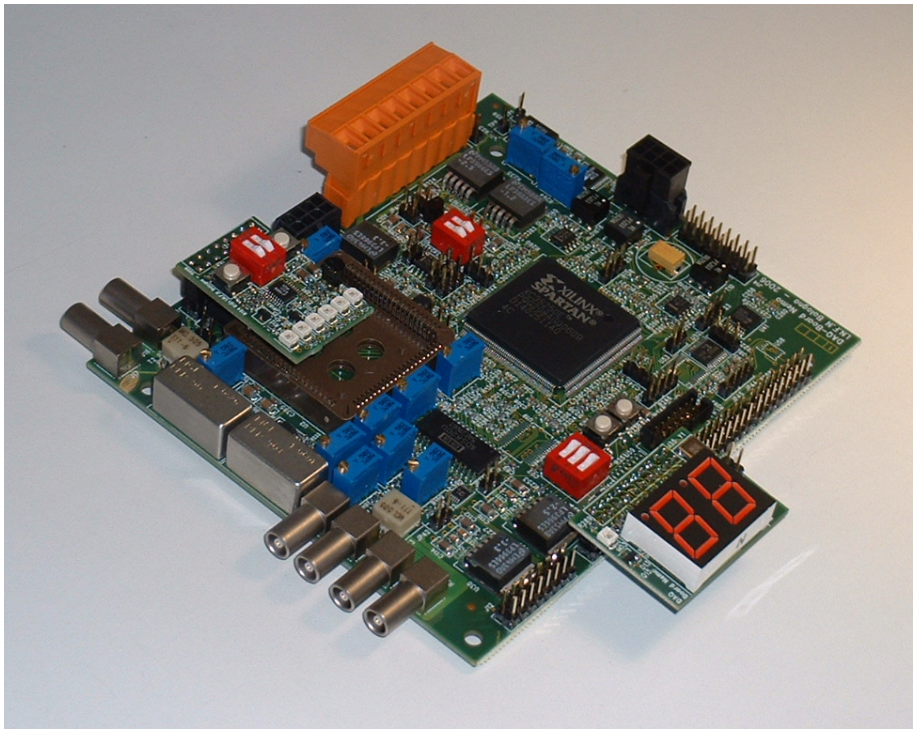

Figure 3.4: **DAQ-Board for the chip LIRA v0.6**.

the digital logic for data communication and acquisition management.

In this context we realized a board fully compliant with the mechanical requirements of the OM's housing and retention system and compatible with the data transmission specification of the NEMO Phase-1 mini-tower. In Fig. 3.4 is presented a picture of the board we have realized.

The following sections give a description of the DAQ-board, starting from the chip LIRA (*Analog Delay Line*) and moreover introducing all the other components, such as the digitizer, the FCM interface and last, but not least, the FPGA (*Field Programable Gate Array*). The FPGA is a modern programmable logic device whose flexibility has been exploited for the realization of a custom readout architecture.

Part of the work realized for this thesis concerned the physical integration of all these components on a PCB-board and the realization of an optimized firmware for the FPGA device.

In the last section of the chapter this custom firmware will be presented, discussing the management of the external components and the implementation of the data transmission protocol.

3.3.1 The chip LIRA

The goal of this project is to realize a front-end capable to increase the linear dynamic range of the sampled signal which would require, as previously said, at least 13 bits for a 300 PE pulse (§ 3.2). The idea is to sample the same physical signal contemporary at different gain levels in order to have two linear representations of the same event. In this way it is possible to choose a-posteriori which is the better set of samples to send onshore.

Each PMT pulse is then acquired both from the anodic output, where the gain factor is maximum, and from one of the last dynode where the gain factor is lower. In this way we can have a linear representation even for those signals that would saturate the anodic channel.

In a full-digital solution this would require at least 2 fast ADC at 200MHz, which means four at 100 MHz (§ 3.2). The benefit obtained wouldn't be worth the total increase in power dissipation and cost. By means of this it has been realized a custom integrated circuit that contains two double-channel *Analog Delay Lines* used in multi-buffer mode in order to decrease dead time. Multi-buffer means the following: since an analog delay line can't sample while it is in read-mode, the second instance acts as an event buffer while the first is emptied out and vice-versa.

A threshold trigger and classifier has been integrated as well in the LIRA chip called T&SPC Unit (*Trigger and Single Photon Classifier*). This element is responsible for the acquisition level-0 triggering and for the labeling of each event. The label associated to an event is fundamental during the readout of the chip LIRA as it indicates the number of samples to be taken for that event and point out the dynamic channel to be digitized. A time threshold indicates how long the signal lasts, and hence how many samples are to be taken, then a two-level voltage threshold point out which is the dynamic channel to be digitized. The choice we introduced, justified by the observation of typical PMT background events, is to take 10 samples for the SPE events from the anodic channel.

By the use of the T&SPC unit, we made possible to switch on and off the analog acquisition channels online, with a consequent further power saving. As the trigger discrimination and classification of the waveform takes about 60 ns, a delay line of about 80 ns must be added at the inputs of the two Analog Delay Line (see Fig. 3.5)[13]. This means that 4 pre-trigger samples are taken for each event, these are useful for a good shape reconstruction of the pulse.

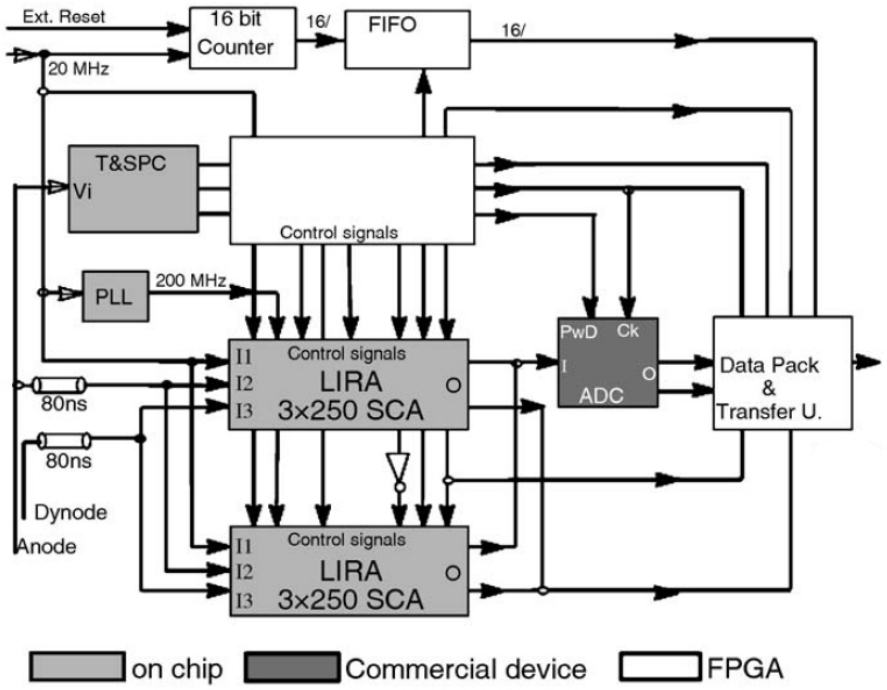

Figure 3.5: **LIRA (ver. 06) block diagram**. The bright grey elements are integrated on chip, while the others are implemented with commercial devices.

The Analog Delay Lines are realized with an array of 250 switched capacitors (SCA), in which each capacitor retains an analog sample of the signal. A fast discretization in time takes place (200 MHz), while each analog sample will be digitized only later, during the read operation, at lower speed (10/20 MHz) still aiming to keep low the power consumption. As previously said, a mean background rate of 50 KHz, mostly composed by SPE signals (50 ns long), generates an acquisition duty cycle of only 0.25%, the remaining 99.75% of time is available for the reading phase.

The generation of the 200 MHz acquisition clock is realized by a PLL integrated in the chip locking on a 20 MHz input frequency (Master Clock[1]). Since all the digital logic is running externally at 20 MHz, accurate time labeling of each event have to be realized inside the chip. The analog technique developed required the add of another

[1]This is the chip Master Clock, which is different from the board MC which will be introduced later on

SCA channel, and it works as follows: the Master Clock square wave is sampled in the third channel of the Analog Delay Line together with the pulse signal. Ten samples taken at 200 Mhz cover a whole period of the 20 MHz clock oscillation, allowing to reconstruct the precise position of each pulse sample with a resolution of 5 ns.

3.3.2 The analog circuits

The DAQ-Board is a mixed signal PCB with separated powering and ground planes in order to avoid the interference of the digital noise on the analog devices. The chip LIRA is the most important but not the only analog device on the board. For example, a simple signal conditioning circuit has been integrated to adapt the PMT pulse to the dynamic range of the chip inputs. The main digitizers as well needs very clean references for an accurate sampling of the LIRA outputs. In addition, two other signal converters are employed on this board for the control and monitoring of the PMT High Voltage Power Supply Unit (HVPSU).

The signal conditioning circuit

The main purposes of this circuit are to reverse the signal polarity in order to have a positive pulse and to filter the band of the incoming signal. A positive pulse is needed as the chip LIRA works only with positive voltage values, and the low-pass filters on the inputs are needed to avoid aliasing. The filters used, *PLP90* manufactured by Mini-Circuit[2], cut frequencies above 90 MHz with a loss $>$ 20 db for frequencies between 120-160MHz.

For the polarity inversion and galvanic decoupling two inductive transformers were used. Their conversion ratio is 1:1 and the bandwidth granted is 0.004 $\rightarrow$ 300 Mhz, with an insertion loss $<$1.10 db in the range between 0.02 $\rightarrow$ 300 MHz.

In this circuit the 80 ns delay lines mentioned above, are not integrated on board, but can be added or bypassed using two couples of LEMO mono-polar connectors.

The circuit schematic for the filtering and eventual delay of the anodic signal is showed in Fig 3.7.

[2] www.minicircuits.com

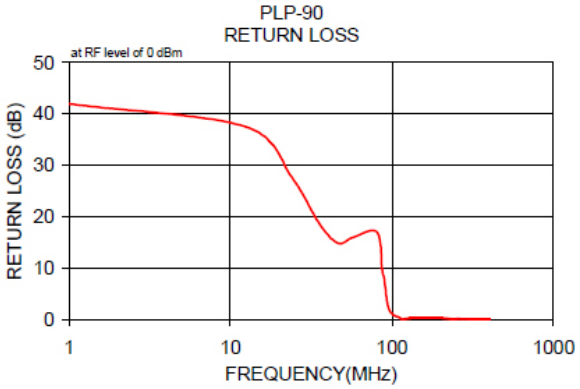

Figure 3.6: **PLP90 return loss graph**.

The DC power extraction circuit

In the NEMO Phase-1 project, power is delivered to the OMs on the same medium used for digital data lines. Digital protocol uses a differential communication standard (RS485), it is then possible to exploit the DC component of the signal, in other words its common mode, to deliver power to the module. A circuit for the DC component extraction from the signals has been integrated, using as reference the FEM board (§ 3.2) of NEMO Phase-1.

The uplink medium is provided with a 5V common-mode extracted to feed the main Vdd power plane of the board; grounding is provided in the same way by the downlink. The main power source of the DAQ board is then ensured by the voltage gap between the two common modes. A 1:1 pulse transformer on the front-end of each transmission line performs the DC extraction as shown in the schematic of Fig 3.8.

The primary coil of each pulse transformers is connected to the ends of the twisted pair cable coming from the FCM. On the central contact of the primary coil is then extracted the common mode voltage of the differential signal.

We said that the common modes of the transmission lines are put to 0 and 5V, which do not agree with the related RS485 standard. On the central contact of the secondary coil is then imposed a fixed voltage by a resistive ladder in order to bring back the signal in the standard range. After the DC extraction and the re-adaptation of the signals,

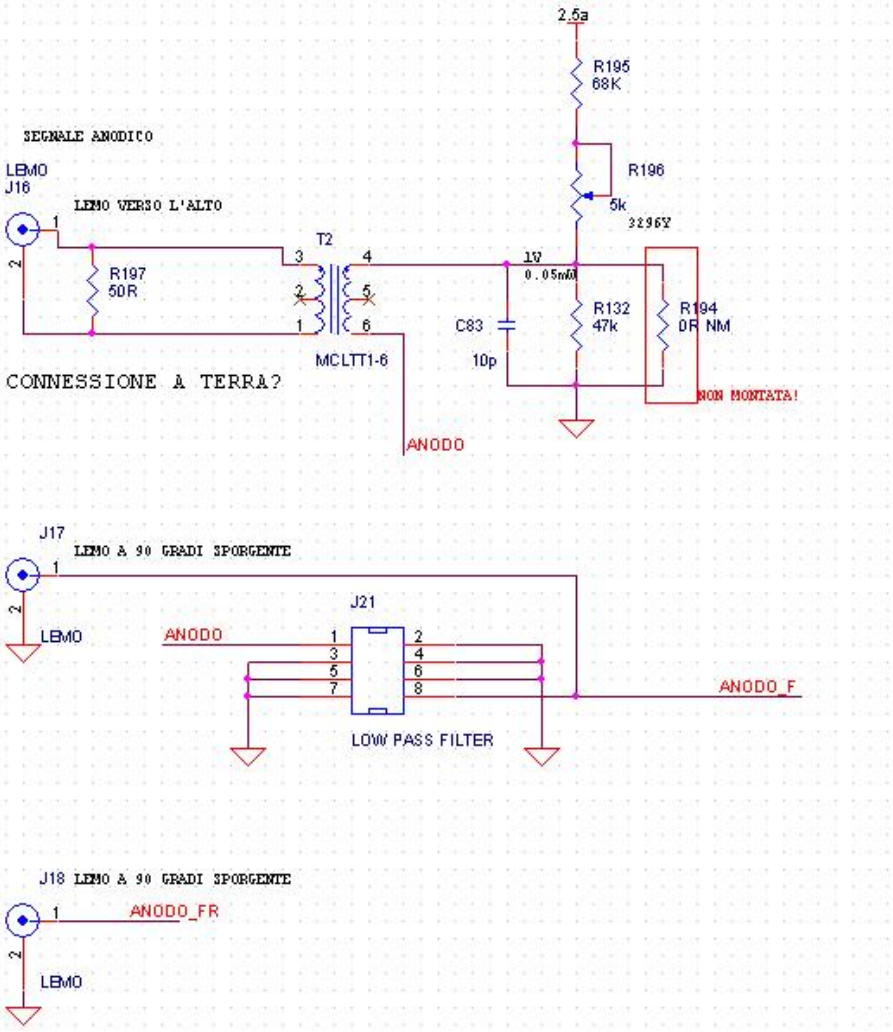

Figure 3.7: **Filtering and open delay loop for the anodic signal**. The incoming anodic signal is formerly decoupled from DC components by the use of an inductive transformer (T2) and then passed into a passive low-pass filter. The output is then put on an external delay loop to be realized between the LEMO connectors J17 and J18.

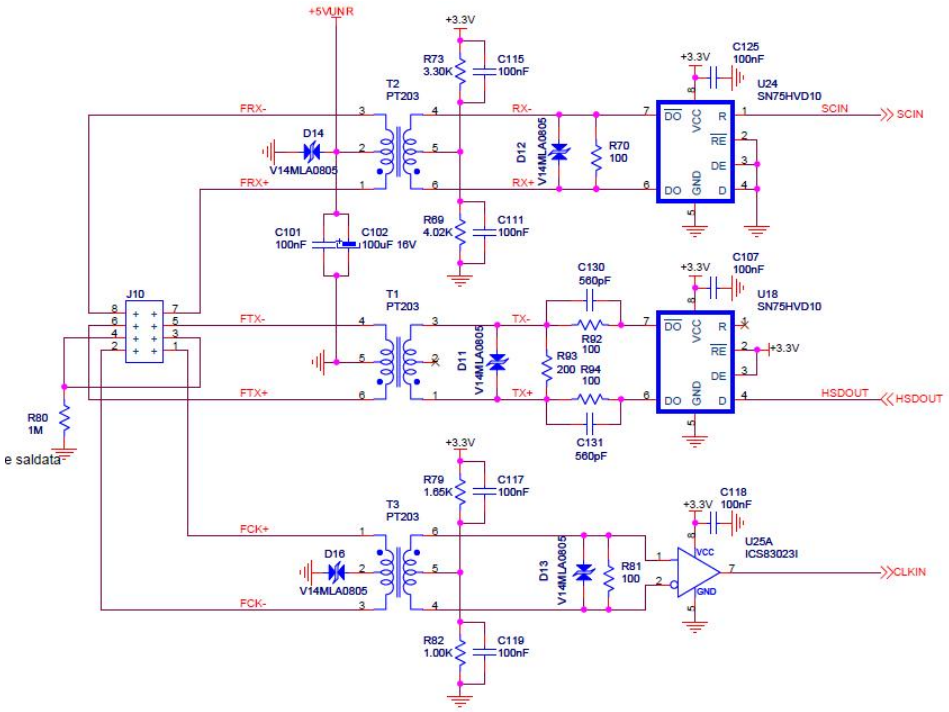

Figure 3.8: **DC component extraction and decoupling circuit**. The three data lines end on PT203 pulse transformers for galvanic decoupling. The common mode voltage of the data lines are then extracted from the central pin of the primary coils in order to to provide the GND and the +5V(unregulated) power supply.

a commercial RS485 transceiver[3] is used to interface the transmission lines to the FPGA I/Os.

The introduction of pulse transformers implies the use of a physical-link protocol that avoids DC and low frequency band components. In a DC-free transmission line they would completely distort the original signal. Hence the 8b/10b DC-free protocol has been employed to comply with NEMO Phase-1 specification [23]. The implementation of this protocol is realized by the firmware residing in the FPGA and then it will be described in the relative subsection.

Once the main 5V is on board, a series of linear regulators are used to create all the tension levels that the board requires. A main 5V stabilizer is used, then a couple of 3.3V and of 2.5V regulators provide

[3]Texas Instrument SN65HVD30

power separately for the analog and the digital circuits.

For bench test purposes an auxiliary power connector has been foreseen. It gives access to all the power and ground planes of the board allowing to power them independently. This technique was useful to evaluate the power consumption of each section of the boards.

The PMT monitor and control interface

The *High Voltage Power Supply Unit*, mounted at the base of the PMT, is controlled an monitored by the DAQ Board. A DAC is used to set the control voltage of the HVPSU in a range between 0 and 2V, this tension will be then multiplied by a factor 1K. The voltage ramp during the power on and power down of the PMT will be digitally controlled directly by the FPGA.

A feedback of the HVPSU output is provided in a low voltage signal. This analog value, ranging as well between 0 and 2V, represents the high voltage on the anode of the photo-tube divided by a factor of 1K. This value is monitored by a dedicated slow ADC connected to the FPGA. The converters adopted to this purposes are the Analog Device AD5330 and AD7810.

The analog multiplexer

It has been discussed that the LIRA chip presents two instances of a three-channel buffer, and that all the channels of an instance work perfectly synchronous. This means that they are all written and read in parallel by a common control logic. Since only one instance at a time is in read mode and since only one of the dynamic channels has to be digitized for an event, an external analog multiplexer has been introduced that selects one of the two dynamic channels of one of the two instances. In this way we avoid the waste of 3 ADCs.

As the third Analog Delay Line channel samples digital CMOS signals of a 20 MHz square wave, they don't need to be converted, and thus both clock channels are connected directly to the FPGA.

The analog MUX is a 4-to-1 Analog Device ADG704, with fast switching times of the order of 20 ns connected to the signal's output of the two buffers. It is driven by the FPGA that knows in advance which sample of the event, and which event is coming out of the LIRA buffer: in case of a SPE event it will connect the ADC input to the anodic channel. It will be held in this position for a duration of 10 samples, that at 10 MHz rate it means 10×100 ns $= 1$ μs. Otherwise, in case of a high-dynamic NSPE event, it will connect the ADC to the

dynodic channel for the necessary time. The scheme is presented is Fig.

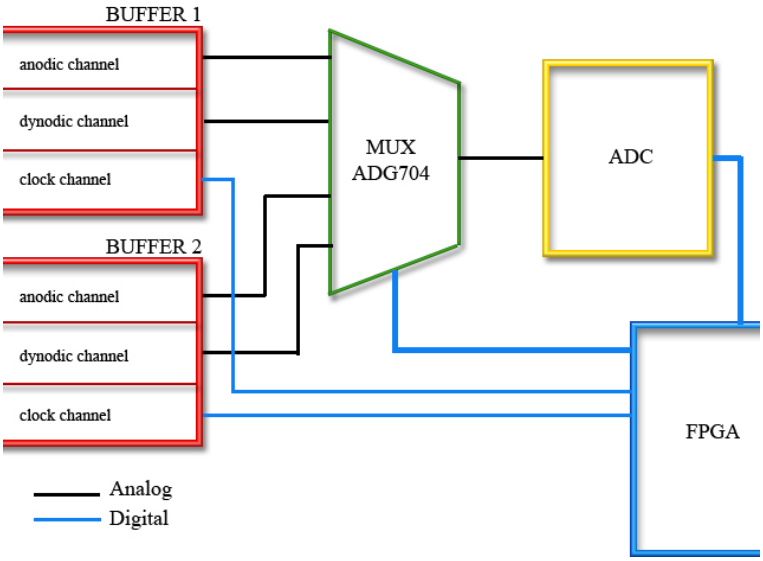

Figure 3.9: **Application scheme of the ADG704 analog multiplexer**.

Analog to digital converter

A key element of the acquisition chain is the data digitizer. The device chosen is a Burr Brown ADS901, a low-power pipelined converter with differential input and a 10-bit parallel output. The maximum working frequency is 20 MHz [20].

The pipeline of the ADC has 9 stages with each stage containing a two-bit quantizer and a two-bit digital-to-analog converter. Hence it follows that 5 clock cycles are required to have the first valid datum on the parallel output.

The main characteristics of this component are the low power dissipation ($<$ 45mW at 10MHz) and a good SINAD[4] of 49 db at 9 MHz.

A scheme of the ADC architecture is given in picture 3.10.

[4]Signal to (Noise+distortion)

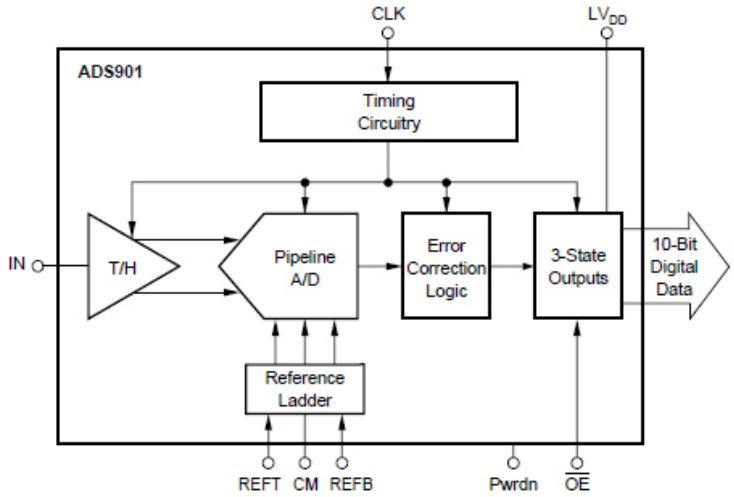

Figure 3.10: **The Burr Brown ADS901 ADC**. Pipelined data converter with differential input and power down capability.

3.3.3 The Field Programmable Gate Array

The main digital element of the board is the FPGA. This technology is diffusing more and more in the high-tech scenarios for its low cost per logic-cell and its great flexibility. This programmable device descended from the PAL[5] and the subsequent CPLD[6] families. Its most powerful characteristic is the possibility to re-arrange its logic elements in order to represent almost any possible digital architecture.

In the case of a microprocessor or a micro-controller the hardware architecture is static and it is designed to execute a well defined set of instructions. The programming code is then a sequence of these instructions. The FPGA instead, has no defined architecture but a set of logic elements which should be configured and interconnected at will to realize a well defined structure, that in principle could be a microprocessor as well. The firmware which is loaded inside an FPGA then, is not a sequence of instructions but a topological description of the configuration and interconnections of the basic logic elements.

These basic elements are called CLB[7] (Configurable Logic Blocks), mainly made up of a programmable combinatorial net called LUT[8]

[5]Programmable Logic Array
[6]Complex Programmable Logic Device
[7]using Xilinx terminology
[8]Look Up Table

and a relative array of registers[9]. Any sequential logic machine can be realized with the opportune interconnection of these two elements.

Depending on the manufacturer, the model and the dimension an FPGA can have a different number of CLB and each CLB can count on different slices (the minimum logic element). The FPGA mounted on this board is a Xilinx Spartan3E with 500K gates packed in a pq208 plastic package (device ID: XC3S500Epq208). A scheme of the CLB of a Spartan3E FPGA is proposed in Fig. 3.11.

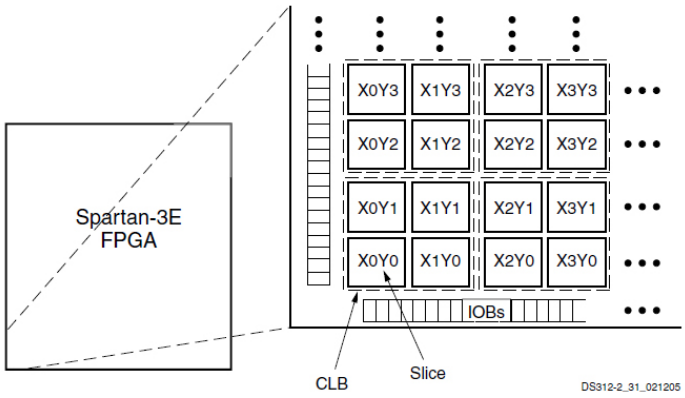

Figure 3.11: **Basic logic elements of a Spartan 3E FPGA**.

Each CLB is made up of 4 slices, and each slice is composed by 2 LUTs and 2 registers. Each LUT is equivalent to a digital multiplexer with a 4-bit selector, therefore it can implement any logical function $y = f(x_0, x_1, x_2, x_3)$.

Downloading the configuration file to the FPGA means that the 16 input line of the MUX have been tied to GND or Vdd defining so a specific logical function f. At runtime, "user signals" will be the four selectors of the MUX (x_0, x_1, x_2, x_3) and y will be the result of the combinatorial SOP operation feeding the output register of the LUT. In Fig. 3.12 the slice architecture is represented: the LUTs and the output registers have been described above, while the other components are auxiliary elements allowing the cascading of different slices for the realization of more complex logic.

A LUT can also be configured as a block of distributed memory,

[9]D-type Flip Flop

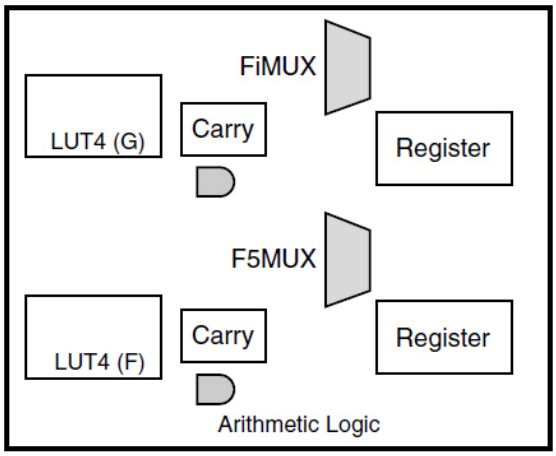

Figure 3.12: **Inside the slice of a Xilinx Spartan3e FPGA**.

where 16 bits of information are stored and which can be randomly accessed via the 4-bit selector. A set of distributed memory blocks is known as distributed RAM and it is suited for buffering small amount of data anywhere along signal paths.

An alternative to the distributed RAM is the Block RAM, which provides a more suitable solution in case of larger amount of data. This is a real RAM device embedded in silicon together with the programmable logic. The main features of this component are the dual port technology, which allows two independent accesses to the common block of RAM, and the configurable port aspect ratio, that allows different widths between the input and output data ports.

The device in use on board can count on more than 360 Kbits of block RAM, extensively used in the firmware architecture for the realization of FIFOs and shift registers.

Like the Block RAM, there are many other devices that are deployed in the silicon chip to support the mere programmable logic. For example, in the Spartan3E device that we used, it is present a dedicated clock infrastructure and 4 DCMs (*Digital Clock Managers*), which are appointed to frequency division/multiplication and phase adjustments.

As will be explained in the clocking section 3.3.4, the master clock provided by the FCM is multiplied by an external programmable PLL (*Phase Locked Loop*) to feed the LIRA chip which requires a low jitter input clock. This was thought because DCMs have a worse jitter performance if compared to analog PLLs.

The FCM master clock at 5 MHz is then processed by the FPGA internal DCMs to feed only the firmware logic. Being both the DCM and the external PLL locked on the same parent clock, the mix of analog and digital clock synthesizers has not been considered as an issue.

In the device in use there are also 20 multipliers located within the silicon die. They perform primarily two's complement numerical multiplication but can also perform some less obvious applications, such as simple data storage and barrel shifting. Each multiplier performs the principle operation $P = A \times B$, where A and B are 18-bit words in two's complement form, and P is the full-precision 36-bit product, also in two's complement form.

Optional registers can be added at the input and at the output stages of the multipliers, which can be used for storing data samples and coefficients or, when used for pipelining purposes, to boost the multiplier clock rate.

In our particular application these components are used for timing reconstruction of the samples extracted out of the LIRA chip as will be shown in the firmware section 3.3.7.

The chip has several I/O pads even if not all of them are available to the user application, for example the configuration interface has a set of dedicated pads and so have the power-supply rails as well. On the other hand, all the remainder user's pins are available to bring in and carry out any signal of the programmed logic. Another useful feature of the FPGAs that deserves to be pointed out is the possibility to assign different I/O standards to different groups of pin.

In the spartan 3E device the programmable I/O blocks are the elements between the core logic and the user I/O pins, a wide choice of digital standards is available (CMOS, LVTTL etc.). Some differential standards are also supported, for example LVDS and LVPECL, in this case I/O pairs are individuated on predefined couples of pins.

All the user input/output pins are grouped into four areas called I/O banks, each bank shares the same communication standard and the same V_{CCO} pin (see below). Each bank has a dedicated V_{CCO} which is used as a voltage reference for the programmed I/O standard on that bank. The allowed voltage references depend on the supported standards and in particular they are: 1.2V, 1.5V, 1.8V, 2.5V, 3.0V and 3.3V.

Even if our application foresees only one I/O standard (LVCMOS33 powered at 3.3V) we made possible to choose between 2.5V and 3.3V at each bank V_{CCO} pin using a jumper selector in order to leave some flexibility.

A list of the required power supplies is given:

- V_{CCINT}: 1.2V powering the low-voltage core logic (technology 90 nm).
- V_{CCAUX}: 2.5V auxiliary supply for the dedicated configuration pins, DCMs, differential drivers and JTAG interface.
- $V_{CCO0,1,2,3}$: the four dedicated supplies for the I/O output drivers.

A circuit to monitor these three power rails is required in order to grant the correct configuration of the FPGA at power-up and during its duty. For this reason there is a built-in POR (*Power On Reset*) circuit that validates the configuration of the device.

It has two main tasks, the former is to hold the integrated logic for the auto-configuration at power-up in a reset state until V_{CCINT}, V_{CCAUX} and V_{CCO_2}[10] supplies reach their nominal value. In this way configuration take place only when all the logic is correctly powered. The second task is to hard reset the FPGA if a power supply oscillation cross downwards one of the POR thresholds. In this case, the FPGA configuration file can get corrupted as it is retained in volatile memory leaving the device in a harming unknown state, hiding an imminent failure to the user.

To prevent the hardware from performing a wrong operation, a hard reset to the configuration logic is activated taking all the user I/O in a high impedance state. The reset is kept active until supplies become stable again, hence a new configuration cycle take place. With this behavior the probable fault is automatically restored and, in those situation where the failure is not much evident, it is detected more easily.

When the POR is released, the configuration logic of the FPGA can act differently on the base of the state of three dedicated pins: M0, M1 and M2. Depending on the code used, the FPGA can auto-configure itself downloading the bitstream from the configuration PROM or wait to be configured via the JTAG interface directly by the user (see 3.3.3). Several PROM programming modes are foreseen such as Master Serial, Slave Serial, Slave Parallel and Byte-wide Parallel. In our application the PROM is connected to the FPGA with a serial link and the only programming mode allowed is the master serial. In this case the PROM interface is clocked by the FPGA itself.

[10] Tension on I/O bank 2 is required as the ports that receive the configuration bit-stream share this supply with user's I/O.

The JTAG chain

The JTAG (*Joint Test Action Group*) or *Boundary Scan* protocol is a standard developed with debugging purposes. This communication interface foresees a daisy chain interconnection of devices, terminated on the master controller that accesses the slave's registers. In a nutshell, it is a simple serial protocol which allows to investigate chips behavior during their duty, exploiting very few control pins.

As it was thought for debugging purposes, the main controller is typically an external device that can be removed once the test phase is ended wasting no room on the PCB. JTAG interface is hence very widely used in factory's serial production lines for the final quality control.

Providing a very simple and almost costless way to access chip registers in read and also write mode, JTAG interface can be exploited for programming purposes as well. Once programmable devices should be placed physically into a programmer to be configured and then mounted; using JTAG interface this is no more necessary because it is possible to perform the so called *In-System Programming* (ISP). In other words, a programmable chip can be formerly soldered to the PCB and then programmed directly inside its environment.

The signal required to drive a JTAG chain are:

- TDI: the serial input line of a chip.
- TDO: the serial output line towards the next chip of the daisy chain.
- TMS: the state machine control signal.
- TCK: the clock of transmission.
- TRST: reset the TAP controller (see below). This is an optional signal.

TMS and TCK are connected in a star configuration to all the chips of the chain.

The working principle is quite simple, each chip must have a *Test Access Point* controller (TAP), a finite state machine which is the interface between the standard communication protocol and the specific registers of each chip. A graphical representation of the FSM is given in Fig. 3.13.

There are two types of registers associated with boundary scan. Each compliant device has one instruction register and two or more

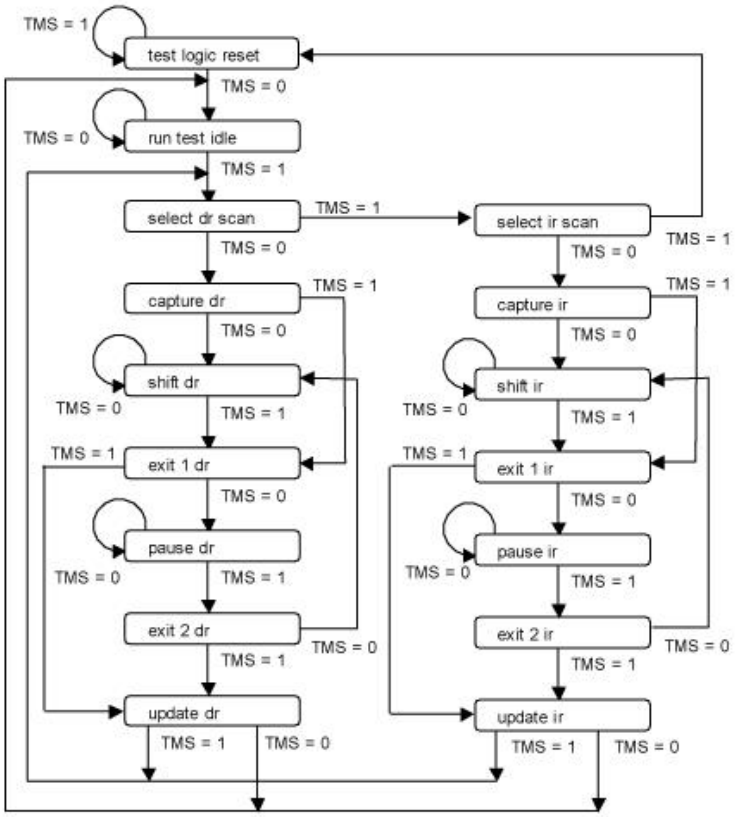

Figure 3.13: **Test Access Point controller finite state machine**.

data registers. The instruction register holds the current instruction. Its content is used by the TAP controller to decide what to do with the data that are received after. Most commonly, the content of the instruction register defines which of the data registers has to be accessed.

There are three primary data registers, the Boundary Scan Register (BSR), the BYPASS register and the IDCODES register. Other data registers may be present, but they are not required as part of the JTAG standard.

- BSR - this is the main testing data register. It is used to move data to and from the inside of a device.
- BYPASS - this is a single-bit register that passes information from TDI to TDO. It allows other devices in a circuit to be tested with minimal overhead.
- IDCODES - this register contains the ID code and revision num-

ber for the device. This information allows the device to be linked to its Boundary Scan Description Language (BSDL) file. The file contains details of the Boundary Scan configuration for the device.

In our specific application, the JTAG chain is used to program both the FPGA and the configuration PROM. This two devices are the elements included in the DAQ-board boundary scan chain as shown in Fig. 3.14

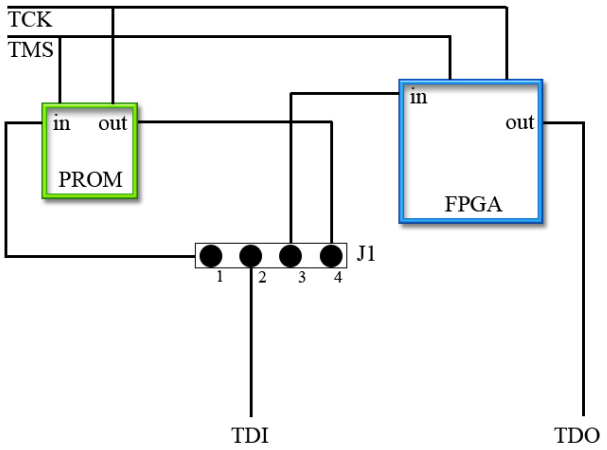

Figure 3.14: **JTAG chain on the NEMO DAQ-board for LIRA chip**.

The jumper J1 was introduced to include or exclude the PROM in the chain: if one jumper cap is positioned between 2-3, the FPGA only will be part of the chain, while a cap on 1-2 and one on 3-4 will close the loop over both devices. A connector on board allows to interface the Xilinx programming cable to the boundary scan.

Boundary scan has also been exploited on this board for its original purpose: a powerful tool for debugging. Xilinx realized a specific software, called ChipScope, which allocates some of the FPGA resources to integrate the equivalent of a logic state analyzer accessed by the JTAG programming cable.

Safe Dual Boot

In the design of the DAQ-board we included a secondary configuration PROM. This choice was taken to foresee firmware flexibility even when the board will be deployed with the tower. The FEM board of NEMO

Phase-1 uses a DSP during re-configuration, in our case this is not possible as we decided to use not this component. Hence the FPGA must perform itself the required actions for a remote re-configuration, and these are: receiving the new bit-stream file through the slow control interface, programming the auxiliary PROM via a dedicated JTAG port, point to the auxiliary PROM, and finally auto-reboot itself.

A second PROM was introduced for security reasons: if something fails during the remote re-programming of a unique PROM, there is no chance to make the FPGA come up again with a proper configuration. In this case communication with FCM would be unrecoverable, with the consequent loss of the whole optical module. In the solution that we adopted, called *Safe Dual Boot*, the primary PROM is never overwritten, leaving a chance to recover from a programming error.

A scheme of *Safe Dual Boot* is given in Fig. 3.15.

As previously said, FPGA is configured by the PROM in master serial mode, which means that the whole configuration file is shifted serially inside the gate array. A latched switch selector has been introduced on this serial line, with the reset state contact normally closed on the main PROM.

Once the firmware file is loaded into the secondary PROM, the FPGA connects to it switching the latched selector and then auto-reboot itself. In normal operating mode, during the FPGA reboot, the selector remains latched to the previously selected PROM, allowing to load the chosen firmware. If something fails, the connection with the OM is lost. In this case a power cycle of the DAQ-board is sufficient to force the reboot from the primary PROM, with an immediate reactivation of the link.

3.3.4 Clock distribution

All the electronics on the DAQ-board are synchronous on the same master clock provided by the FCM on a dedicated twisted-pair cable. A decoupling circuit, very similar to those used for the data lines, is included even if there is no DC component extraction on this line. The need of a complete galvanic decoupling between the DAQ-board and the FCM is due to safety reasons, we must remember that high voltages are presents in an optical module.

The clock is provided in *Low Voltage Differential Signaling* (LVDS), a standard that can cover shorter distances respect to the RS485 but which improves the sharpness of the signal edges. An LVDS receiver has been integrated to generate the CMOS signal to drive the FPGA

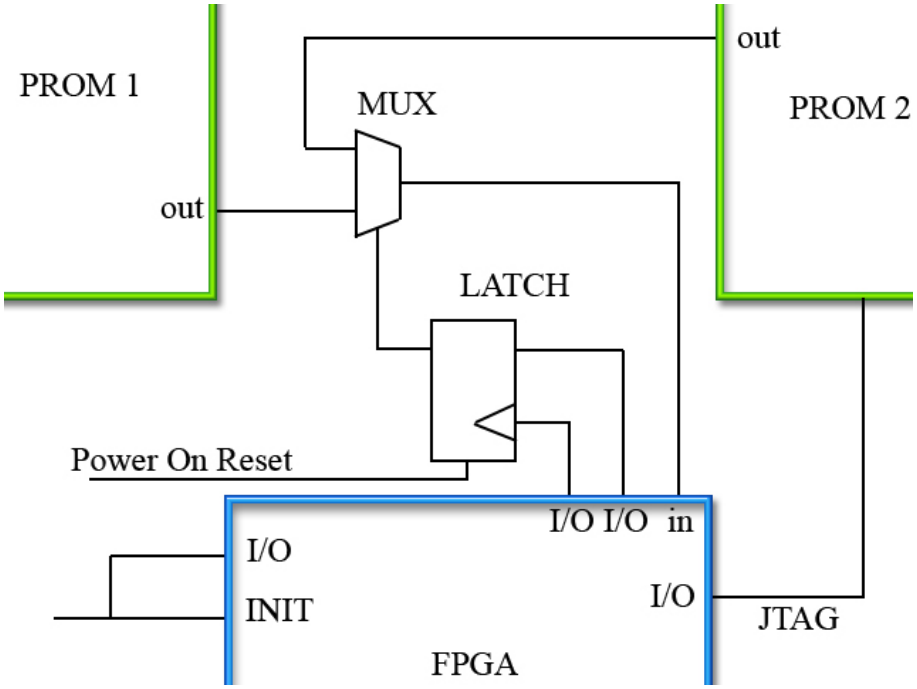

Figure 3.15: **Safe Dual Boot hardware architecture**. User's pin driven by firmware logic are indicated with *I/O*, *INIT* and *in* are pins reserved for configuration.

and the external PLL.

As was previously mentioned, the clock provided by the FCM has a frequency of 4.86 MHz while the chip LIRA requires a stable square wave 4 times faster. For this purpose an external PLL has been integrated on board. The device in use is a On Semiconductor NB3B502 with programmable frequency multiplier.

In the main working scheme the PLL is used to feed the chip LIRA only, while the FPGA exploits directly the 4.86 Master Clock. However, in order to leave some flexibility to the project, some other clock sources have been connected to the FPGA. The PLL multiplier output for example, and a 20 MHz crystal oscillator were brought to other clock pins of the FPGA foreseing a bench test phase without the FCM connection.

3.3.5 Debug features

The DAQ-board for the chip LIRA was realized to meet all the standard requirements for both communication and mechanical specifications. This meant to integrate all the analog and digital electronics in a very small area of about 11x12 cm^2. Therefore the debug elements foreseen in the project were mounted on two mezzanine cards to be removed once the test phase has ended. These cards housed also all the LEDs indicators, which are of no use under the sea and furthermore can cause damages if they accidentally turn on while the high voltage is present on the PMT.

One debug module includes a two digit seven segments display, a 20 MHz quartz oscillator and the DONE LED. This LED is driven by the configuration logic of the FPGA, and it is a visual indicator of the state of the gate array. When turned off it means FPGA is configuring or not configured at all, when it is on indicates that a configuration is present and ready.

On the second mezzanine there are the supply monitors: an array of LEDs where each one indicates the presence of tension on a specific plane both for digital and analog supplies (5V, 1.2V, 2.5V analog, 3.3V analog, 2.5V digital and 3.3V digital). In addition, two user push buttons and two switches were mounted and connected to the FPGA I/O.

An integrated circuit for temperature measurements has been mounted on this card together with a discrete bipolar junction transistor as sensor. This is a 2N3904 base-emitter junction with the collector tied to the base and it is driven and readout by a National Semiconductor LM83 temperature monitor. It basically measures the temperature-dependent current that flows in the diode-configured transistor and it interfaces to standard digital logic using an SMBus protocol.

These two mezzanine (Fig: 3.4) are both mounted on strip connectors. The former is leaning out of the bottom side while the one, equipped with temperature sensor, is right on the top of LIRA socket.

3.3.6 Transmission protocols

Before starting to describe the firmware implemented on the FPGA, as it is strictly related to the transmission protocols in use by the NEMO collaboration, a general explanation of them will be given.

The description of the physical link, consisting of a DC-extraction circuit and an RS485 transceiver, has already been given in the previous sections, now we will discuss in a bottom-up way the stack of protocols

involved in the data transmission.

The FCM-DAQ board transmission is based on a stack of protocols, as it happens in the TCP/IP protocol suite. Each layer of the stack has it own rules, giving determined services to the layer above by using the services of the layer underneath.

Fortunately, the FCM-DAQ protocol stack is much simpler than TCP/IP and it foresees only two layers. The lower layer, which means the one that deals directly with physical link, is a standard protocol that is in use also on Gigabit Ethernet transceivers, it is called 8b/10b. The upper layer, instead, is proprietary as it has been tailored on the needs of this particular kind of application. In the upper layer two different protocols co-exist, sharing the same layer-0 protocol and hence the same physical medium: the *Data Transmission Protocol* and the *Slow Control Protocol*. As previously explained, connection towards the front-end is meant only for slow control requests, so no Data Transmission Protocol is implemented on it.

The Layer 0 protocol: 8b/10b

In section 3.3.2 we described the DC power extraction circuit, which uses pulse transformers 1:1. Inductive transformers act as a high-pass filter with a certain cutoff frequency. A particular modulation must be introduced then to avoid the presence of DC and low frequency components, otherwise the original signal would be completely distorted.

For this purpose a digital modulation protocol called 8b/10b has been adopted. It brings mainly two significant advantages: it transmits a DC-free signal and it introduces redundancy, that is used for error detection and transmission of special control characters.

The 8b/10b coding scheme was initially proposed by Albert X. Widmer and Peter A. Franaszek of IBM Corporation in 1983 [47]. The encoder, on the transmitter side, maps the 8-bit parallel input to a 10-bit output. This 10-bit output is then shifted out through a high-speed serializer (Parallel-in Serial-out 10-bit Shift Register). The serial data stream is transmitted through the transmission media to the receiver. The high-speed deserializer (Serial-in Parallel-out 10-bit Shift Register) on the receiver side converts the received serial data stream from serial to parallel. The decoder then re-maps the 10-bit data back to the original 8-bit form (see Fig. 3.16).

When the 8b/10b coding scheme is used, the serial data stream is DC-balanced and has a maximum *Run Length* of 5. DC-balanced means that it keeps almost the same number of 0's and 1's on the

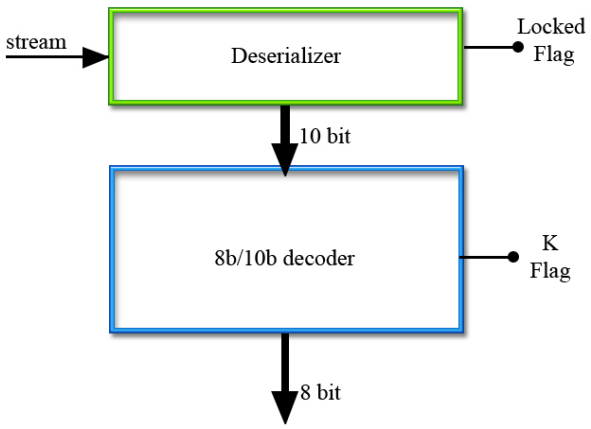

Figure 3.16: **Deserializer and decoder 8b10b**. The decoder transforms the 10-bit parallel input into an 8-bit parallel output plus a K flag which individuates the command codes.

stream while the Run Length is defined as the maximum number of contiguous 0's or 1's.

The 8b/10b encoder converts 8-bit codes into 10-bit codes. The encoded symbols include 256 data characters, named Dx.y, and 12 control characters named Kx.y.

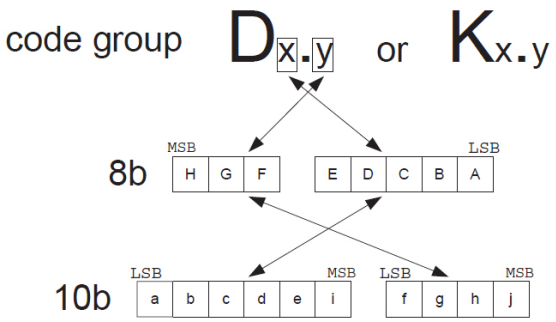

Figure 3.17: **Conversion scheme in the 8b/10b modulation algorithm**.

The coding scheme breaks the original 8-bit data into two blocks, 3 most significant bits (y) and 5 least significant bits (x). From the most significant bit to the least significant bit, they are named as H, G, F

3b Decimal	3b Binary (HGF)	4b Binary (fghi)
0	000	0100 or 1011
1	001	1001
2	010	0101
3	011	0011 or 1100
4	100	0010 or 1101
5	101	1010
6	110	0110
7	111	0001 or 1110 or 1000 or 0111

Table 3.1: **3-bit to 4-bit encoding values**

and E, D, C, B, A. The 3-bit block is encoded into 4 bits named j, h, g, f. The 5-bit block is encoded into 6 bits named i, e, d, c, b, a as shown in Fig. 3.17. The 4-bit and 6-bit blocks, also called sub-codes, are then recombined into a 10-bit encoded value.

In order to create a DC-balanced data stream, the concept of block *Disparity* is used to balance the number of 0's and 1's. The disparity of a block is calculated by the number of 1's minus the number of 0's. The value of a block that has a zero disparity is called disparity neutral.

If both the 4-bit and the 6-bit blocks are disparity neutral, a combined 10-bit encoded data will be disparity neutral as well. This would create a perfect DC-balanced code. However, this is not possible because only 6 out of the 16 possible values of the 4-bit block are disparity neutral and they are not enough to encode the 8 values of the 3-bit block. Likewise, only 20 values of the 6-bit block are disparity neutral and they are not enough to encode the 32 values of the 5-bit block. Having both the 4-bit and 6-bit blocks an even number of bits, the disparity is not possible to be +1 or -1, therefore values with a disparity of +2 and -2 are also used in the 8b/10b coding scheme.

In Tab. 3.1 and Tab. 3.2 are shown respectively the conversions of the 3-bit and the 4-bit blocks into the 5-bit and 6-bit sub-codes, note that some block has two or more different representations.

Concatenating the 4-bit and 6-bit blocks together generates the 10-bit encoded value. The 8b/10b coding scheme was designed to combine the values of the 4-bit and 6-bit blocks so that the worst case disparity value of the 10-bit code can be at most +2 or -2. For example, the 4-bit encoded values with disparity value +2 is not combined with the 6-bit encoded values with disparity value +2 because this would generate a code with disparity +4.

5b Decimal	**5b Binary (EDCBA)**	**6b Binary (abcdei)**
0	00000	100111 or 011000
1	00001	011101 or 100010
2	00010	101101 or 010010
3	00011	110001
4	00100	110101 or 001010
5	00101	101001
6	00110	011001
7	00111	111000 or 000111
8	01000	111001 or 000110
9	01001	100101
10	01010	010101
11	01011	110100
12	01100	001101
13	01101	101100
14	01110	011100
15	01111	010111 or 101000
16	10000	011011 or 100100
17	10001	100011
18	10010	010011
19	10011	110010
20	10100	001011
21	10101	101010
22	10110	011010
23	10111	111010 or 000101
24	11000	110011 or 001100
25	11001	100110
26	11010	010110
27	11011	110110 or 001001
28	11100	001110
29	11101	101110 or 010001
30	11110	011110 or 100001
31	11111	101011 or 010100

Table 3.2: **5-bit to 6-bit encoding values**

We introduce now the concept of *Running Disparity* (RD) which is the sum of disparities of all the previous characters sent. It is tracked during transmission in order to keep balanced the line. Having the characters a maximum modulus of disparity equal to 2, the RD will be confined between two values, at most distant 2, if a wise sequence of the encoded words is generated.

In order to do that, every byte is associated to two 10-bit character encodings: the RD+ and the RD- representation. Encodings with disparity either +2 or 0 (disparity neutral) belongs to the RD- group while the RD+ values are the encodings with a disparity either -2 or 0. When the Running Disparity is maximum the RD+ encoding of a byte will be chosen, having a negative or null disparity it will keep the RD confined in the allowed range.

By convention, the transmitter assumes a Running Disparity equal to -1 at start up. When the first byte arrives, the encoder will then use the RD- encoding (disparity +2 or neutral) trying to balance the line. If the datum encoded is disparity neutral, the Running Disparity would not be changed and, for the next byte, the RD- encoding would still be used. Otherwise, the Running Disparity changes into +1, and next byte is encoded RD+. Similarly, if the current Running Disparity is positive (RD+), and a 0 disparity word is encoded, the Running Disparity will still be RD+. Otherwise, it would be changed from RD+ back to RD- and the RD- value would be chosen again.

Fig. 3.18 shows all the possible evolution paths of disparity during the transmission of a character. We can see that at the beginning of the code disparity is +/-1 as previously mentioned, and that the maximum number of successive identical bits (Run Length) is 5. The dynamic evolution of disparity, for every bit transmitted on the serial line, is called *Digital Sum Variation* (DSV), and in the particular case of the 8b/10b code it is limited between +/-3.

In addition to the RD+ and RD- encodings for the 256 bytes (the D-codes), there are also 12 RD+ and 12 RD- codes that still allow to respect a run length of 5 and a DSV = +/-3 on the data stream. These are the *Control Characters* used for the implementation of line services, we usually refer to them as *K-codes*. The RD+ and RD- representations of the K-codes are shown in Tab. 3.3.

When a control character is received, the decoder informs the upper protocol that the decoded value is a control code and not a data byte.

These codes are useful during the link negotiation phase, for data-packets delimitation and, in our specific case, they are employed to

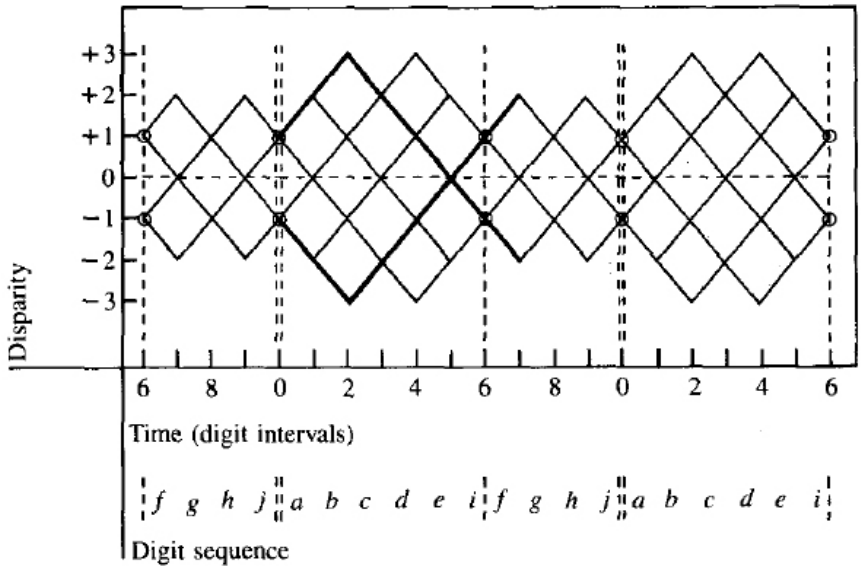

Figure 3.18: **Evolution diagram of the Digital Sum Variation for 8b/10b encoding**. Double dotted lines indicate the start of a character. As foreseen the disparity at the beginning of a coding can be +1 or -1. The diagram represents the evolution bit after bit of the disparity with any possible character: 1s increment disparity, while 0s decrement it.

transmit also real-time commands. When the transmitter is powered up, the transmitter starts to send a sequence of *comma characters* (K28.5) and the receiver tries to lock on it. The sequence of successive K28.5 codes in the RD+ and RD- form, allows to univocally delimitate the beginning of a character as no possible overlapping subset correspond to a valid symbol.

The k-codes used as packet delimiters, depend to which protocol the payload belongs, so they will be described in the relative protocol subsection.

The only real-time command foreseen up to now is the TCR (*Time Counter Reset*) control character K28.2 sent to the front-end. When a TCR request is received, in the backward direction is transmitted the relative acknowledgment character (§ 3.3.7).

All the received invalid characters generate an error-flag on the decoder. Redundancy is then exploited for the detection of errors which can generate invalid codes or violations in the Running Disparity. In case of a RD violation, the error occurred not necessarily in the last

	8bit code	RD-	RD+
K28.0	000 11100	001111 0100	110000 1011
K28.1	001 11100	001111 1001	110000 0110
K28.2	010 11100	001111 0101	110000 1010
K28.3	011 11100	001111 0011	110000 1100
K28.4	100 11100	001111 0010	110000 1101
K28.5	101 11100	001111 1010	110000 0101
K28.6	110 11100	001111 0110	110000 1001
K28.7	111 11100	001111 1000	110000 0111
K23.7	111 10111	111010 1000	000101 0111
K27.7	111 11011	110110 1000	001001 0111
K29.7	111 11101	101110 1000	010001 0111
K30.7	111 11110	011110 1000	100001 0111

Table 3.3: **Control characters coding**

character received. Therefore the standard impose to consider invalid all the last three characters received. The upper protocol is informed of such errors by dedicate flags, in case, it must be programmed to take the appropriate measures.

The Data Transmission Protocol

This is one of the two application protocols used in this layer and it is meant to transmit in a compact way all the information relative to a recorded event. Two kinds of information have to be transmitted onshore:

- The timing information relative to the first sample of the event. It must individuate univocally a precise time in a period of 500 μs with a precision of 5 ns.
- The digital information of the recorded waveform, as it is sampled with an 8-bit dynamic, this will be the width of a sample field.

Each data packet is delimited within a BDP *Begin Data Character* and an EDC *End Data Character*, respectively K28.0/k28.6 and K28.1. The dimension of the payload is arbitrary but it must contain at least the time stamp and one sample.

The first field of the payload is reserved to the time stamp. It is 16-bit wide and in big-endian notation (MSB first). How the required timing resolution (5 ns for 500μs) is achieved will be explained in the firmware subsection.

	DATA			
layer	8 bit	16 bit	N x 8 bit	8 bit
L1	-	time stamp	samples	-
L0	K28.0 or K28.6	payload		K28.1

Table 3.4: **Data Protocol enclosure within the 8b10b delimiters**.

After the time stamp field, all the 8-bit data characters received before the arrive of an EDC are interpreted as waveform samples, transmitted in temporal order from the first to the last of the event. The heading time stamp refers to the sample that immediately follows it.

The Slow Control Protocol

In order to send Slow Control commands and answers, a dedicated protocol has been implemented in both directions. Every SC-packet starts with a BSC (*Begin Slow control Character*) (K28.3) but, differently form the data protocol, no closure of the packet is required. Slow Control, in the original architecture, was managed by a 24-bit DSP, which determined the length of a standard SC word. As this length is now fixed by standard, the minimum amount of SC information that can be exchanged is a 24-bit word. Longer instructions or replies are always multiple of this quantity.

A SC packet can be made up of a variable number of words, each of them headed by a BSC character. The first word of every packet, called *header*, brings information on the length of the packet itself. There are two kinds of packet

- Single Word Packet (SWP): consisting of one word only: the header.
- Multiple Word Packet (MWP): consisting of a variable number of words, from a minimum of 1 to a maximum of 1+65535.

The structure of the header is explained in Tab. 3.5.

MWPB (*Multiple Word Packet Bit*) If 1 indicates that the packet is composed of more than one word. If 0 means that the header is the only word of the packet.

RSNB (*Response Packet Bit*) If 1 means that the packet has been generated as a reply to a request or as an acknowledgment to a command.

	Bit #	Field Name	Field Length
MSb	[23]	MWPB	1 bit
	[22]	RSNB	1 bit
	[21:16]	CMDCODE	6 bit
LSb	[15:0]	DATA_LENGTH	16 bit

Table 3.5: **Header structure of a SC packet**

CMDCODE (*Command Code*) It is the code that individuates a command to be executed or a request.

DATA_LENGTH This couple of bytes has different meanings depending on the value of the MWPB:

- MWPB = 1: DATA_LENGTH is the unsigned int representation of the number of words that follow the header.
- MWPB = 0: DATA_LENGTH is the data field of the specific command, request or reply individuated in the CMDCODE field.

There are two kind of slow control communications between the PMT and the FCM, in one case the FCM makes a request or give a command to the front-end electronics, otherwise the DAQ-board spontaneously send some environmental information to shore.

In the first case it is foreseen, even when it is not necessary, that the front-end replies at least with a single-word acknowledge packet. This will have RSNB = 1 and the CMDCODE relative to the command received. For example, if a threshold set command is sent to the OM, no reply is required in principle, but an acknowledgment must be sent anyway in order to check the reception of the command.

An ulterior check is made on the SC communication using a redundant word added at the end of each SC packet. In case of a single-word packet the check word is simply the repetition of an identical single-word packet. On the other hand, if a multiple-word packet has been sent, the redundancy word is the bitwise XOR of all the previous words of the packet.

As the SC protocol shares the same transmission medium with the data protocol, a maximum amount of the total bandwidth (16 effective Mbit/s) has been assigned to the slow control. In normal conditions it is foreseen that complex Slow Control operations are performed only when the acquisition is turned off, but in any case, to refrain the SC from stealing too much bandwidth to data transmission, it is imposed that only one word of SC can be sent in a time window of 125 μs.

3.3.7 Firmware architecture

In this section is described the firmware running on the FPGA. The main tasks of the digital logic are managing the acquisition process and providing connectivity towards and from the floor concentrator.

The whole firmware project has been developed using VHDL code, a hardware description language. This kind of language could seem similar to a programming code at first sight but the basic concept is though very different. VHDL is meant to describe parallel logical operations to be synthesized into a hardware port netlist, while a programming code, like C, generates sequence of instructions for a specific hardware.

The software environment used for the firmware development is the Xilinx ISE, a tool provided by the same FPGA manufacturer. This environment includes mainly a source code manger, the code editor and a graphical interface to the implementation programs.

When the synthesis take place, formerly the high level language is compiled and transposed into a netlist of logical ports which perform the required operations. The synthesized netlist is then used to implement the design into a specific target device, in our case a Spartan 3E FPGA. The implementation phase is subdivided into three main parts: *map*, *place and route*, and generation of the *configuration file*.

Mapping basically makes a 1 to 1 correspondence from the gates required by the netlist and the physical resources of the device (LUTS and registers). Afterwards these elements have to be placed in a convenient way in order to ease the interconnection between them and to optimize path delays; this is what happen during the place and route phase. The last step in the implementation procedure is the condensation of the programming information into a binary configuration file, which is the sequence of bits that will be physically shifted into the configuration memory of the device.

The reference image that schematize the whole firmware is presented in Fig. 3.19 in order to help the reader during the description of the several blocks.

Control Unit and Time counter

The analog sampling is performed by the LIRA chip which however is not provided of a control logic and hence it can not operate in a standalone configuration. The programmable logic of the FPGA supports the LIRA chip providing the required CU (*Control Unit*). It is meant to drive the two analog buffer instances and to implement the DPTU

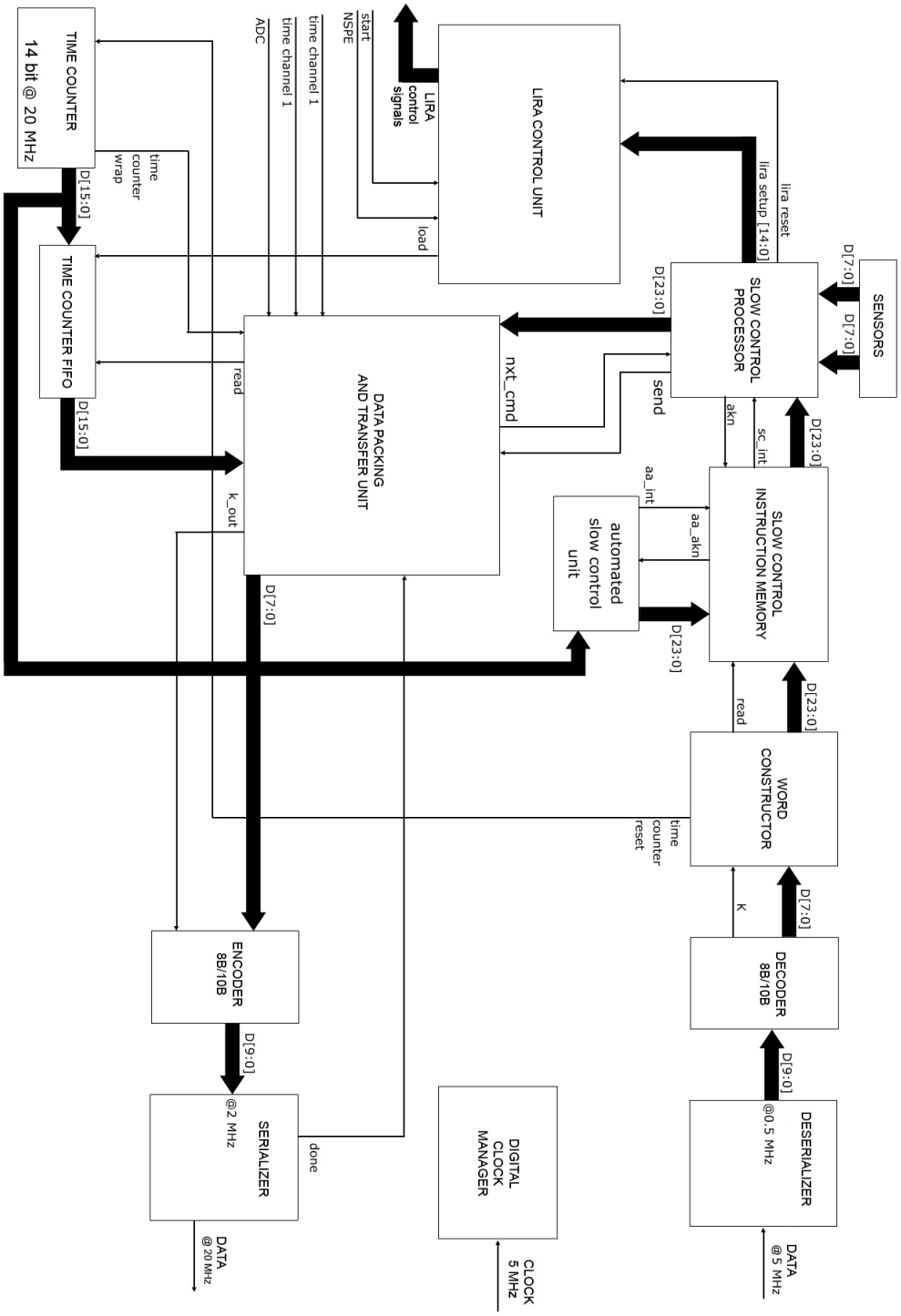

Figure 3.19: **Firmware architecture scheme**.

(Data Packing and Transfer Unit), the component that organizes raw data samples into standard data packets.

As shown in the previous Fig. 3.5 and in Fig. 3.19, the control unit receives the triggers and the classification information about the incoming pulse from the T&SPC. The 80 ns delay introduced at the sampler input allows the T&SPC to classify the signal, and the CU to start the acquisition before the arrive of the first over-threshold value at the LIRA's input.

When a pulse is detected to be over threshold by the T&SPC, it sends a start signal to the control logic accompanied with a NSPE (Non Single Photo Electron) information bit. If this bit is set to a logical 0 it means that a SPE triggered the acquisition and in this case, by convention, 10 samples are shifted in. If the classification bit is set to 1, instead 100 samples are taken, because a complex and longer waveform could have generated the trigger.

After the arrival of a trigger, several operations are done by the real time CU:

- It starts the analog acquisition by enabling the sampling clock on the front-end chip and it takes the right amount of samples on the base of the classification information.

- It stores the classification bits. During the readout operation they will discriminate the kind of events stored inside the analog memory. Without this information it would not be possible to delimitate the sets of samples belonging to different events while they come out of the analog buffer.

- It stores the value of a cyclic counter running at 20 MHz and wrapping every 500 μs inside the *Time Counter FIFO* as a coarse time stamp relative to the event. The time counter will be described later on but basically it acts as a stopwatch taking partial timings.

As was mentioned, the LIRA chip contains two independent analog memories with three channels each. They should work in buffer mode which means that, in ideal conditions, when one is reading the other should be writing or, at least, ready to write. The depth of the analog FIFOs is 250 samples, which correspond to 25 SPE.

The CU keeps track of the analog memory occupation, and set high the called *almost full flag* when a programmable threshold of the occupancy is crossed. This flag is meant to indicate that the analog FIFO can go full on next event and, in this situation, the role of the

two samplers must be switched. The one that gone *almost-full* has to be read and the other sampler must be put in writing mode if possible. In a very conservative way, the almost full threshold was set to 150, foreseeing a successive NSPE event.

When the almost-full flag of a buffer is activated, it is put in read-mode as soon as possible. During this phase no writing operations are allowed and the analog samples are shifted out at 10 MHz rate.

In each of the two LIRA's analog buffers the three channels work synchronously in parallel, so every sample is always temporally aligned to the corresponding other two but, on the other hand, they are also shifted out all together. Though, the use of an analog multiplexer made possible to use a single ADC to perform data conversion. See Fig. 3.9 and refer to section 3.3.2 for the interconnections of ADC to the MUX.

The control logic knows in advantage which sample of the event, and which event is coming out of the LIRA buffer because the classification of the T&SPC was previously stored. Then, during a reading operation in case of a SPE event, the FPGA will connect the ADC input to the anodic output of the buffer. The MUX will be held in this position for a duration of 10 samples, that at 10 MHz rate it means $10 \times 100\ ns = 1\ \mu s$. Otherwise in case of a NSPE it will link the dynodic channel output for a duration of 100 samples, that is $100 \times 100\ ns = 10\ \mu s$. All the samples are store inside the *Sample FIFO* whose width reflects the resolution of the ADC which is 10-bit wide. This memory, like all the other in use, is realized with dual port RAM, allowing the push and pop operations to be completely independent. In this case the storing rate is 10 MHz, but the reading is faster as it is interfaced to the data processing unit of the DPTU.

The logic levels of the clock channel are stored as well in a dedicated memory called *Time Channel FIFO*. This is a 1-bit width FIFO filled with 0s or 1s depending on the logic level that come out of the analog buffer.

Once the event samples have been extracted and digitized, the FPGA still keep track of the classification of those events, because it will be necessary once again during the building of a standard data packet. The memory that contains classification bits to be used during data packing is called *Classification FIFO*.

To summarize, the Control Unit is appointed to the switching of the two buffers (from read to write mode and vice versa) and to the filling of all the FIFO mentioned above.

Now let us have a closer look at the time counter. It is made up of a binary counter synchronous on the 20 MHz clock, which is periodically reset by the FCM in order to control the synchronization, exploiting a

real-time command of the 8b/10b protocol. The period of the counter is 500 μs which means counting from 0 to 9999 at the nominal rate (20 MHz).

A control logic is also integrated in it to drive the input of the adjacent *Time Counter FIFO*. The value of the counter is pushed into this FIFO if a trigger arrives. In this way we collect all the partial timings of the events without stopping the counter. Furthermore, as it is foreseen by the communication protocol, an answer to each TCR (*Time Counter Reset*) request must be sent back to the FCM. The sequence of these answers must be perfectly interposed between the event time stamps due to time allegement requirements. For this reason the TCR answer words are encoded inside the Time Counter FIFO as well.

The TCR answer codes are stored inside the 16-bit wide TC-FIFO exploiting a redundancy of two bits. Every valid counter value is between 0 and 9999, whose binary representation requires only 14 bits. The remainder two bits are both 0 if the value stored is actually an event time stamp, otherwise the other 3 combinations are used to code three possible TCR answer codes.

The kind of the answer depends on the value of the counter at the moment of a TCR arrival. Three possible situations can be figured out:

- TCR arrives and TC = 9999: Time counter kept synchronism, reply with a TCR-OK code (encoded at the output port with a K28.7 K-code).
- TCR arrives and TC $\neq$ 9999: Time counter lost synchronism, reply with a TCR-BAD code (encoded at the output port with a K23.7 K-code).
- TC reach 500 μs but but no TCR arrived yet: Something is wrong, reply with a NO-TCR code (encoded at the output port with a K27.7 K-code). In this case the counter auto-resets itself.

The precise ordinal succession of events and TCR answers is of vital importance to keep synchronous the whole apparatus. The FCM receives the data packets time-labeled within a window of 500-μs. Since that is the longest period measurable on a OM, no other temporal information can be included in a data packet. It follows that the arrival on the FCM of two data packets, belonging to different time windows, must be interleaved with the right amount of TCR answers. Each TCR answer is considered then as a counter carry signal, and it is used to increment a wider counter residing on the FCM which has a period of

about a month. The final event structure, that is received on shore, is labeled with this absolute time counter.

Slow Control Processor

Once the Slow-Control requests have been received from the FCM and saved into the *SC-Word Memory*, they are processed by the *Slow Control Processor*. This components has been realized with a *PicoBlaze* 8-bit processor, a pre-synthesized core realized by Xilinix [11]. This is a versatile and cheap core, in terms of logic elements cost, which suited well our application. Another useful characteristic of this component is the constat instruction execution time. It means that every coded instruction always takes the same time to be executed, and this time is 2 clock cycles for any of the instructions.

A scheme of the blocks that form the *PicoBlaze* processor is presented in Fig. 3.20.

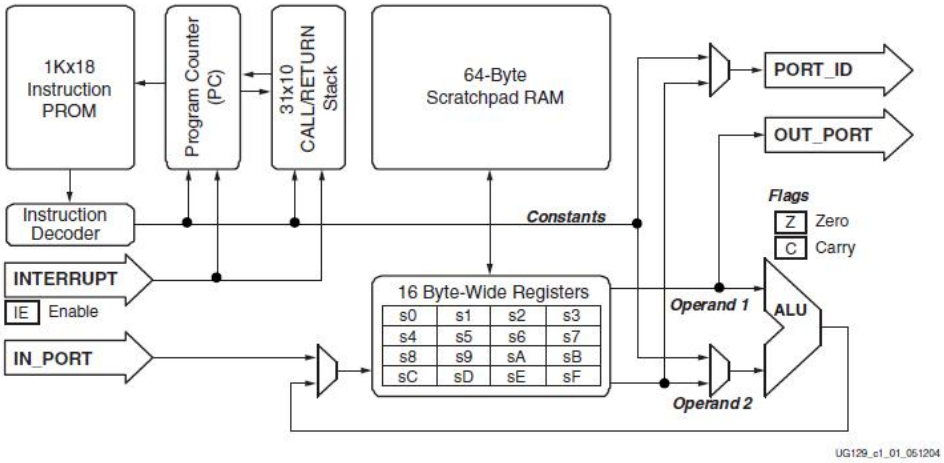

Figure 3.20: **PicoBlaze architecture**. Simple 8-bit RISC soft-processor implemented with the logic elements of the FPGA provided by Xilinx.

The post-synthesis VHDL code of the core (furnished by Xilinx) must be instantiated inside the project and connected to the *Instruction PROM component*. This is basically a Block-RAM instantiation initialized with the processor application program. To help the user

[11] http://www.xilinx.com/ipcenter/processor_central/picoblaze/picoblaze_user_resources.htm

in the generation of the machine-code, an application is provided by Xilinx that converts "human" assembler language into machine-code. This sort of assembly compiler directly generates the VHDL code of the pre-initialized Block-RAM.

In our particular application this processor is used to execute the incoming Slow Control requests, to interface to the environmental sensors and to compose the Slow Control answers that must be sent back to shore. It is clocked by a 50 Mhz clock, synthesized by a DCM locked on the main clock. In this way we increased the speed of command execution since 2 clock cycles are required to perform an instruction; this implies that one operation is executed in 40 ns.

A dedicated FIFO is implemented inside the DPTU component (describer hereafter) to collect the SC answers, which there wait to be sent back. The bandwidth allocation of the two protocols is a task of the DPTU.

Data Packing and Transfer Unit

This is the unit responsible for event construction, data-protocol implementation and Slow Control bandwidth limitation. It also implements some algorithms that made possible to adapt the data produced with our different front-end architecture on the existing data communication standard, that was tailored on different needs.

The most challenging issue was the reconstruction of a standard time stamp. We want to recall that the timing information required for a sample needs a precision of 5 ns over a period of 5μs, which is the period of the coarse time counter on the FCM.

In the NEMO Phase-1 architecture, fine timing information is provided by a 16-bit time counter running at 100 MHz, which can cover the whole 5 μs period with a precision of only 10 ns. This value is the time stamp associated to the couple of samples taken on the rising and falling edges of the 100 MHz clock. During the building of events, samples are put in pairs with the first one always referring to the rising edge. Therefore, a 10-ns resolution for the pair implies a 5 ns timing accuracy on each sample.

In our case, the first sample of the event structure is the first sample over threshold, where the comparison is taken at 200 MHz. In the Phase-1 events, instead, the first over-threshold sample can be the first or the second in the sequence, depending if threshold was crossed on the rising or on falling-edge sample.

A hypothetical full resolution time stamp, counting for over 500 μs with a precision of 5 ns (200 MHz) must have at least 17 bits.

In the Phase-1 architecture, based on the staggered ADCs (where the first sample of an event always refers to the rising edge of the 100 MHz clock) we would have the *lsb* (least significant bit) of the full resolution counter always set to 0, making it a useless information. That is why the present data protocol foresees 16 bits of time stamp only, corresponding to the 16 most-significant-bits of the full resolution counter.

In our architecture it is all quite different, we retrieve the full-resolution timing information combining the 14 bits of a coarse time counter running in the FPGA at 20 MHz, with a fine information which is provided by the chip LIRA.

The issues then, were to reconstruct the full-resolution time label and to fit it into 16 bits only, not enough for the desired resolution:

- 16 bits at 100 MHz : 10 ns precision for 656.350 μs. ENOUGH
- 16 bits at 200 MHz : 5 ns precision for 327.675 μs. NOT ENOUGH

We needed then to deal with two main issues: first of all the recovery of a unique time stamp from the coarse and the fine information; and secondly, resolving the lack of a bit of precision in the time stamp field of the data protocol.

The former problem is solved with a particular algorithm (see later on), and the solution to the latter could be the sequent: formerly find out if the first sample of an event-set corresponds to a rising-edge of the equivalent 100 MHz clock then:

- YES - no problem arise, as it would be a situation identical to the only one allowed by the standard protocol.
- NO - a possible way to comply with the standard would be the rejection of the first sample or the introduction of a fake one.

Anyway this solution presents some difficulties and compromises that are not actually necessary. A smarter use of the 8b/10b low level protocol could avoid the loss of samples or the introduction of fake ones. The bit of information that lacks in the 16-bit time stamp field of the data protocol has been incorporated in the 8b/10b data packet header. A different *Begin Data Packet* K-code is used in case the lsb of the full resolution counter is `1`.

We came to an agreement with the collaboration in order to integrate this additional feature in the data protocol at the level of the FCM. The standard BPD is the K28.0 character, corresponding to a lsb = `0` while the new BDP introduced is the K28.6 for a lsb =`1`.

HEX pattern	BIN pattern	STUs to be added
0x01F	00000 11111	0
0x20F	10000 01111	1
0x307	11000 00111	2
0x383	11100 00011	3
0x3C1	11110 00001	4
0x3E0	11111 00000	5
0x1F0	01111 10000	6
0x0F8	00111 11000	7
0x07C	00011 11100	8
0x03E	00001 11110	9

Table 3.6: **Possible patterns for the Time Channel FIFO**. These are the 10 samples of one 20 MHz clock period. The rightmost bit corresponds to the first sample acquired. To each pattern corresponds a precise number of STUs that has to be added to the coarse counter.

Now we will show in details the algorithm used for the reconstruction of a quasi-standard 17-bit time stamp. The goal is to realize a fast algorithm that can make a conversion into the standard form.

We dispose of of a 20 MHz square wave period sampled at 200 MS/s. For simplicity we will consider only the retrieving of time stamp for an SPE event. In this case 10 samples of the anodic pulse are taken together with 10 samples of the square wave. The possible patterns that can be found in the *Time Channel FIFO* are represented in table 3.6.

Let us call STU (*Standard Time Unit*) a 5-*ns* time interval, being the period of a 200 MHz oscillation. The coarse time stamp, counting at 20 MHz, is then a counter of STU's tens. The first step is then to multiply by 10 this value, afterwards the number of STUs to be added will be established by the time channel pattern. Each pattern individuates univocally a precise number of STUs to be added to the "coarse" time stamp.

This algorithm has been implemented, together with the event building, on a second instance of a *PicoBlaze* processor, at the base of all the DPTU processes.

Since the processor doesn't have an integrated multiplier, we opted for the interconnection of a dedicated *18 ×18 Multiplier* (§ 3.3.3) that is provided within the Sparta3E FPGA. Software multiplication algorithms could be implemented as well but they would be unnecessarily time-consuming operations.

An auxiliary look up table has been configured like Tab. 3.6 to be consulted by the processor, in this way in one clock cycle it obtains the correct number of STUs that need to be added.

Once the time stamp has been reconstructed, the DPTU formats the data packet as described in 3.3.6 into the *Formatted Out FIFO*, including also the BDP and the EDP control characters at the beginning and at the end.

When the DPTU processor is in stand-by, waiting to build up an event, it polls continuously the Time Counter FIFO until something is available in it. This 16-bit wide FIFO contains coarse time stamps (14 bit) and TCR acknowledge codes. If a TCR acknowledge code is found, the DPTU pushes in the Formatted Out FIFO the corresponding K-code and it returns to poll the TC-FIFO. If a time stamp is found, the DPTU starts to reconstruct the full resolution time stamp and the event building takes place.

In the *Slow Control Protocol* section we mentioned the bandwidth quota that is allocated for this protocol, the DPTU performs also the task to mix the two high level protocols into the outgoing data stream limiting the bandwidth of the SC answers. For this purpose a counter is interfaced to the *PicoBlaze* CPU counting the 125 μs that must be interleaved between each SC word. The SC words to be sent, are made available in a FIFO by the *Slow Control Processor*. These words are ready to be included in a slow control packet, the DPTU put them in the *Formatted Out FIFO* with a heading K28.3 control character, which is the chosen Begin Slow-control Character.

The outermost element of the DPTU, which interface directly to the 8b/10b encoder, is called *Out Manager* and it is basically the interface of the Formatted Out FIFO towards the encoder. It is a flow controller meant to keep the outgoing stream running at the fixed constant rate of 2 MByte/s. When no data, nor slow control words nor TCR acknowledge are present in the Formatted Out FIFO, the Out Manager transmits the so called idle character (K28.5). This control character is extremely important in order to keep up the link while the line is not in use, since it is appositely meant to DC-balance the line.

The whole firmware project has been realized trying to optimize the speed performance. The event building and data packing operations are critical as they must work without introducing another bottleneck in the acquisition process. The benchmarks reported in section 3.4.3 demonstrate that we realized a DPTU capable to allocate all the bandwidth available for data transmission, leaving the up-link the only bottleneck of data-acquisition.

Communication interface

All the SC requests/commands that arrives on the front-end are stored into the *Slow Control Instruction Memory*. The only exception is the *Time Counter Reset* request, which is interpreted by dedicated logic right after the 8b/10b decoder. A rigid real-time request cannot be handled by a CPU. The *Slow Control Instruction Memory* is 8-bit wide, and contains the slow control commands in big-endian notation so that the first byte received is the MSB of the 24-bit SC word. The SC memory is filled also with the requests generated by the *Automated Slow Control Unit*. This entity is meant to perform some routine operations like the expedition of sensor's values.

The automated slow control requests are executed like all the others, but they generate in the *Slow Control Response FIFO* a header word with RSNB = 0. The automated SC operations typically concern the readout of environmental sensors; the FCM gathers these values from its 4 OMs and then it fills in a status report frame that is sent to shore every second. This report contains all the vital parameters of the electronics of a whole floor, FCM included. This amount of information is then delivered to the Slow Control Management System (§ 2.2.2) that refreshes the monitor display and takes the opportune measures in case of necessity.

When a Slow Control word is received, the element that interprets the layer 0 protocol is the *Word Constructor*. It basically waits when idle characters are received then, when a BSC arrives, it stores the sequent triad of bytes into the SC memory. At this level of the communication protocol it does not care if the SC packet was single or multi word. Since this component analyzes each single incoming byte, it is also responsible for the real-time decoding and execution of a TCR request.

The communication front-side elements are the serializer/deserializer, and the encoder/decoder 8b/10b. The communication start-up is led by the FCM which starts sending a sequence of comma characters at power-up or whenever it senses that the communication towards the OM is lost. A sequence of comma character allows to determine univocally the beginning and the end of a character. The de-serializer starts in an unlocked state, shifting in the serial data until a comma character is recognized; at this point it is locked and it will refresh the parallel output on the next 10 bit received. A lock flag on the de-serializer informs the electronics that the communication is established and that the link is up. Only at this point the front-end enables the outgoing transmission starting with a sequence of comma characters.

The 8b/10b encoder and decoder are implemented with Xilinxs IP cores (Intellectual Property), which means they are pre-configured components to be included into the project.

3.4 Tests and Benchmarks

Now some results of the preliminary tests and performance benchmarks will be shown.

3.4.1 Preliminary tests

First an electrical test was done on the three mounted boards in order to show eventual short circuits. We tried to power the board independently with each single tension by the auxiliary power connector, bypassing the integrated voltage regulators. Afterwards we provided one single +5V general voltage to constat that all the linear regulators generated the tensions required (§ 3.3.2). All the three boards passed successfully this order-0 test.

The following step was to test the digital components in the JTAG chain. Both PROM and FPGA on each board were correctly detected by the Xilinx JTAG cable. The auxiliary PROM for Safe Dual Boot, connected to a separated JTAG chain, was tested as well. All the devices in the JTAG chain have been recognized and directly programmed by the programming cable.

Once the PROM was programmed, the auto-configuration of the FPGA in master serial mode, after a power cycle, was successful. Two specific firmware projects were realized to test the dual boot feature, one to be placed in the primary PROM and the other into the secondary. These two architecture exploited the debug LEDs on the mezzanines. The primary configuration firmware implemented a reverse counter on the 7 segments display, when 0 is reached it switches the configuration MUX (§ Safe Dual Boot in 3.3.3) towards the secondary PROM and afterwards it reboots the FPGA. At this point the secondary configuration is loaded, simply displaying an OK on the two 7 segments displays.

The final proof for the dual boot feature was to load the same down-counter on the two PROMs, one counting on the left digit, and the other counting on the right digit. We observed a cyclic reboot of the system confirming a repetitive good operation of the Safe Dual Boot infrastructure.

Another test was done on the secondary JTAG chain, that connects

the I/O pins of the FPGA to the secondary PROM's JTAG port. We realized a firmware that made of the FPGA a passive bridge from the Xilinx programming cable and the secondary PROM. In this way it was possible to recognize the flash memory, to perform a reading of its ID code and also to re-program it, passing through the FPGA device. A work-around can be done to include all the three device in the same JTAG chain (1 FPGA and 2 PROMs) exploiting the debug strips as it is shown in Fig. 3.21.

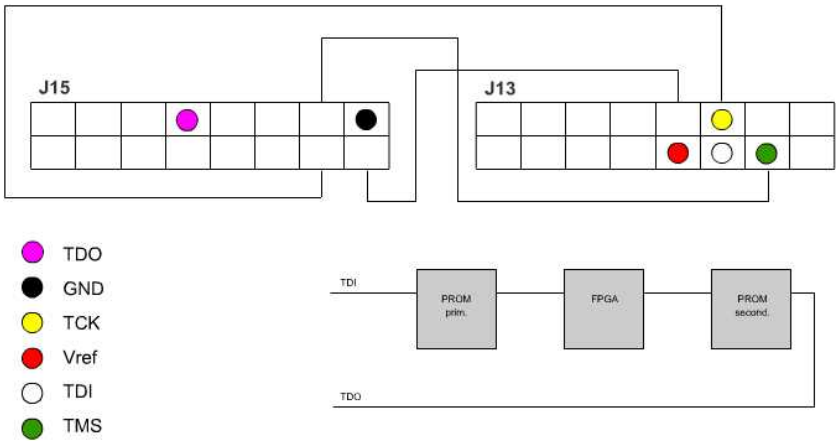

Figure 3.21: **JTAG test workaround**.

Now that is proved that the hardware infrastructure can perform all the requested operations for a remote reconfiguration with Safe Dual Boot, the next thing to do is to foresee an encapsulation of the bit-stream file in a dedicated packet of the Slow Control Protocol.

A brief test on the external PLL programmable chip was realized in order to verify the correct operation of the locked loop and to measure the multiplication factors selected by the relative DIP switch. These tests were performed using the quartz oscillator mounted on the de-tachable mezzanine. To test the frequency multiplication factor we implemented a frequency comparator that counts the PLL oscillations during a period of the quartz. When a multiplying factor is selected with the DIP switch, it was monitored on the 7 segments display. Even in this case the tests gave optimal results.

3.4.2 LIRA acquisition and readout tests

The next hardware feature that have been debugged is the front-end acquisition system that includes the chip LIRA, the Burr Brown ADC and the whole analog interface circuits.

The main tests on this subject were operated in Catania together with the researchers that projected and realized the chip LIRA meant to acquire the PMT signals.

The setup for these tests was made up of a Tektronix AWG arbitrary waveform generator and a digital oscilloscope interfaced to an Agilent logic state analyzer.

The waveform generator allowed to reproduced trains of typical PMT pulses with selectable height, amplitude and frequency in order to simulate SPE and non SPE signals on the front-end inputs.

Some problems arose due to the digital noise that produced undesired effects on two very sensible analog circuits: the feedback circuitry for the 200 MHZ PLL integrated into the LIRA chip and the reference resistor ladders of the signal discriminator (T&SPC).

The bypass capacitors mounted on board, and the decoupling of powering planes were not sufficient to keep clean the analog signals. We tried to improve these conditions adding other external bypass capacitors on the analog power regulators and on close to the analog voltage references. After some re-working of the PCB we reached a stable configuration and the tests could take place.

First of all the functionality of the T&SPC unit has been proved. We sent to the LIRA chip the sequential bitstream containing the thresholds information. The sequence foresees two voltage thresholds and a time window, each piece of information is encoded in 5 bits thus the sequence is 15 bits long. The voltage thresholds are then generated within the LIRA chip by dedicated DAC, while the time window set the up-limit of a counter (refer to section 3.3.1 for details). On the base of these values the pulses should be triggered/not triggered and discriminated into SPE/NSPE events.

Some test results are now presented where the time window discrimination capability was under inspection. For this purpose we programmed the T&SPC register fixing a maximum gap between the two voltage thresholds in order to have all the short pulses marked as SPE, even those with a high profile. Then we fixed a time window of about 80 ns, in this way the T&SPC should classify as NSPE all the events that last longer than this window. In Fig. 3.22 we show the behavior of the classifier in case of a long pulse (superposition of two close pulses). The Agilent Logic State analyzer and the oscilloscope could be inter-

faced together making possible to display on the same waveform both the analog pulse signal and the digital output of the T&SPC.

In Fig. 3.23 instead, is presented a single pulse of the same height as before, but much shorter that is then classified as a SPE event.

All these generated pulses were acquired by the analog delay lines of the chip LIRA under the control of the FPGA. The analog samples acquired are then read out of the front-end and digitized by the ADC.

A picture of an acquired train of pulses is presented in Fig. 3.24. It is a screenshot of the state analyzer software displaying together the waveform triggered on the oscilloscope, interconnected through a GPIB port, and the discrete waveform reconstructed on the digital samples. The oscilloscope presents the analog output of the chip while the probes of the logic analyzer acquire the digital samples put in parallel by the FPGA on a debug strip.

A very close correspondence can be found on the two waveforms but the analog signal is not actually what one would expect. The 16 μs period of the readout (look at the yellow *read enable*) corresponds to 160 samples (160 $\times$ 100 ns, which means 10 samples/μs), the value fixed by the almost full flag policy described in section 3.3.7. All the generated pulses were SPE signals hence only 10 samples were taken for each of them. One should thus expect 16 pulses on the output but this is not so. The causes of this mismatch have been investigated by the group of Catania. Some problems have been found on the sampling clock, whose jitter may alter the dynamic allocation of analog memory cells. All the knowledge deriving from the tests of the chip on this board helped the creators of chip LIRA for the realization of a new analog sampling chip called SAS which should overcome the problems encountered on the last version of chip LIRA.

For a systematic acquisition and storage of the tests results, in Bologna we realized a useful tool for the analysis of the sampled waveforms. This graphical software, realized with LabView, allowed to store the digital samples acquired in a convenient data format in order to display the archived waveforms in a second time. The operation of this software called *Event Visualizer* was combined with ChipScope, a powerful debug tool provided by Xilinx that exploits the JTAG interface.

ChipScope is basically a logic state analyzer integrated on the FPGA, which foresee an IP (*Intellectual Property*) core to be synthesized with the user code, and a PC software that interfaces to the embedded logic state analyzer via programming cable. The trigger of ILA (*Integrated*

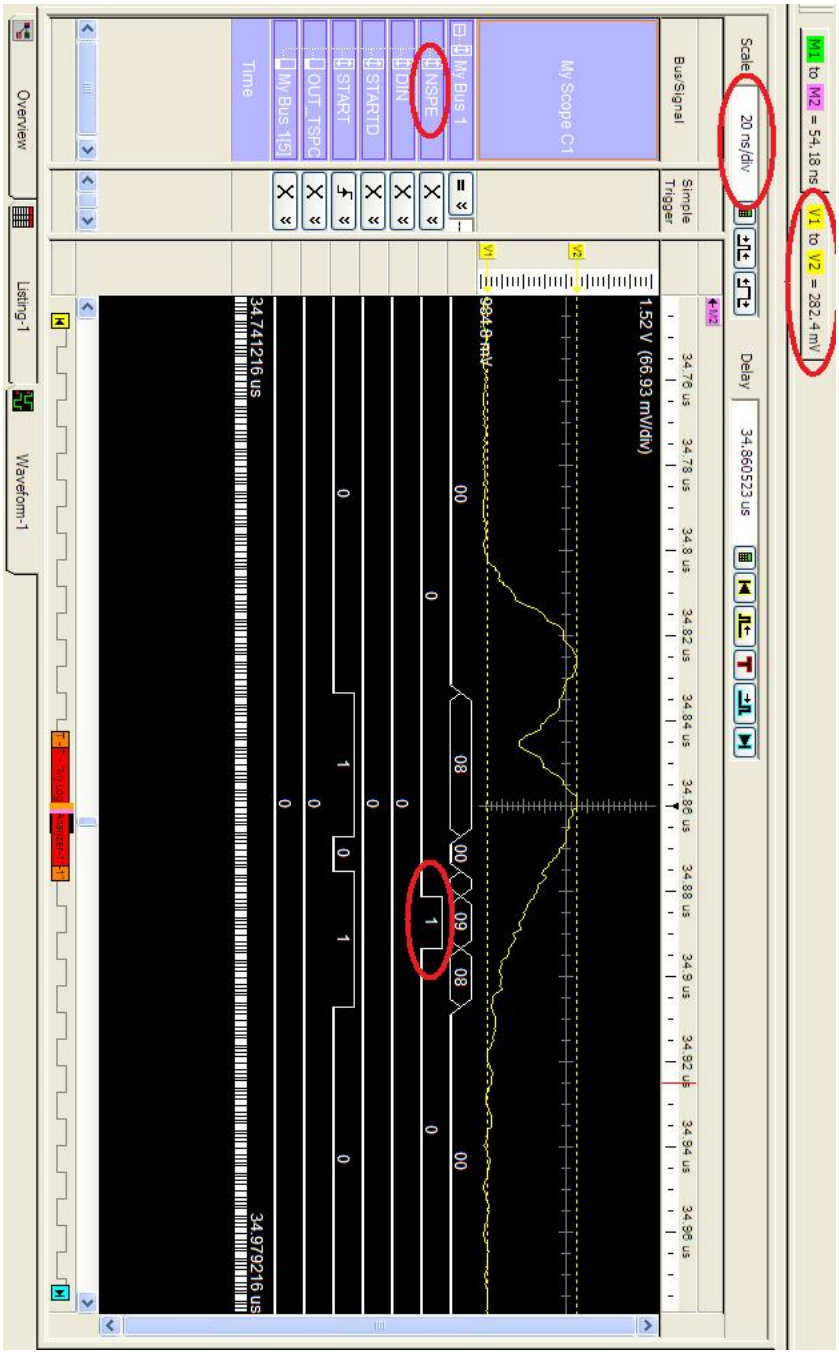

Figure 3.22: **Double pseudo-PMT pulse and T&SPC classification**. The two pulses were generated close enough that they are detected as a one but wider. The classification bit NSPE activates exactly after 80 ns, since that is the programmed time window. In the figure have been highlighted the time scale, the voltage threshold gap and the classification bit. The triggering bit it the StartD signal which fall outside this time window.

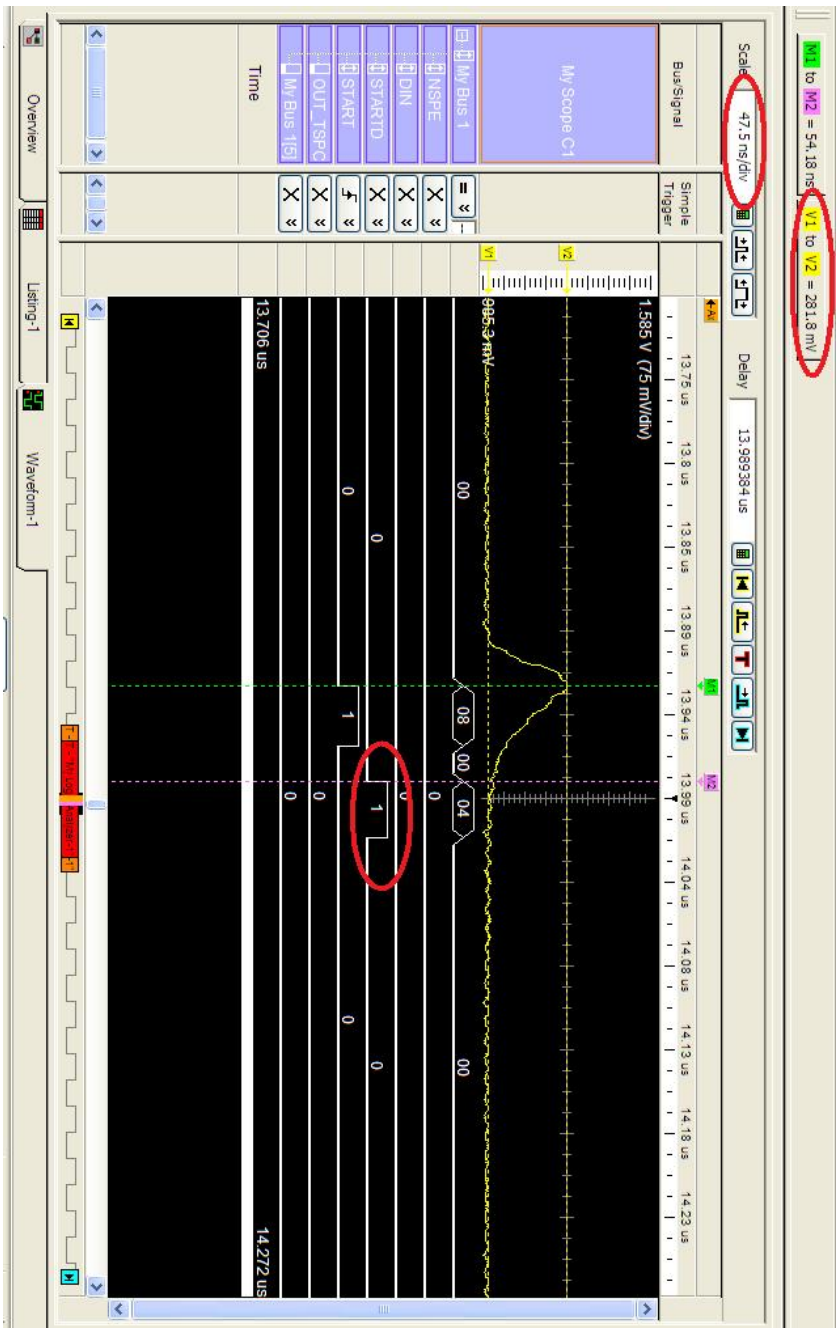

Figure 3.23: **Single pseudo-PMT pulse and T&SPC classification**. The pulse is shorter than the time window, thus a trigger for a SPE event is generated on the StardD signal.

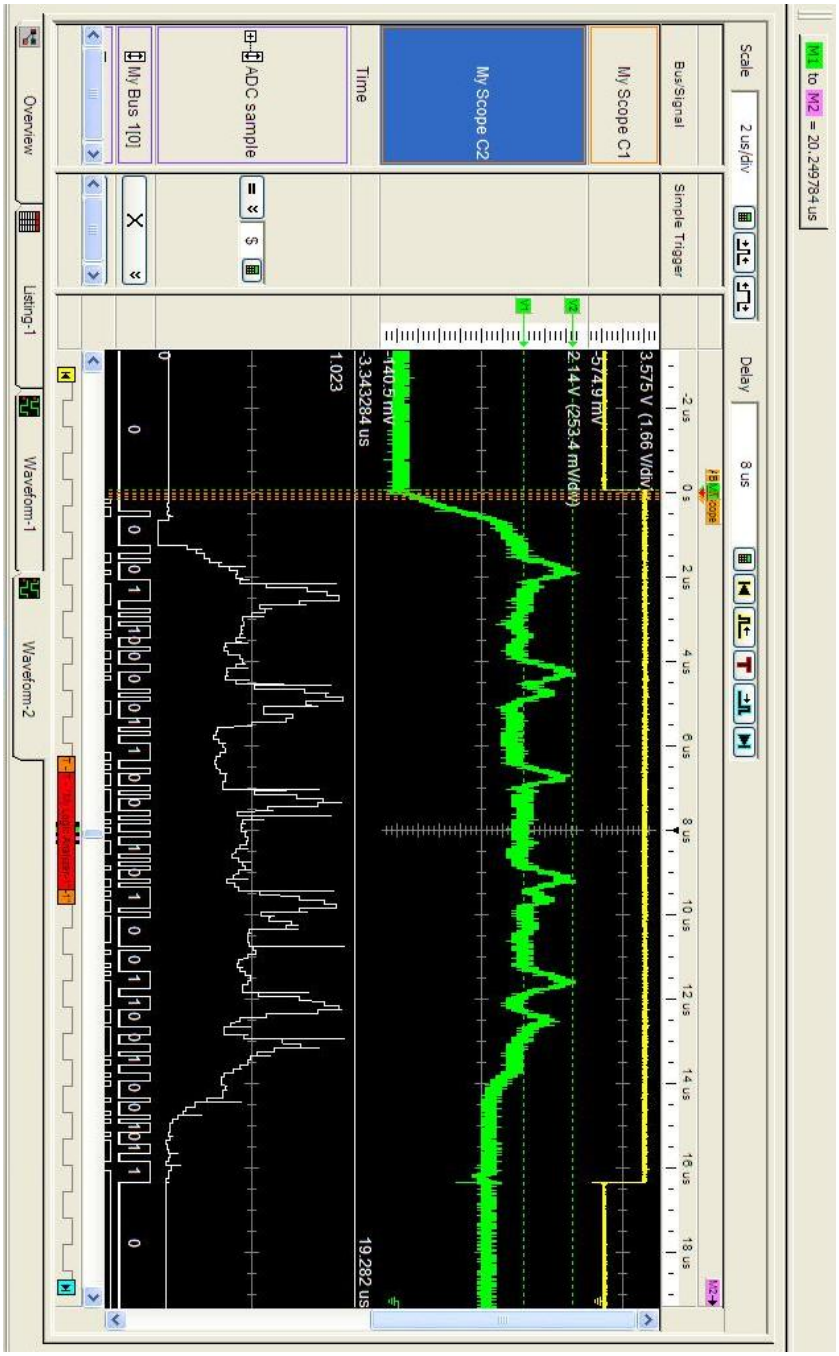

Figure 3.24: **Logic Analyzer waveforms**. The analog samples coming out from the chip LIRA (range 1-2V) are presented in the green waveform taken with oscilloscope probe (250mV/div). The yellow signal is the LVCMOS read enable signal sent to that channel of the chip. The corresponding digital samples coming out of the ADC are acquired and graphically presented by the logic state analyzer. The time scale of the graph is 500 ns/div.

Logic Analyzer) was set on the *read enable* signal and when the triggered digital waveforms were uploaded to the PC memory, we used a ChipScope feature that allows to save the sequence of samples on a proprietary .prn file.

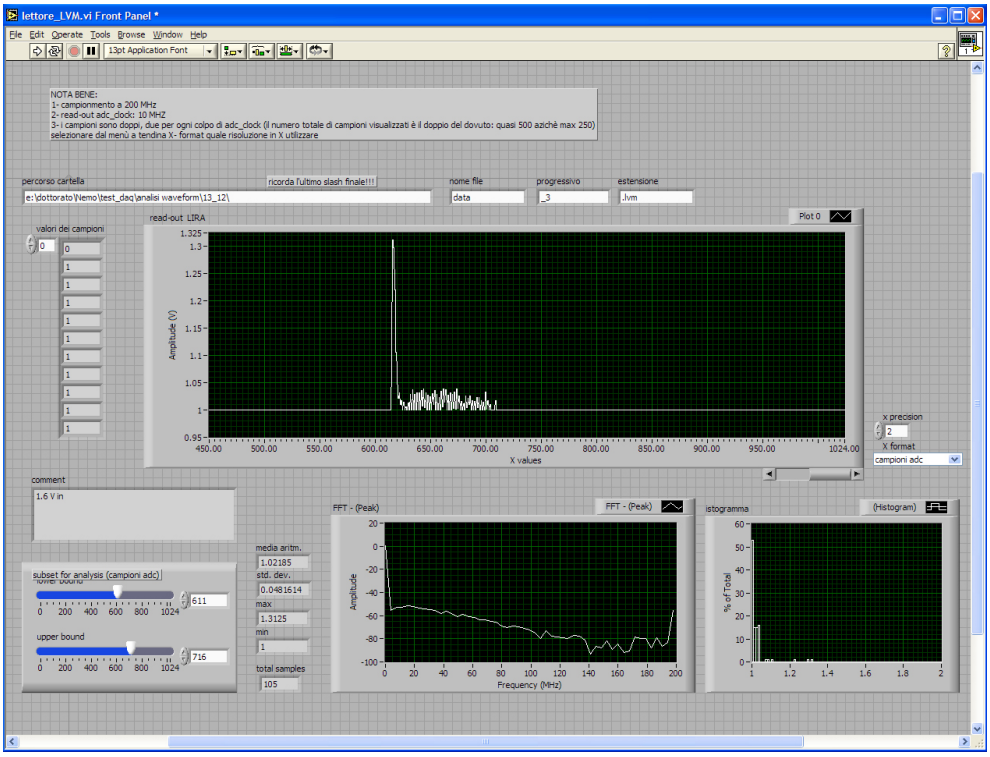

Figure 3.25: **Event Visualizer screenshot**. It is a GUI for the plotting of .lvm waveform files. File path is specified on top. The X scale can be chosen to be presented as ADC samples or time in μs. The numerical value of each sample can be examined on the sample-array inspector on the left. In the lower part of the screen some tools for analysis are provided like a FFT graph and a histogram of samples' dynamic.

Once the data are on a file, our LabView software reads this file, presents the waveform in a graphical way, adds eventual user's comments and saves all this information into a more standard *LabView Measurement File* (.lvm). A screenshot of the program interface is presented in Fig. 3.25. In this case a single pulse is presented, 0.350mV high and right 10 samples wide (5 samples/div). The noise that follows is due to the reading of uninitialized analog memory cells.

The instrumentation setup for this phase of the tests that was re-

alized in Bologna is presented in Fig. 3.26. In parallel to the more advanced test bench prepared in Catania which worked on an equivalent board, we wanted to realize our personal setup structure more concentrated on the debug of the Control Unit.

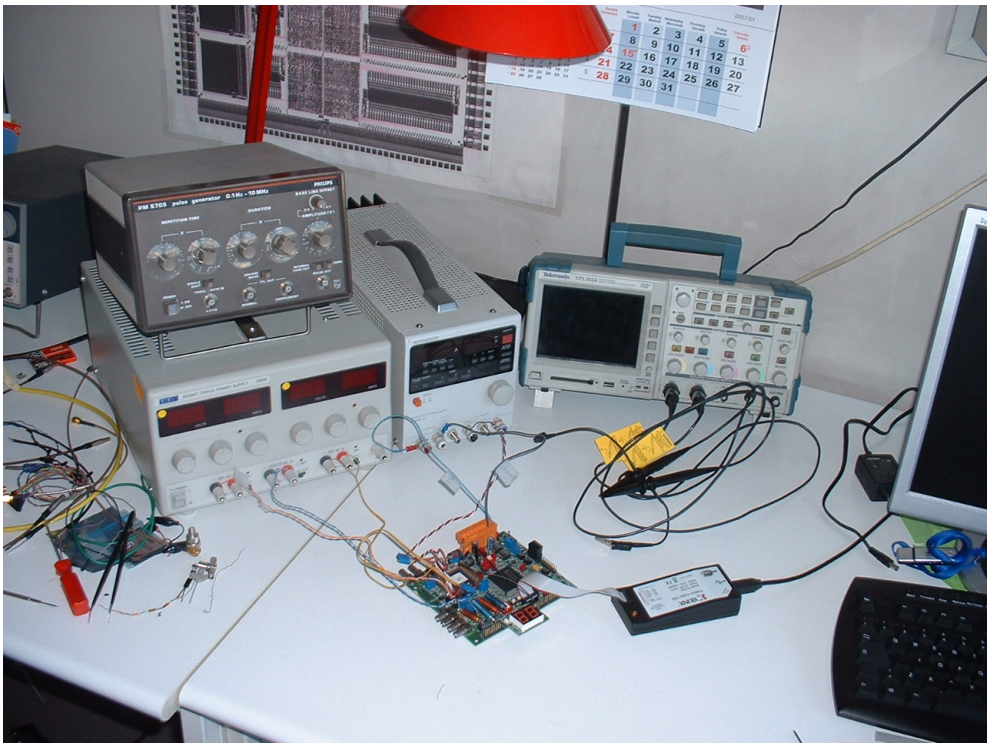

Figure 3.26: **Signal-acquisition test setup**. Instrumentation setup realized in Bologna for the signal acquisition testing.

3.4.3 Data processing benchmarks

As our main purpose was the realization of a reliable and fast read-out architecture we wanted to stress with some simulation benchmarks the readout system and extrapolate some results about the performance.

These tests were performed with software emulations of the firmware, exploiting the same environment used for code debug which is the Mentor Graphics Modelsim tool. This is a VHDL compiler that takes in input the code to be emulated and a test bench file, typically written in VHDL as well. In this file the testing vectors are described and applied to the top entity of the firmware hierarchy which is included

with a *component* statement.

Before the description of the test bench results we must give some theoretical numbers and calculations related to the data acquisition and transmission rates.

As the SPE events are the main component of the 50KHz background measured in the test site, all the calculations that show refer to SPE events only. A whole buffer can store 25 SPE events, with 10 samples each. Supposing to transmit 8-bit samples[12], each event is made up by 80 bits. To these the 16 bits of the time stamp must be added. Every event has also a command character header and a footer (BDC and EDC § 3.3.6) that are 16 bits more. The whole SPE data packet is then 112 bits long. Now we must consider the block encoding 8b/10b, which adds an overhead of 0.25% to the communication. The SPE data packet is then encoded into 140 bits (see Tab 3.7).

		Total
Samples	8 x 10	80
Time Stamp	+ 16	96
Head. & Foot.	+ 16	112
Encoding	× 1.25	**140**

Table 3.7: **SPE data packet dimension in bits**.

We can count on a transmission bandwidth of 20Mbit/s, it follows that the maximum event rate that can be transmitted is

$$\frac{20 \text{ Mbit/s}}{140 \text{ bit/SpeEvt}} = 142.857 \text{ KSpeEvt/s}.$$

This is a theoretical limit that is almost thrice the expected mean rate.

In Fig. 3.27 there is a graph, in logarithmic scale, of the decreasing affordable event rate in each stage of the data acquisition. The first stage, the faster one, is the analog acquisition which can afford 20 MSpeEvt/s (corresponding to the nominal acquisition rate of 200 MS/s), while the last stage is the data transmission (packed and coded) whose rate has been calculated before and it is about 143 KSpeEvt/s. Between each stage there is a FIFO to compensate the gap of writing and reading rates: the buffer between the first and second stage (acquisition-digitizing) is the switched-capacitor array of the chip LIRA

[12]The AD901 ADC has 10 bits of resolution, in our benchmarks we foresee to cut the precision of these samples to 8 bits to emulate the present front-end electronic.

while the buffer that stores samples as they are being packed is the Sample FIFO, finally the output buffer is the Formatted FIFO (§ 3.3.7).

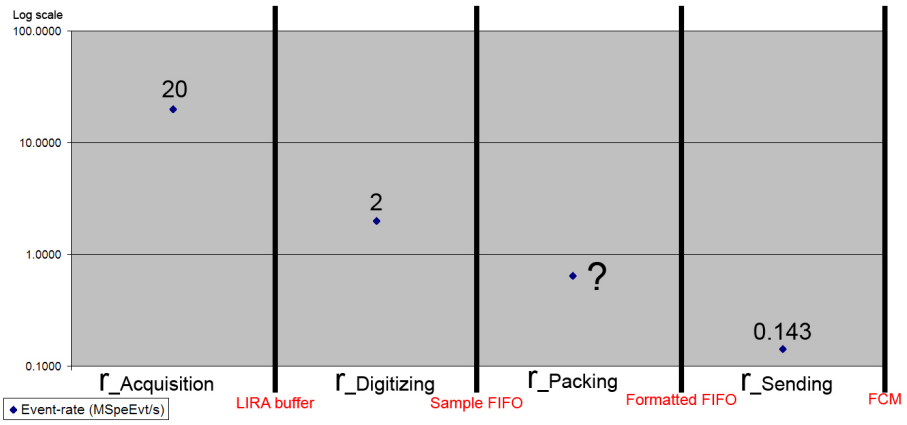

Figure 3.27: **Maximum event-rates graph at different stages for SPE events (MSpeEvt/s, Log scale)**. Acquisition rate derives from the 200MS/s LIRA sampling frequency. Digitizing event rate is limited by the 20 MHz ADC speed. Transfer rate is analytically inferred by the uplink bandwidth and protocol overhead. The maximum event rate of the Packing stage has to be established by simulations.

It is difficult to establish analytically the performance of the data packing unit because of its complexity (the software algorithms running on the sequential processor, the use of embedded multipliers, higher frequency etc...), and for this reason the value missing in the graph has been investigated by the simulations.

The main goal of the firmware project was to realize a packing unit in line with the exponential trend showed in Fig. 3.27, in this case the constant rate performance of the whole system would be dominated only by the up-link bandwidth which is the slowest ring of the chain. Now, what we would like to show with our simulations is that the readout of the samples and the following event building and formatting is not a bottleneck in the performance. This condition would be verified if the maximum event rate of the DPTU processor is greater at least of 143 KSpeEvt/s. The objective in fact was to be able to occupy all the available bandwidth of the up-link. In this case the dead time would be a function only of the transfer bandwidth and of the FIFO depths.

From the VHDL simulations it is possible to extract the parameter

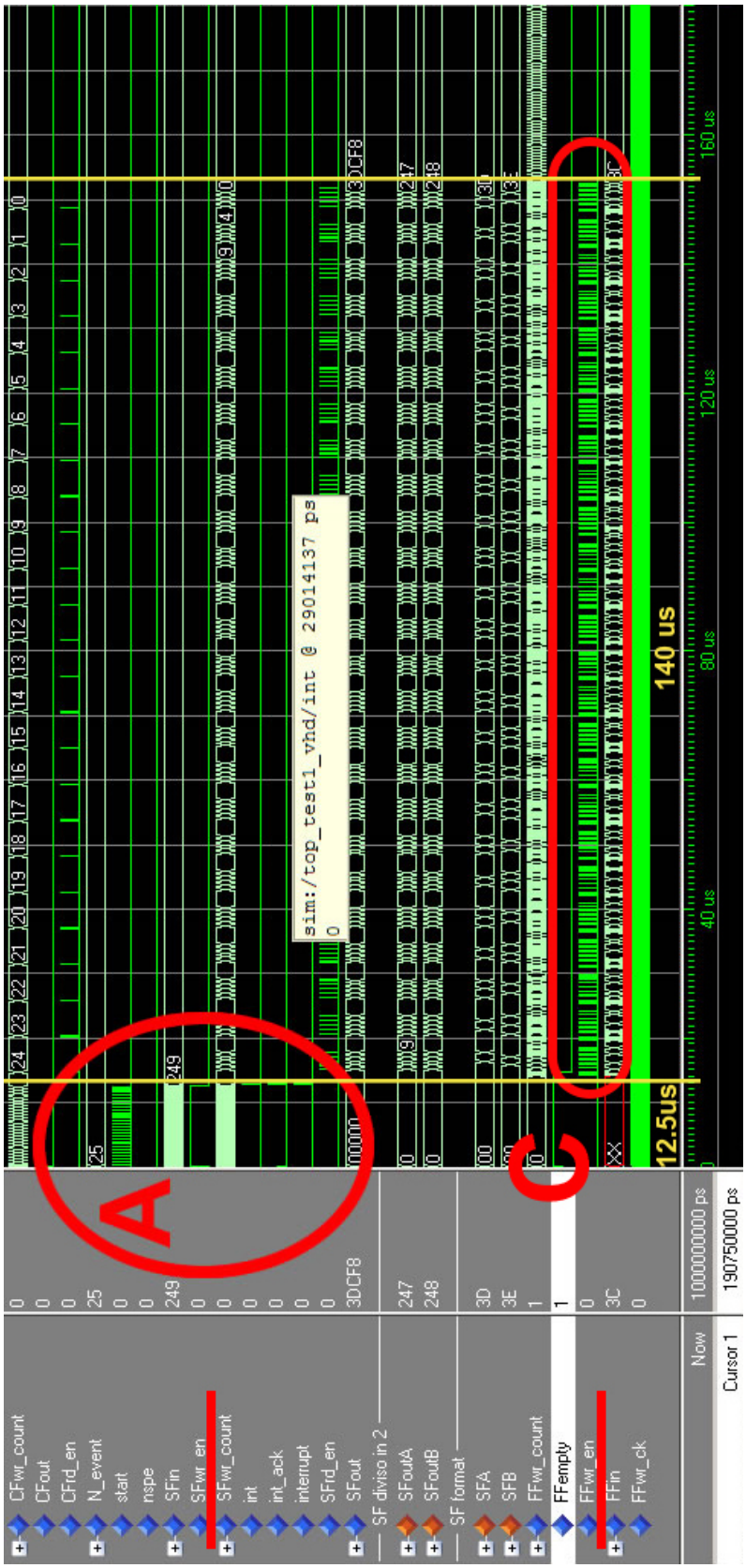

Figure 3.28: **Simulation wave-chart**. LIRA readout and data packing.

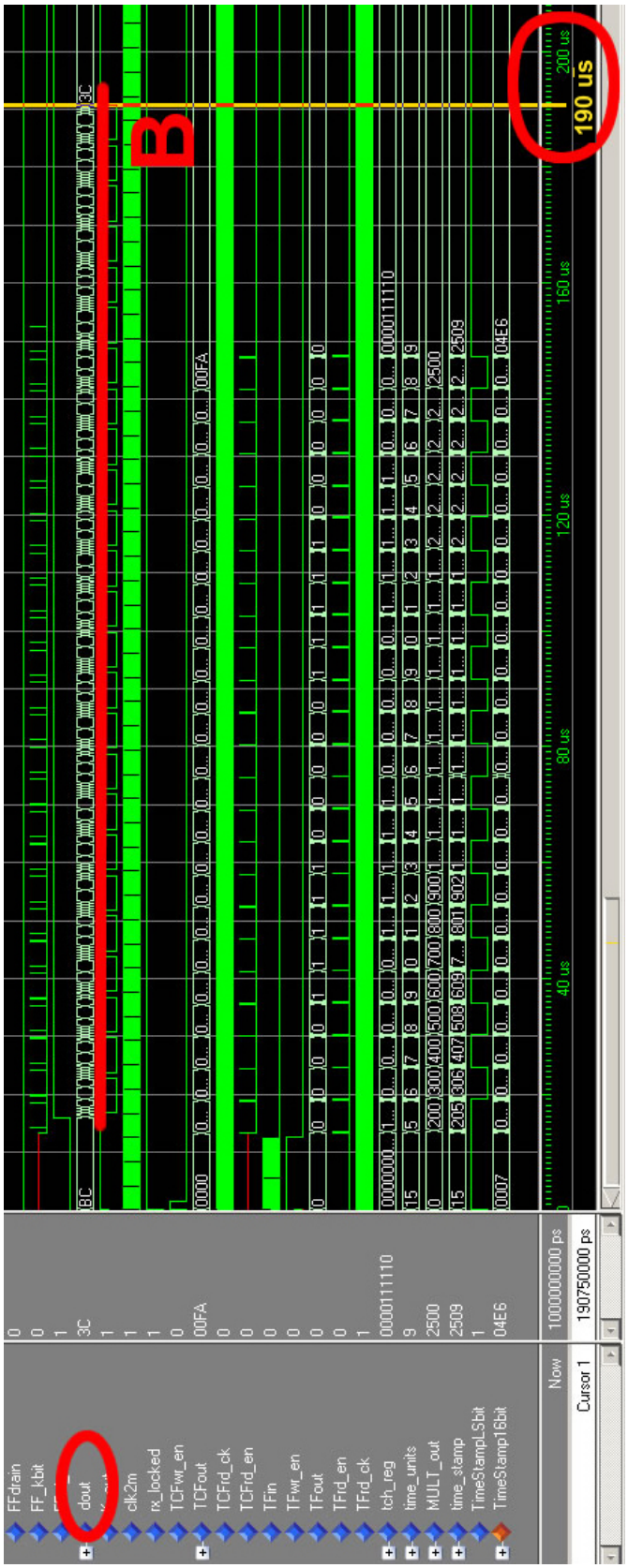

Figure 3.29: **Simulation wave-chart**. Data transfer on *dout* signal.

we miss. In Fig. 3.28 and Fig. 3.29 a waveform graph is presented, it has been generated during a global simulation of the firmware. In the showed region it is possible to observe the whole readout procedure of a full LIRA buffer. The key parts of the graph for our extrapolations are highlighted in red. In the area marked with an **A** are presented the the signals relative to the filling of the Sample FIFO: 250 samples at 20 MHz are stored in 12.5 μs. When a whole LIRA buffer has been read out the data packing precess starts. Time taken by the DPTU to format the whole event set can be estimated looking at the FFwr_en signal (Formatted FIFO write enable) marked with a **C**. This signal runs for 140 μs which means that the average event rate that the DPTU processor can afford is about 25 SpeEvt/140 μs = 179 KSpeEvt/s.

The goal is then achieved as this value is greater than 143 KSpeEvt/s, the maximum event rate allowed by the transmission stage, and it is more or less in line with the exponential trend of Fig. 3.27. The stream indicated with **B** in Fig. 3.29 is the character sequence that enters the 8b/10b encoder, it is possible to see that the whole readout process ends within 190 μs. It's easy to show that this is largely enough as each channel of the LIRA chip, having an average rate $\mathbf{r}_0$ of 50 KSpeEvt/s, and a depth $\mathbf{d}$ of 25 SpeEvt, fills up in a mean time $t = \mathbf{d}/\mathbf{r}_0 = 500$ μs.

It is important to explain that all the rates mentioned in graph and calculations are supposed to be *constant*. This means that if the background has a constant rate of 143 KSpeEvt/s, we would always be able to afford it, no matter how deep our FIFOs are. Otherwise in case of higher continuous rates ($\mathbf{r}_{Sending} < \mathbf{r}_0 < \mathbf{r}_{Packing}$)we would assist to a progressive saturation of the Formatted FIFO as the input rate is greater than the output one, until it goes completely full. At this point the outgoing bandwidth of the Sample FIFO would be limited to the Formatted FIFO outgoing rate ($\mathbf{r}_{Packing} = \mathbf{r}_{Sending}$), which implies that the Sample FIFO will start to fill up as well.

In this case FIFOs' depth determines the time to live of the DAQ system before going to a busy state; the time to live of each FIFO is defined as:

$$\text{Time to live } = \frac{\text{FIFO depth}}{\Delta \text{ rate}} \tag{3.1}$$

where Δ rate is the difference between the incoming and outgoing rate.

For even higher data rates ($\mathbf{r}_{Packing} < \mathbf{r}_0 < \mathbf{r}_{Digitizing}$) the Sample FIFO can go full before the Formatted FIFO, depending on the relation of the two times to live:

$$t_{SF} \lesseqgtr t_{FF} \; ; \; \frac{d_{SF}}{r_0 - r_{Packing}} \lesseqgtr \frac{d_{FF}}{r_{Packing} - r_{Sending}} \tag{3.2}$$

If $t_{SF} < t_{FF}$, once the Sample FIFO is full, it would have the incoming bandwidth limited to $\mathbf{r}_{Packing}$ and this cannot occur as the digitizing from LIRA is always performed at the nominal speed of 2 MSpeEvt/s. Additional space left in the Formatted FIFO would then be wasted, as the system has already reached a busy state. The subdivision of the FPGA block RAM for the two FIFO should then be optimized to avoid this situation. One should thus equals the two terms of relation 3.2 and find which is the dependance of the ratio d_{SF}/d_{FF} from $\mathbf{r}_0$. This is showed in Fig. 3.30.

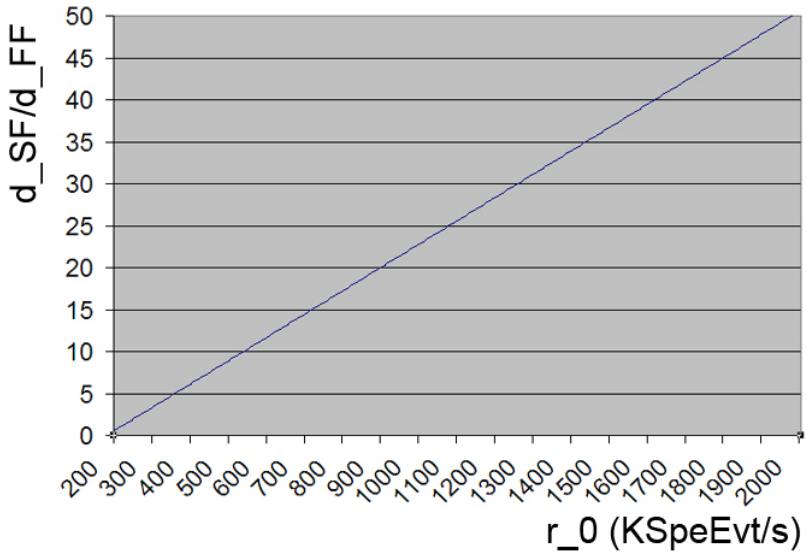

Figure 3.30: d_{SF}/d_{FF} **ratio vs** $\mathbf{r}_0$ **when** $\mathbf{r}_{Packing} < \mathbf{r}_0 < \mathbf{r}_{Digitizing}$ **and the two time to live are equalized**. In this condition the optimization of FPGA available memory follows the ratio expressed by the blue line.

As the $\mathbf{r}_0$ of the burst is not fixed, one can find an interval of values that give at least a mean optimization. Supposing that the rate of burst is uniformly distributed over the interval 200-2000 KSpeEvt/s we had chosen to realize a Sample FIFO 25 times greater than the Formatted FIFO.

3.4.4 FCM communication test

This section will describe the latter test performed on the DAQ board. As we have tested the acquisition of pulse signals with the LIRA chip, and we have obtained good results with the firmware performance test

bench, we wanted to test our system within the real data acquisition infrastructure.

Thus we set up in our laboratory a full data acquisition chain in order to test the transmission capability of our board. The experimental set up was made up of the DAQ board connected by the foreseen twisted pair cable to the off-shore FCM.

Due to FCM board dimensions, the on-shore counterpart could not be housed inside the main acquisition PC. A PCI extender board has then been used as a bridge towards an empty PC chassis where the on-shore FCM has been installed. The drivers of the hardware have been installed in the operating system used for the tests (Microsoft Windows XP).

Figure 3.31: **DAQ board communication test bench with FCM**. Leftmost, in left picture the off-shore FCM connected to the DAQ board. Close up in right picture, inside the chassis there is the on-shore FCM plugged to the PCI extender.

The DAQ board and the two FCMs have been powered by a DC power supply while the PCI extender board in the empty chassis was powered by a dedicated ATX standard power source.

The two FCMs were then connected with two mono-modal fiber patch-cords, one for each stream direction. In each patch-cord was added a 15 db attenuator to emulate a cable length of 20 Km.

A picture of the experimental setup is presented in Fig. 3.31.

The communication towards the front end has been tested with the exchange of Slow Control packets. The software interface that we used for these tests is the FCM Manager (`FCMmgr.exe`), an application that runs on the acquisition PC. This software provide an interface between the hardware and the user with a *Graphical User Interface* and a network server. In our case we used the Slow Control debug interface of the GUI.

The Slow Control interface works as follows. Each command line must contain a 32 bit word in hexadecimal notation. These commands are sent to the FCM's DSP and the first line must individuate which of the four available front-ends is addressed. The bit interval `[23:16]` of the first word contains the information of the addressed module coded in the following manner: `0x09` for the FE on connector number 1, `0x0A` for connector 2, `0x0B` for connector 3 and `0x0C` for connector number 4. The remaining 16 bits of the first word individuate the length of the following word array to be sent expressed in bytes. An example is presented in Tab. 3.8.

32 bit word	**meaning**
`0x000C0008`	Address `0x0C` and data length = 8 bytes (=2 words)
`0x00331254`	Slow control command `0x33`, operand `0x1254`
`0x00331254`	Check word

Table 3.8: **Code sequence example for the slow control debug interface**. In this case LIRA threshold set command has been sent to the front-end module board on channel 4.

These are 32 bit `unsigned_int` representation of the 24 bit word that reach the DSP on the on-shore FCM, for this reason a couple of hexadecimal 0s must be added at the beginning of each word.

For each request sent to the front-end there will be a related response in the response frame of the console. If everything works fine the response code will echo the transmitted command with the RSN bit toggled (see 3.3.6). Also the response packet have a check word and we must find it in the returning pattern.

A separate window is dedicated to the *Periodic Data Packets* foreseen by the protocol; Each PDP frame is made up of 36 words with a fixed format and it is refreshed every second. Having one front-end only connected to the FCM we find empty all the field related to the other three DAQ-boards, but in the words 8, 9, 10 and 11 of the frame we can find the OM3 temperature, humidity, high voltage settings and

thresholds.

These tests gave good results as it was possible to send instructions and receive back the response packets. The automated slow control unit worked fine as well as the pattern in the PDP frame were correctly received.

Part II

The data acquisition system for the characterization and test of a Monolithic Active Pixel Sensor

Chapter 4

High-resolution vertex detectors

In modern high-energy physic experiments, particles are accelerated to ultra-relativistic velocities and then collided to study the elementary constituents of matter and their interaction properties. Accelerators are built to generate high interaction rates in a well known space point in order to provide the necessary amount of high-resolution event statistics for an efficient analysis.

Interactions are performed accelerating particles towards a fixed target or by head-on collisions of two projectiles. The LHC (Large Hadron Collider) at CERN and Tevatron at Fermilab, for example, work with head-on collisions. This is because the energy that can make new particles is the *center of mass energy*, and in a head-on collider experiment this is simply the sum of the two beam energies. In stationary target experiments, instead, when relativistic speeds are reached by the accelerated particle, the center of mass energy is significantly less that the sum of the two interacting particles.

In the hot spots where particles are collimated and collided, big detectors are built to identify all the particles that come out of the interaction point. Many of these have a very short mean lifetime and decay into more stable particles before reaching the first layer of the detector, generating secondary interaction vertices. Typically detectors are made up of several layers which surround the interaction spot.

The innermost layers usually have a higher spatial resolution in order to track, with the highest precision possible, the direction of each incoming particle, typically of the order of 10 μm. This is very important because the interaction and decay vertices can be distinguished by the geometrical extrapolation of these directions. Once the topology of the process is know, calorimeters and long range trackers exploit strong

and constant magnetic fields to extrapolate the energy and the nature of those particles.

High resolution vertex detectors, are exploiting by several years the silicon technology. It provides the high spatial resolution required with few noise-hits, and an affordable radiation hardness. Typically the closest detector to the beam pipe is a barrel of silicon pixel sensors due to their very high spatial resolution. The outer layers of the trackers are made of silicon strip sensors, still with high spatial resolution but in one dimension only. For this reason the cost per area of silicon strips is usually much lower respect to pixels. A lower cost of strips allows to instrument the same solid angle of the pixels layer at larger radii.

In general, silicon technology is relatively cheap and diffuse as the producing foundries are spread worldwide to feed the huge electronic market. Moreover, a silicon detector can be thinned to few hundreds of microns, drastically reducing the material budget of the whole vertex detector. A high material density of the detector would increase the probability of secondary interactions with the detector matter producing multiple scattering, a side effect that physicists want to avoid as it "blurs" the particle track.

In the following sections the vertex individuation problem in a standard high energy experiment will be described. As the second part of this thesis concerns an innovative readout system for pixel sensors, a brief description of a standard pixel application is then given to let the reader discern the innovations introduced.

4.1 The individuation of vertices

In high energy hadronic colliders, after a collision a jet of flavored heavy mesons is produced. Typically their mean lifetime ($< 10^{-10}s$) allows them to travel only for few hundreds of microns away from the primary vertex before decaying into more stable particles. As they decay before reaching the first layer of the detector, they must be inferred by their child-particle's tracks. The problem arise if the precision with which one reconstructs the track does not permit to discern the secondary vertices from the primary, in this way it would't be possible to individuate the decaying particle.

If we knew with absolute precision the direction of each particle that reaches the first layers of the detector, we would be able to individuate all the branchings deriving from the main vertex, and eventually to discern some interesting exotic quark mixture with extremely short lifetime/short range.

To maximize the precision in the individuation of a vertex, as shown in Fig. 4.1, the measure on the first layer must be the most accurate one.

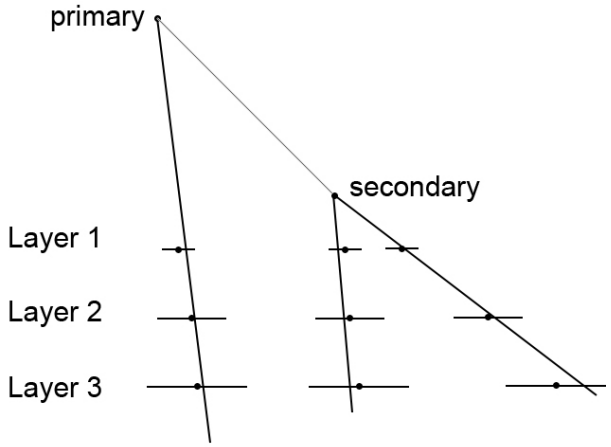

Figure 4.1: **Vertices reconstruction scheme**. Three detector layers, decreasing precision with radius. The light track is inferred as parent of the secondary vertex

That is the reason why the first layer of vertex detectors is typically instrumented with pixel technology, up to now the particle detector with the highest resolution (tens of microns). As the radius from the beam pipe increase, the instrumented area that needs to be covered increases linearly with it ($A = 2\pi\rho z$). A loss in spatial accuracy is considered necessary and affordable to keep low the overall cost of the detector, thus cheaper technologies are used in next layers. It must be considered that pixel technology is used also since it is capable to acquire a high density flux of particles: as the density decreases with radius, it is possible to cope with the same event rate exploiting other types of silicon detectors, such as silicon drift or silicon strip.

4.2 The ALICE ITS vertex detector

The ALICE detector (*A Large Ion Collider Experiment*) is a dedicated heavy-ion detector to exploit the unique physics potential of nucleus-nucleus interactions at LHC energies.

The main tasks of the ALICE vertex detector, called ITS (*Inner Tracking System*), are to localize the primary vertex with a resolution

Layer	Type	r(cm)	$\pm z$ (cm)	Area (m^2)	Channels
1	pixel	3.9	14.1	0.07	3 276 800
2	pixel	7.6	14.1	0.14	6 553 600
3	drift	15.0	22.2	0.42	43 008
4	drift	23.9	29.7	0.89	90 112
5	strip	38.0	43.1	2.20	1 148 928
6	strip	43.0	48.9	2.80	1 459 200
Total area				6.28	

Table 4.1: **Dimensions of the ITS detectors**

better than 100 μm, to reconstruct the secondary vertices from decay of hyperons and D and B mesons, to track and identify particles with momentum below 200 MeV/c, and finally to improve the momentum and angle resolution for particles reconstructed by the *Time Projection Chamber* (TPC).

As the vertex detectors aim to determine with high precision the interaction vertices, they are usually very close to the beam pipe; in this case the ITS is composed by 6 cylindrical layers coaxial with the beam pipe, located at different radii, between 4 and 43 cm.

The Inner Tracking System has been realized exploiting three different silicon detector technologies. The 2 innermost layers are instrumented with pixel detectors, for a total of about 10 million channels, as the expected particle density can reach 50 particles per cm^2. The two intermediate layers are build with SDD (*Silicon Drift Detectors*), while the two outer layers, where the density is expected to be of one particle per cm^2, are equipped with double-sided SSD (*Silicon Strip Detectors*).

The four outer layers have analogue readout and therefore can be used for particle identification via dE/dx measurement in the non-relativistic ($1/\beta^2$) region. This feature gives the ITS stand-alone capabilities as a low-p$_t$ particle spectrometer.

The main parameters for each of the three detector types are summarized in Tab. 4.1 and 4.2 [6].

The momentum and impact parameter resolution for low-momentum particles are dominated by multiple scattering effects in the material of the detector; therefore the amount of material in the active volume has been kept to a minimum. The silicon detectors used to measure ionization densities (drift and strips) must have a minimum thickness

Parameter	m.u.	Silicon Pixel	Silicon Drift	Silicon Strip
Spatial precision $r\phi$	μm	12	38	20
Spatial precision z	μm	70	28	830
Two track resolution $r\phi$	μm	100	200	300
Two track resolution z	μm	600	600	2400
Cell size	μm^2	50×300	150×300	95×40000
Active area per module	mm^2	13.8×82	72.5×75.3	73×40
Readout channels per module		65536	2×256	2×768
Total number of modules		240	260	1770
Total number of readout channels	M	15.729	0.133	2.719
Total number of cells	M	15.7	34	2.7
Average occupancy (inner layer)	%	1.5	2.5	4
Average occupancy (outer layer)	%	0.4	1.0	3.3
Power dissipation in barrel	kW	1.5-2.0	0.51	1.1
Power dissipation end-caps	kW	-	0.41	1.5

Table 4.2: **Comparison between the three ITS detector types**. A module represents a single detector front-end chip.

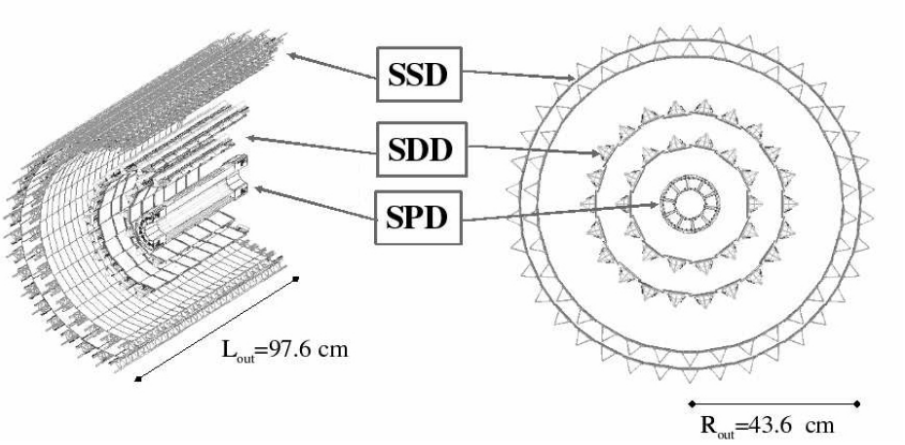

Figure 4.2: **Layout of the ALICE ITS detector**.

of approximately 300 μm to provide acceptable signal-to-noise ratio.

4.2.1 Silicon Pixel Detector

The SPD is a fundamental element for the determination of the position of the primary vertices as well as for the measurement of the impact parameter of secondary tracks originating from the weak decays of strange, charm and beauty particles.

The SPD is based on hybrid silicon pixels, consisting of a two-dimensional matrix (sensor ladder) of reverse-biased silicon detector diodes bump-bonded to readout chips. Each diode is connected through a conductive solder bump to a contact on the readout chip corresponding to the input of an electronics readout cell. The readout is binary: in each cell a threshold is applied to the pre-amplified and shaped signal, and the digital output level changes when the signal is above a programmed threshold.

The ladder consists of a silicon sensor matrix bump-bonded to 5 front-end chips. The sensor matrix includes 256 $\times$ 160 cells measuring 50 μm ($r\phi$) by 425 μm (z). Longer sensor cells are used in the boundary region to ensure coverage between readout chips. The sensor matrix has an active area of 12.8 mm ($r\phi$) $\times$ 70.7 mm (z). The front-end chip reads out a sub-matrix of 256 ($r\phi$) $\times$ 32 (z) cells. The thickness of the sensor is 200 μm, the smallest that can be achieved with an affordable yield in standard processes. The thickness of the readout chip is 150 μm; the readout wafers are thinned after bump deposition,

before bump bonding. The two ladders are attached and wire bonded to the high density aluminium/polyimide interconnect (pixel bus). A 200 μm clearance between the short edges allows for dicing tolerances and ease of assembly. The pixel bus carries data/control bus lines and power/ground planes. The *Multi-Chip-Module* (MCM), wire bonded to the pixel bus and located at the end of the half-stave, controls the front-end electronics and is connected to the off-detector readout system via optical fibre links (see Fig. 4.3

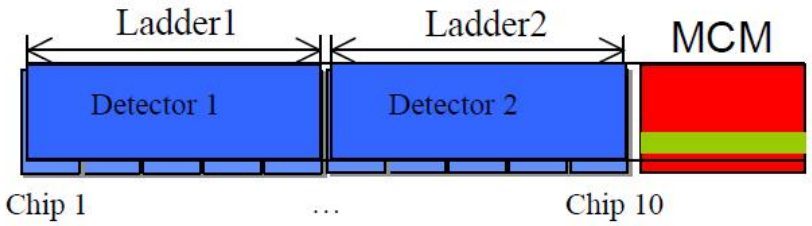

Figure 4.3: **Schematic diagram of an half-stave**. Two ladders are connected to a Multi-Chip-Module by means of a pixel bus to form an half-stave.

Two half-staves are attached head-to-head along the z direction to a carbon-fibre support sector to form a stave. Each sector supports six staves: two on the inner layer and four on the outer layer. Ten sectors are then mounted together around the beam pipe to close the full barrel. In total, the SPD (60 staves) includes 240 ladders with 1200 chips for a total of 9.8×10^6 cells. The sectors are equipped with cooling capillaries embedded in the sector support and running underneath the staves (one per stave). The heat transfer from the frontend chips is assured with high thermal conductivity grease.

The ALICE pixel readout chip is a mixed signal ASIC developed in an IBM 0.25 μm CMOS process (6 metal layers) with radiation tolerant layout design. Each chip contains 8192 readout cells of 50 μm $\times$ 425 μm arranged in 32 columns and 256 rows. The size of the chip is 13.5 mm $\times$ 15.8 mm including internal DACs, JTAG controller, chip controls and wire bonding pads. The chip clock frequency is 10 MHz.

The ALICE1LHCB chip [44], was developed to serve two very different applications, tracking and vertex detection in ALICE and particle identification in the RICH detector of LHCb. To satisfy the different needs for these two experiments, the chip can be operated in two different modes. In tracking mode all the 50 μm $\times$ 425 μm pixel cells in the 256$\times$32 array are read out individually, whilst in particle identification

mode they are combined in groups of 8 to form a 32×32 array of 400 $\mu m \times 425\ \mu m$ cells. Anyway we will consider only the vertex-detection operation mode only.

Both the analog and digital circuitry has been designed to operate with a 1.6V power supply, and the total static power consumption is about 500 mW. The pixel cell itself is divided into an analog and a digital part. The analog front-end consists of a pre-amplifier followed by a shaper stage with a peaking time of 25 ns. In the full chip implementation the front-end is differential, with one input carrying the detector signal and the other tied to a clean reference.

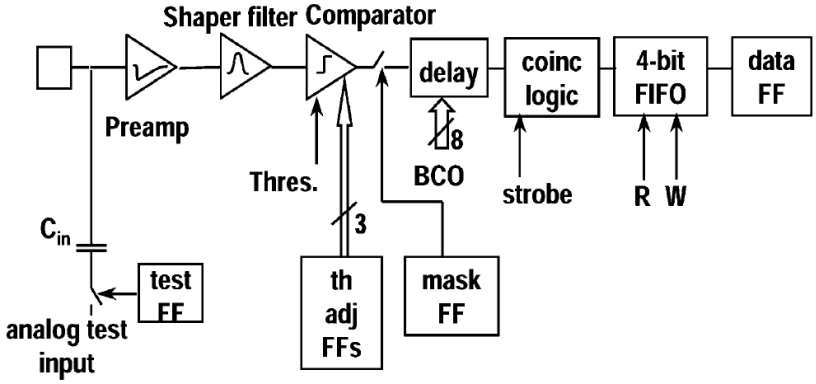

Figure 4.4: **Schematic block diagram of the content of one pixel of the ALICE1LHCb chip**.

Each pixel can be individually addressed for electrical testing and for masking. The mask flip-flop allows the pixel to be enabled or disabled, in case of noisy pixel this should prevent from injecting spurious data into the data stream.

A discriminator compare the output of the shaper (see Fig. 4.4) with a threshold fixed globally across the chip. Each pixel contains three logic bits which can be used to adjust the thresholds on a pixel-to-pixel basis. Together with the global threshold for every chip this provides the required pixel-to-pixel uniformity over the full system.

The outputs of the discriminators in the pixel matrix provide both a fast-OR and fast-multiplicity signal which are output off-chip. The former gives a digital pulse if one or more pixels are hit on the chip. The latter gives an analog current proportional to the number of pixels hit. Both signals can be used for diagnostic purposes and for triggering

as they come immediately after a hit has been detected.

The discriminator output feeds the digital part of the cell. In the first stage there are 2 digital delay units storing the hit during the trigger latency. When a hit is received from the discriminator an 8 bit Gray-encoded counter is latched into the digital delay line. If a trigger coincidence is found, then the hit is passed to the next stage, a 4-event FIFO which acts as a multi-event buffer and de-randomizer. Finally the content of the FIFO cells, waiting to be read out are loaded into a flip-flop by the Level-2 trigger. All the data flip-flops of a column form a shift register, and data is shifted out using the system clock (see Fig. 4.5).

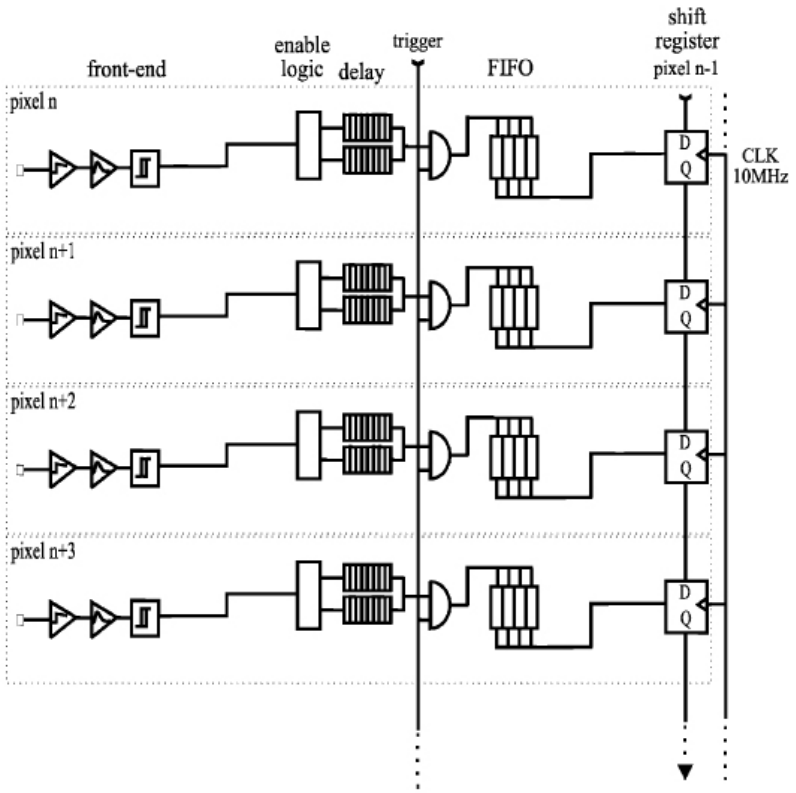

Figure 4.5: **Schematic view of the readout of pixel column in the ALICE1LHCb chip** .

Five flip-flops are used also to latch the configuration and programming bit of each cell. One switch on/off the test input to the front-end,

one mask or activate the pixel cell, and the latter three are used for the fine threshold adjustment as described above.

Chapter 5

APSEL4D - a MAPS chip with integrated readout logic

With the increasing luminosity of modern accelerators ($10^{34}\ cm^{-1}s^{-1}$ at LHC) increases also the flux of particles coming out of the interaction point. For this reason faster detectors are being developed for the stringent requirements of the experiments at future colliders, such as the SuperB Factory or the International Linear Collider. They will need to fulfill very tight requirements on position resolution, readout speed, material budget and radiation tolerance.

Vertex detectors need particularly a technological upgrade as they are required to be closer and closer to the interaction point, where particle density can reach several tens of particles per cm^2. This means a higher radiation dose and a higher hit-rate, with the consequent need of fast sensors with high granularity.

The main challenge is the upgrade of pixel sensors as they are exposed to the highest radiation dose and it is difficult to speed-up their read-out process for their extreme density. That's why new kinds of silicon pixel detectors are investigated at the moment.

Traditional pixel sensors integrate on a silicon chip a bi-dimensional array of cells that can be addressed typically in a pixel-by-pixel way or column-by-column by external read-out logic. The typical sensing technique is based on the collection of the charge that the impinging charged particle forms in the epitaxial layer. The electrons move simply by diffusion then they are collected by the cathode of the N-well/P-epitaxial reverse-biased diode. Charge to voltage conversion is provided by the sensor capacitance, and thus collecting electrodes are kept as small as possible to increase the conversion factor ($V \propto Q/A$ where A is the collecting surface). Three NMOS transistors are then typically used to address and reset the single pixel, for this reason this simple

architecture is called 3T and it is illustrated in Fig.5.1.

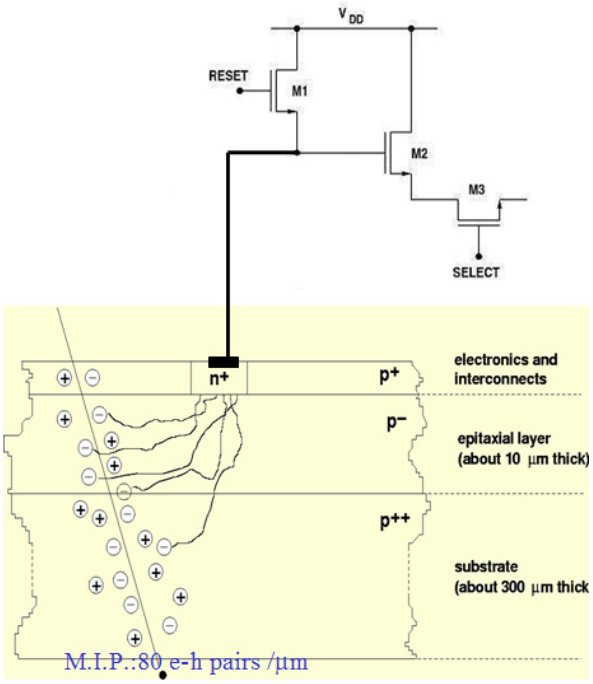

Figure 5.1: **3T NMOS traditional pixel sensor**. The diffusing charge is collected by the N-well electrode and it is read as a voltage tension exploiting the sensor intrinsic capacitance.

This architecture presents several limitations: first of all the collecting area that, as we said, is limited by the voltage conversion factor. On the other hand, since it collects diffusing electrons, a limitation in its surface implies a limitation in the efficiency. In addition the pixel-by-pixel access makes the hit read-out process drastically slow.

5.1 The SLIM5 Collaboration proposal

To overcome these problems the SLIM5 collaboration is working on a Monolithic Active Pixel Sensor (MAPS) which features the following characteristics: the sensor cell is *active*, which means that the front end of the pixel is driven by active electronic components like a preamplifier, a shaper and a discriminator; in addition it is *monolithic* as it integrates a standard CMOS read-out logic. Modern foundry technologies allow to

create a large deep N-well to separate the P epitaxial layer from another P region. Inside this region ordinary NMOS operational amplifiers can be implemented to realize the active circuitry of the pixel, and the large deep N-well can act as a large electron collecting anode. A cross-section view of a deep N-well (DNW) architecture is shown in Fig. 5.2.

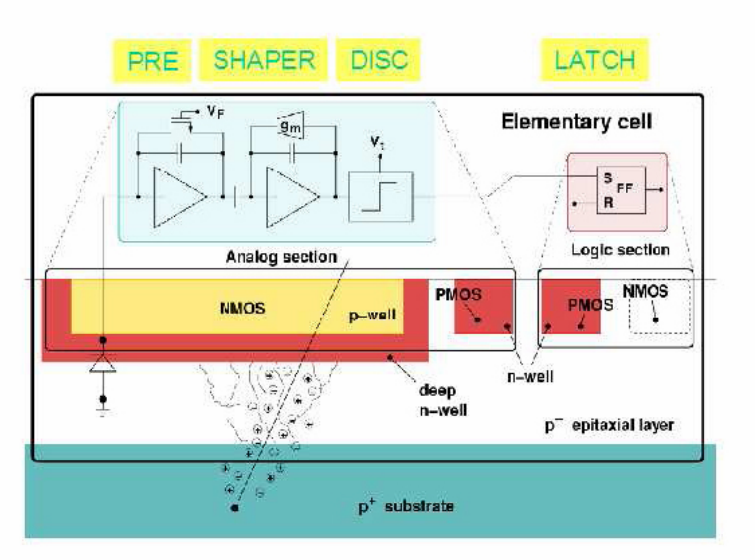

Figure 5.2: **Monolithic Active Pixel Sensor**. The deep N-well acts as a charge collector and the inclosed NMOS electronics perform signal amplification, shaping and discrimination.

Once the signal has been amplified and discriminated, the CMOS latch, realized outside the DNW, stores the hit/non-hit binary information of that cell. The latch can thus be easily interfaced to the sparsification logic realized with CMOS standard-cells. In addition, now that the voltage gain is determined by the feedback capacitance of the charge preamplifier, the size of the collecting electrode can be increased up to about 900 μm^2 in a pixel cell of 50 μm pitch. It is thus possible to include within the pixel some small competitive n-well regions, crucial to develop the CMOS logic for the readout, still keeping the sensor fill factor at the level of 90%.

During the activity of the SLIM5 collaboration several prototype chips of a series called APSEL were realized with the STMicroelectronics 130 nm triple well technology. The starting projects implemented single pixel and small matrix of pixels with simple sequential readout

logic. The results on these prototypes proved that the new design, proposed for DNW MAPS, is viable and that it presents good sensitivity to photons from ^{55}Fe and electrons from ^{90}Sr.

These studies led to the production of a third series of the APSEL chip. APSEL3 has been carried out simply after some rearrangement of the front-end pixel in order to lower the Signal-to-Noise ratio and power consumption. The total sensor capacitance has been reduced from 500 fF to 300 fF using for the new collecting electrode a combination of a DNW region and a standard N-well area. Power dissipation of the single pixel resulted halved down to 30 μW. Particular attention has been given also to cross-talk problems induced by the digital logic signals. One of the six metal layer available in the used technology has been used as a shield between the sensor and the digital lines.

Two different version of the APSEL3 chip have been realized, each one with a different implementation of the shaper. One adopting as feedback a transconductor (APSEL3T1) and the other exploiting a current mirror circuit (APSEL3T2).

The characterization of the APSEL3 chips with radioactive sources confirmed the expected improvements: for the two front-end versions Signal-to-Noise ratios have been measured between 20 and 30 for *Minimum Ionizing Particles* (MIP) from a β source. The response curve of the APSEL3T1 chip sensor to the ^{90}Sr electrons is shown in Fig. 5.3 (taken from [24]). The cluster signal has been fit with a Landau distribution with a most probable value of 128 mV, corresponding to a SNR of 24.

For the absolute gain calibration a ^{55}Fe source has been used for its sharp 5.9 KeV emission line. With this source, the average cluster signal measured for a MIP correspond to about 1000 electrons, while the average pixel equivalent noise charges in the two version of the front-end are respectively 46 e^- and 36 e^-.

A first version of a DNW MAPS device, integrating a 8×32 matrix with a smart integrated sparsifier and readout logic, has thus been realized and it was named APSEL3D, where D stands for *Digital*. The immediate following revision, APSEL4D, is a general improvement of the previous versions; the improvements deal mainly with the readout logic and with the pixel layout, trying to reduce cross-talk effects. A bigger matrix has also been integrated, now it covers an area of about 10 mm^2. The 4D version was then submitted to a test beam and so the architecture of this latter revision will be described in more details.

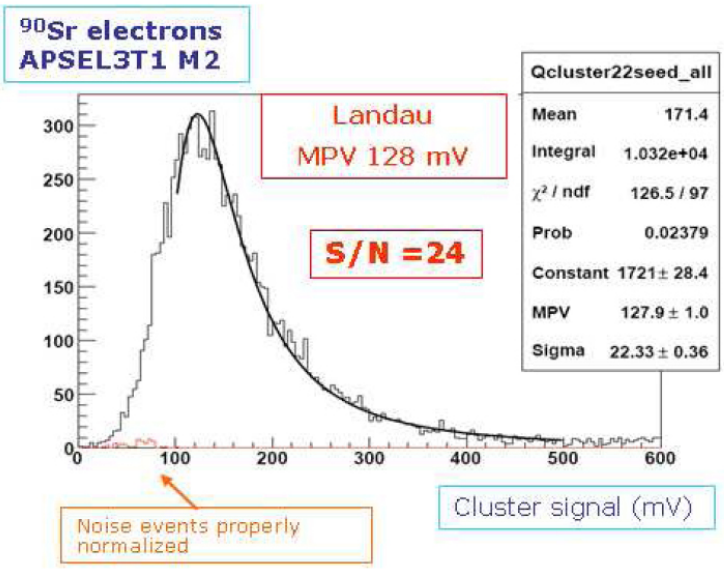

Figure 5.3: **APSEL3T1 cluster signal of the 3x3 pixel matrix**. Beta radiation detected from a ^{90}Sr source.

5.2 The APSEL4D chip

Due to the good results achieved along the first three series of chip APSEL, the collaboration went on in the characterization of a DNW MAPS device with a greater granularity and a smarter readout logic.

A new data sparsification logic has been developed to be integrated in a wide matrix chip, exploiting the benefits of the technologies achieved in the previous chip series. The readout logic, that has been realized synthesizing a high-level *Hardware Description Language* (HDL), sparsifies hit-data and provides the time-tamp information for the hits. The sparsification logic that interface to the latch of each pixel is realized using standard-cells and, as previously said, it is integrated on the sensor substrate. A key feature of the readout logic is its data driven nature, permitting to use the tracker information as a first level trigger.

5.2.1 The Matrix

APSEL4D features a 4096 square-pixel matrix, 32 row by 128 columns, with a pitch of 50 μm.

The main objective of this project were:

1. To minimize the logical blocks realized with PMOS inside the active area in order to preserve the collection efficiency.
2. To minimize the digital lines crossing the sensor area, this allows a readout scalability to larger matrices and reduces the residual cross-talk effects.
3. To minimize the pixel dead-time by reading and resetting the hit pixels as soon as possible.

The sparsification and readout logic has been realized outside the sensible area of the matrix. It will be discussed in more details in the next subsection.

A picture of the APSEL4D chip bonded on the carrier module is presented in Fig. 5.4.

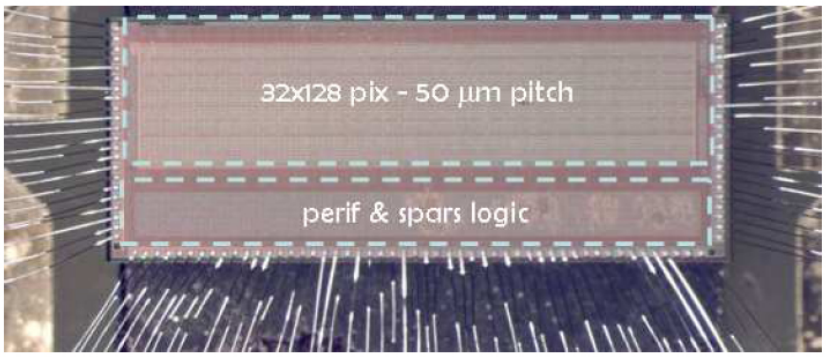

Figure 5.4: **APSEL4D chip bonded on carrier**. The 32×128 matrix is situated on top, while the read-out logic is located in the bottom part of the chip.

In order to minimize the digital lines crossing the active area, the matrix is organized in Macro Pixels (MP), square groups of 4×4 pixels as it is shown in Fig. 5.5 (from [11]). Each MP has only two private lines for a point-to-point connection to the sparsification logic. One is used to indicate that in the MP at least one pixel has been hit, and the second line is used to freeze the whole MP until the hits have been read out.

Each pixel has been realized following the APSEL3T1 front-end flavor (transconductor in the shaper) and the layout is shown in Fig. 5.6.

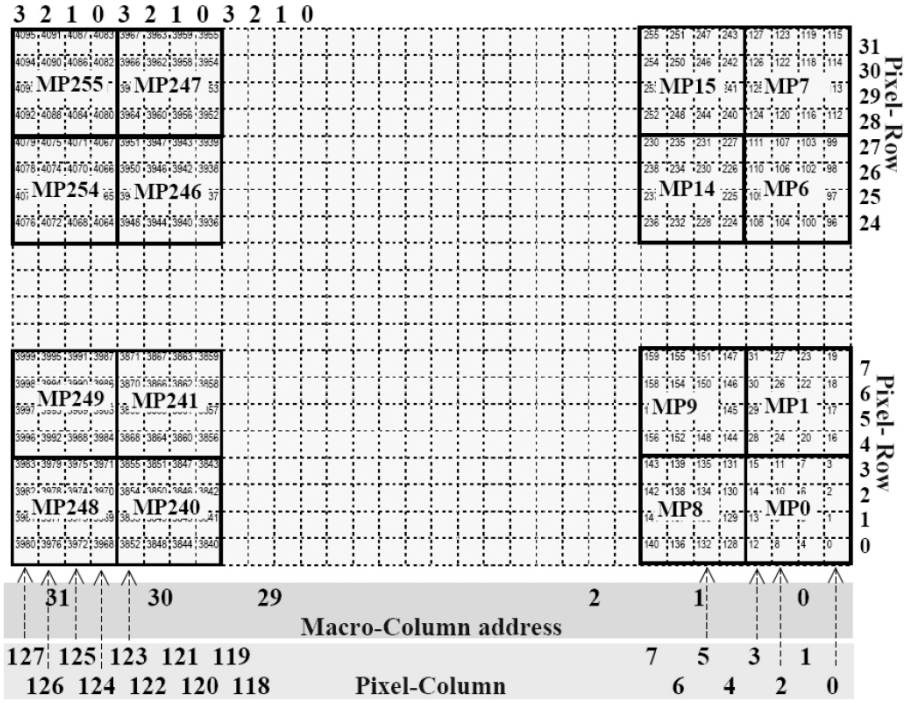

Figure 5.5: **APSEL4D matrix layout and subdivision**. 4096 pixels, 32 rows × 128 columns and 256 Macro Pixels

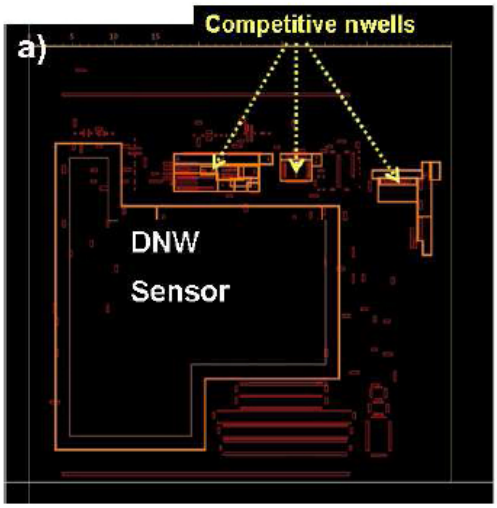

Figure 5.6: **Pixel layout**. The collecting electrode is marked as DNW Sensor and the competitive N-wells are the CMOS N-type implantations.

The pixel matrix has been realized with a full custom design and layout, in opposition to the readout logic which was developed by synthesized standard cells. During the CAD design of the chip a manual placement of the custom matrix was required to interconnect it to the digital logic block.

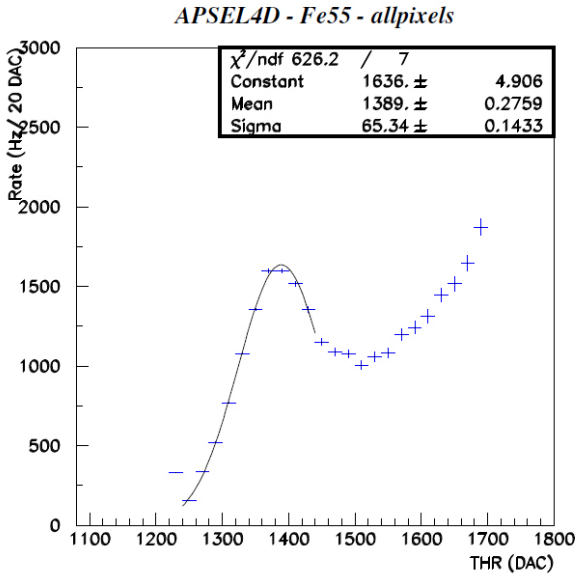

Figure 5.7: **APSEL4D absolute calibration with a ^{55}Fe source**. The total contribution of all pixels has been summed up. The graph is a function of the discriminator threshold.

A first characterization of the chip was realized with the 5.9 KeV peak of a ^{55}Fe source. Since no analog information is available, the photo-peak is reconstructed from the differential rate as a function of the discriminator threshold. With this technique an average gain of 890 mV/fC has been measured with a typical dispersion of about 6% inside the matrix. In Fig. 5.7 is shown the ^{55}Fe calibration peak, summing up the contribution of all the pixel of a matrix.

Noise measurement and evaluation of the threshold dispersion have been performed on the 32×128 pixel matrix measuring the background hit rate as a function of the discriminator threshold. With a fit to the turn-on curve an equivalent noise charge of about 75 e^- (10.5 mV) has been evaluated.

5.2.2 The readout logic

Sparsification and readout logic has been synthesized using VHDL (*Very high speed Hardware Description Language*) in order to provide high-level smart procedures. The gate-level design obtained after synthesis has then been implemented using standard-cell libraries provided by the STM CMOS 0.13 μm technology. After the computer aided layout, a manual routing operation allowed to connect the control lines of the digital logic to the full custom pixel matrix.

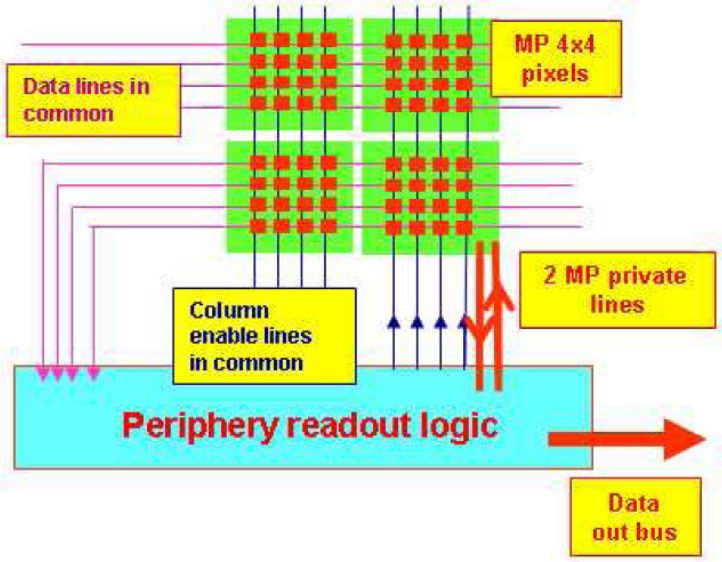

Figure 5.8: **APSEL4D matrix/logic interconnection signals**. 32bit-wide row readout bus, 128 column enable signals and 2 private lines for each MP, one for the hit interrupt and one for the freeze command.

The readout system works on three main clocks: the readout clock (*RDclk*, designed to run at 100 MHz), the BC clock (typically 5 MHz), and the *Slow Control Interface* clock. Simulations indicate that the readout system can cope with an average hit rate up to 100 Mhit $s^{-1}cm^{-2}$ if a master clock of 80 MHz is used, while maintaining an overall efficiency over 99%.

The readout system uses the following signal infrastructure (see 5.8): 256 private hit-lines connect each MP to the digital logic in order to indicate if one (or more) of their 16 pixels get hit. 256 more private lines allow to freeze independently each MP. Moreover a 32 bit column-

wide common data bus is used for the readout operations.

When a pixel goes over threshold, the associated MP's hit-line gets fired. Other pixels within the same MP can still be fired before the arrival of a BC clock rising edge. When a BC edge arrives, the digital logic freezes the status of all the MPs that have been fired; this means that the hit/not hit status of their 16 pixels can change no more, until the next readout and reset phase.

The readout phase take place in parallel, one pixel column per *RDclk* cycle. A 32-bit row bus connects the desired pixel column to the readout logic that is meant to sparsify the hits and to associate a time label to them.

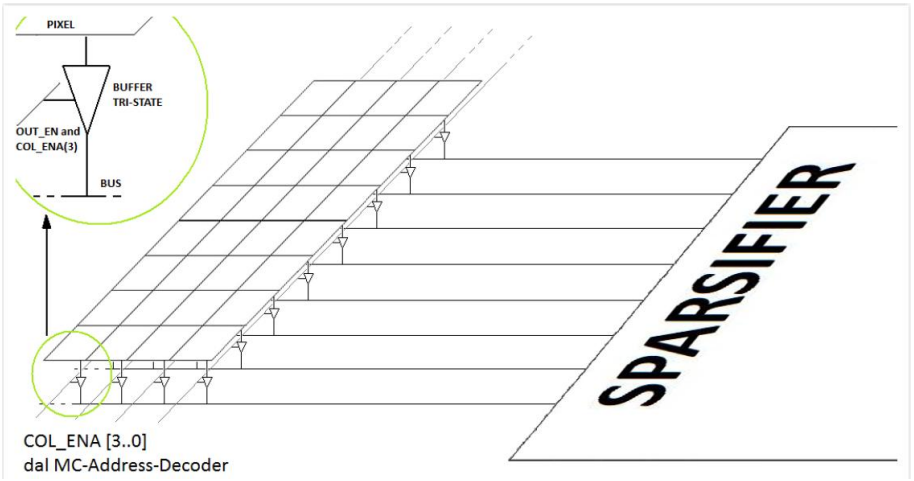

Figure 5.9: **Row readout bus architecture**. 32bit-wide bus for the readout of the latches of a pixel column.

Each column of pixels is provided with a global enable signal, decoded by the readout logic in order to have only one column at a time driving the common row bus as shown in Fig. 5.9. The redout logic enables sequentially the 4 columns of a macro-column if at least one MP of them is fired, reading one column per *RDclk* cycle. During the fifth *RDclk* cycle all the pixel latches of the macro-column (4×32 pixels) are reset. In principle thus, a fully fired matrix can be read in 128 (column read) + 32 (MP reset) *RDclk* cycles.

Once a hit pixel is read, it is sparsified and time-labeled; these spatial and temporal coordinates are then stored in a 20-bit word whose structure is showed in Tab. 5.1. A set of hit buffers (called *barrels*) are implemented inside the chip digital logic to retain a maximum of 160

hits while they are sent over the *data out* bus. This is the chip data output port which has a bandwidth of 1 hit per *RDclk* cycle.

fields	**bits**	**information**
hit[19:15]	5	*Pixel row* ($0 \rightarrow 31$)
hit[14:13]	2	*MP column* ($0 \rightarrow 3$)
hit[12:8]	5	*Macro Column* ($0 \rightarrow 31$)
hit[7:0]	8	*Time Stamp* ($0 \rightarrow 255$ BC edges)

Table 5.1: **APSEL4D hit format**. *MP column* is the relative address of a pixel column within a MP; *Macro Column* is the global address of a MP column. Time stamp is a temporal label associated to a counter of BC-clock edges

Now let's have a deeper look at the hit format. The row address is univocal, individuated in the *Pixel Row* field by an number from 0 to 31 that follows the scheme of Fig. 5.5. The global column address, instead, must be calculated with the following formula:

$$\text{Global pixel column} = \textit{Macro Column} * 4 + \textit{MP column}$$

The sparsifier logic is modular, which means that the 32-bit common row bus is analyzed by four separate sparsifier, each one working on 8 rows only. Every sparsifier has its own private barrel as shown in Fig. 5.10. The *Sparsifier-OUT* module is responsible then for the queuing of hits into the *Barrel Final*.

The data-driven architecture pushes out the hits from the APSEL4D chip on a 21-bit parallel bus, in which the 21st bit is a data-valid flag.

Several additional features are present in the digital logic, included to provide on-chip debugging capability and slow control communication.

A full size dummy matrix has been realized within the digital logic in the form of a 4096 array of latches remotely configurable via the slow control interface. It was meant to test the readout capability of the chip without the interaction with the real matrix. When the chip is configured, it can be started in real or dummy mode connecting the readout logic to the real sensor matrix, or to the dummy digital matrix.

A pixel kill-mask feature is implemented in the chip as well, mask patterns can be loaded into the dedicated registers via slow control. In details, it is possible to mask entire rows or single MPs. This feature is

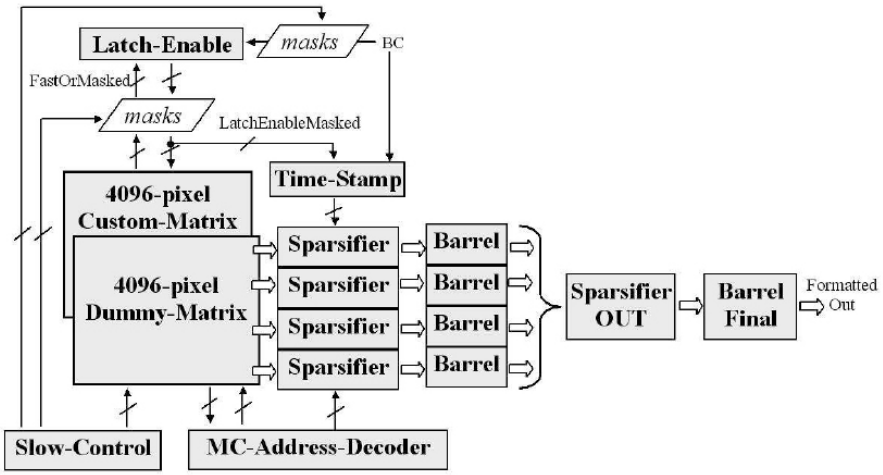

Figure 5.10: **APSEL4D readout logic**. The matrices are connected to the sparsifiers that store the hits into the relative barrels. Each sparsifier controls a group of 8 rows. The MC-Address-Decoder is responsible for the readout loop over the fired macro-columns. The Slow Control Interface manage the configuration and monitoring of the chip.

meant to exclude noisy pixels that waste readout time and transmission bandwidth.

The *Slow Control Interface* operates on the 8-bit wide SC data bus, it is synchronous with the SCclock, and receives commands on a 3-bit wide bus called SCmode. The instructions on the SCmode bus can address single registers or initialize sequential loads of bytes in an array of shift registers.

The SCmode coding `001` individuates, for instance, the loading instruction of the 256+32 masking bits. In this case, on each SCclock edge, 8 bits a time are loaded into the mask buffer, starting from the row masking and following the order shown in Fig. 5.11.

In a similar way the hit pattern is loaded into the dummy matrix with the SCmode code `000`. In this case the whole operation is quite long and it takes 4096/8 = 512 SCclock cycles. Once the pattern is loaded, the *Dummy_actual* register must be set high with the SCmode = `010`. The external dedicated signal *Apply_hit* is used to make the hits visible to the readout logic once the chip is in run mode.

Once the chip is configured, it can be put in run mode with SCmode = `111`. If the logic is connected to the real matrix, there will be a certain

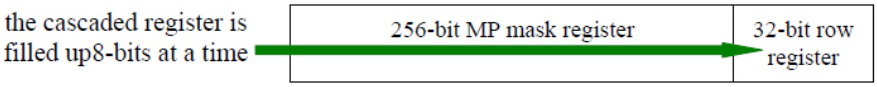

Figure 5.11: **APSEL4D mask shift register.** The first byte received corresponds to the mask of the first 8 rows if the whole mask pattern is successively loaded. Successive bytes individuate the mask patterns for the MPs as numbered in Fig. 5.5.

data rate at the output, depending on the analog threshold and on the signals at the sensor cells. If the dummy matrix is set, instead, one should expect at the output the same hits that were previously loaded.

The whole digital logic can be reset with an active-low dedicated pin, but a *soft_reset* can also be sent with a SCmode = `100` in order to set the BC time counter to 0.

Chapter 6

The Beam-Test

The performance characteristics of the APSEL4D chip were investigated with a beam-test at the CERN accelerator facilities. A strip telescope has been used for the resolution and efficiency studies and a dedicated Data Acquisition System has been developed and realized.

The Beam-Test started at the beginning of September 2008, and ended in about 20 days. The experimental area was situated at the T9 level of the Proton Synchrotron accelerator, providing protons at 12 GeV with a particle density of about 30K particles per bunch. Each bunch was $\sim 1.5\ s$ long and the line was operated at about one spill per minute.

In this chapter will be given a description of the Beam-Test telescope setup and of the DAQ system.

6.1 The Telescope

A silicon strip telescope was installed for the tracking of particles impinging on the MAPS sensor. The DUT (*Device Under Test*) was situated between the two couples of double sided silicon strip detectors constituting the telescope.

The strip used are made with thinned silicon in order to keep low the material budget. This was meant to guarantee a low particle absorption and, more important, a low multiple scattering probability. Multiple scattering is in deed an undesired effect as it can worsen the results about the matrix resolution performance. A scheme of the telescope setup is given in Fig. 6.1.

Every silicon strip sensor is bonded on a hybrid module with three readout chips called FSSR2 (*Fermilab Silicon Strip Readout*) each one acquiring 128 channels, for a total of 384 strips per module. The FSSR2

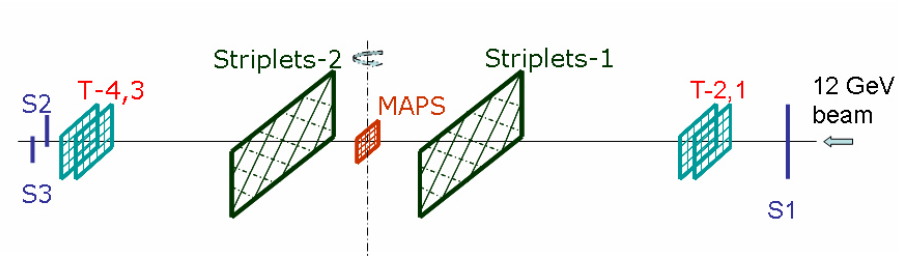

Figure 6.1: **Telescope setup scheme**. 4 layers of double sided silicon strip made up the triggering telescope (T1,2,3,4). The MAPS layer is inserted in the middle to investigate the efficiency and the resolution of the APSEL4D pixel matrix. Two layers of silicon striplets were also included as DUT for characterization but they will not be discussed in this thesis. Scintillators at the two ends were used for beam monitoring.

technology is a TSMC 0.25 μm CMOS with enclosed NMOS for radiation tolerance.

This chip was realized for the Forward Silicon Tracker of the BTeV experiment that was meant to operate at the Tevatron accelerator of Fermilab [45]. A scheme of the analog front-end architecture of a channel is shown in Fig. 6.2.

The chip is a mixed-signal integrated circuit as it integrates the readout CMOS electronics as well. The architecture is self-triggered with an analog storage and virtually acts as the sparsification logic of a pixel matrix (Pseudo-Pixel architecture [26]). The FSSR2 architecture is a modified version of the FPIX2 chip, the readout chip for the BTeV pixel detector. This allowed a significant simplification in the DAQ system, as the two kind of front-ends (MAPS and strips) produced a very similar data-driven flux of data that could be managed in a more uniform way.

The power dissipation of an FSSR2 chip is less than 4 mW per acquired channel. The integrated digital logic features a BCO counter for time labeling of the events, a programming interface for the slow control operations and a set of programmable registers. The codified data words are then packed and serialized directly at the hybrid level.

Each hybrid module is then connected to an *Interface Card* with a multipolar differential flat-cable carrying the data busses and the control signals. Every card, which is mounted in a rack of the experimental area, interconnects the two flat cables (the P and N side) of a strip detector to a set of LVDS transceivers. The role of the transceivers is to

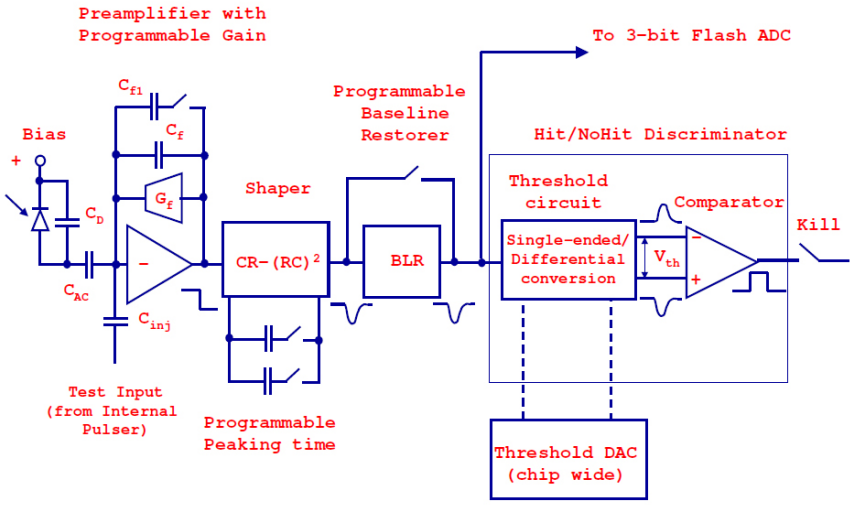

Figure 6.2: **FSSR2 chip analog front-end**. It integrates a preamplifier with programmable gain, a shaper and a baseline restorer that feed a hit/not hit discriminator. A strip is considered fired if a global programmable threshold is crossed.

drive the communications towards and from the DAQ system housed in the counting room. The DAQ architecture and logic will be discussed in details in the next section.

On the Interface Card are also present the low voltage power lines to supply the 6 readout chips (3 for each side).

Each double-sided strip module was incorporated inside a metal box that protects the delicate silicon detector. Then it incorporates a small PCB for the electronic connections and the taps for the air-flux cooling. All the telescope layers were properly chilled with a cooling system based on low-pressurized nitrogen and dry-air, pumped directly on the sensitive silicon areas and on the readout chips.

In Fig. 6.3 there is a picture of the detector experimental setup.

The APSEL4D chip is integrated on the DUT pedestal within its test board. The bare chip is bonded on a small carrier printed circuit board, that can be plugged and unplugged easily through strip connectors to the *APSEL4D Test Board*. In this way several chips could be tested on beam with a simple exchange of carriers.

The APSEL test board was realized both for in-lab test purposes and beam-tests. For this reason there is a hole in the PCB right be-

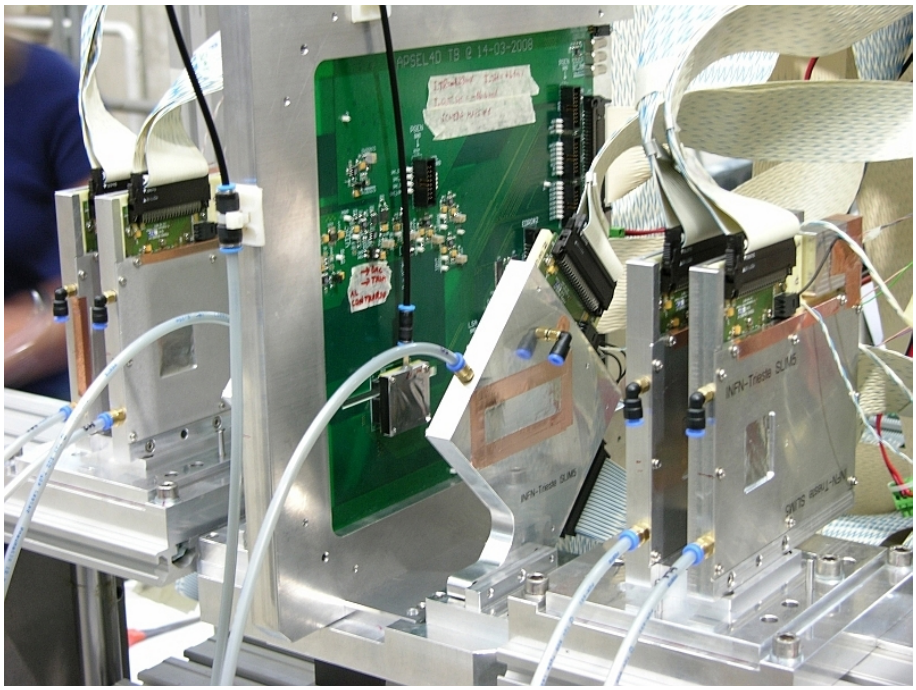

Figure 6.3: **Integrated telescope**. At the two ends there are two strip modules, each one double sided. On the DUT pedestal are mounted the striplets at 45° xy angle and the APSEL4D Test Board within the metallic retention frame.

neath the chip carrier to reduce the material budget on the beam. Few microns of aluminum shield the bare silicon surfaces of APSEL4D and of the strip sensors as well, but the calculated contribution to the production of multiple scattering events is negligible.

The LVDS transceivers are integrated on the test board and so is the Digital to Analog Converter for the remote setting of the analog threshold. Transmission lines are then driven directly on board, and the low voltage power lines are integrated as well, nevertheless a passive *Adapter Card*, similar to the Interface Card for strip modules, is used and mounted in the interconnection rack. This card is meant to match the signals coming out of the test board on 3, 2.5m-long, flat-cables into two standard halogen-free 68-pin SCSI shielded cables. These 30 meter long cables, which are used both for MAPS and strip, run from the experimental area to the counting room.

All the mechanical structure is sustained by a robotized table, customized for the needs with Bosch-Rexroth aluminum framing modules. Every strip detector module is mounted on a micro-metric screw, for the x and y fine adjustments. Torsion on the screws is provided by remotely-operated electrical step-motors. In addition he DUT demonstrator support, where MAPS and striplets are hold, can also be operated to adjust the angle θ between x and z axis. A control server for the step-motors was operating in the experimental area while a LabView interface application run in the the control room, to remotely operate the layers motion. This application makes also available the coordinates of the telescope to the DAQ system.

6.2 The DAQ System

The data acquisition system comprehend all the hardware, the firmware and the software that was meant to acquire data from the front-end modules, and to initialize and run the telescope electronics as well. The the front-end data are stored in a hard drive together with the telescope configuration settings and the environmental parameters as well. The information comprehend the raw and fine position adjustments of the robotized table, the voltage thresholds imposed on the front-end chips, the setup of the trigger and so on.

The heart of the DAQ hardware is a couple of 9U VME boards, called EDRO (*Event Dispatch and Read Out*), whose role is to receive the digital hits coming from the 30-meter LVDS cables, and to build up the events: structured data blocks containing all the hits occurred within a certain temporal window. The two boards work in master-

slave mode, the master board perform the trigger while the slave receive trigger information from the former. The master-slave assignment can be done by a software configuration at start-up.

Once the events are built, they are sent to the DAQ PC on a high-throughput optical link. Here the events are finally stored in a hard drive and the run configuration is recorded in a dedicated data-base. The Run Control program on the DAQ PC coordinates also the start of each run, performing the configuration setup.

The configuration of the system is performed via VME BUS exploiting a VME CPU installed in the crate. Connection with the DAQ PC is provided by an Ethernet link.

A scheme of the DAQ system hardware is showed in Fig. 6.4.

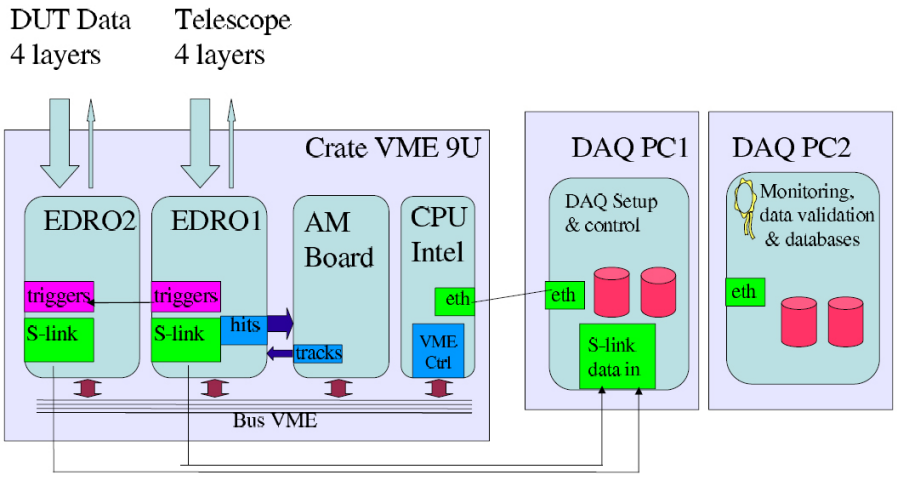

Figure 6.4: **The DAQ System scheme**. The front-end hit streams are conveyed to the EDRO boards that trigger and build the events. The Intel CPU is used for VME read/write bus operations in order to perform slow control operations. Fast data transmissions are implemented over two CERN standard S-link optical interfaces. Associative Memory board add to this system a powerful triggering unit based on track pattern recognition.

6.2.1 The EDRO Boards

The EDRO (*Event Dispatch and Read-Out*) board is a 9U VME board. It is a mother-board holding the 5 mezzanine cards listed below:

- The core mezzanine, integrating a high-end Stratix II FPGA (ES2C130) with 1508 pins designed for the CMS experiment.

There runs the firmware implementing the internal triggering logic and the event builder. In case of external triggering it interfaces towards the triggering element (scintillators, associative memory or master-EDRO).

- Two EPMC (*EDRO Programmable Mezzanine Card*) with a 484 pins Cyclone II FPGA each (EP2C35). These cards provide the digital interface towards and from the front-end readout chips. The firmware on these cards is meant to provide a common data interface to the EDRO core for both strip and MAPS sensors. They also implement all the front-end dependent routines, like initializations, thresholds settings etc. by the use of a simple set of instructions. In a word, they make the front-end architecture almost transparent to the EDRO core. On each of them a front-end connector panel is mounted at 90° to interface the telescope modules through the LVDS cables.

- The HOLA S-link card, a 1.3 Gbps optical link, that has been developed at CERN for fast FIFO dequeuing.

- The TTC-RQ mezzanine card, developed for the LHC triggering and bunch crossing clocking infrastructure. It has been integrated for the 40 MHz clock generation and the EDRO-EDRO synchronization.

A high-bandwidth backbone interconnection is also present on the board dedicated to the transmission of hits towards the pattern bank of the *Associative Memory* board (described later on).

On the EDRO main board another small FPGA is present, dedicated to the management of the VME BUS communication with the logic of the EDRO board. This interface is used for the board registry configuration and monitoring. To each board a different base address is assigned, in our case `0x400` and `0x200`, while a relative 12-bit address individuate a specific register. The combination of base and relative address univocally individuates a register of the system. Two categories of registers are present, the read/write and the read-only registers.

All the registers addressable on the VME BUS physically reside on different devices: for example 32 of these registers are implemented on each EPMC FPGA while the others are located on the EDRO core.

A picture of the EDRO board is given in Fig. 6.5.

Mechanical rigidity is provided by a metallic shielding panel which is held by the EDRO PCB and it is screwed to the 90-degree front-end interfaces. The junction of these interfaces to the EDRO board

is a delicate point as it should resist to the stress of plugging and unplugging the LVDS cables, and it must bear the weight of 4 metallic SCSI connectors.

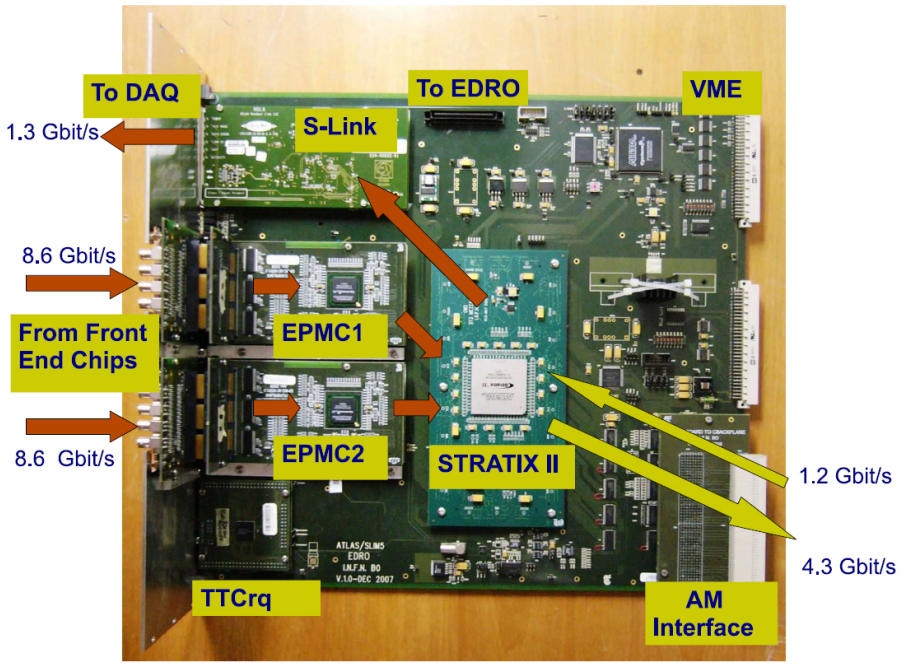

Figure 6.5: **The EDRO board**. The board core is the Stratix II. The two EPMC cards interface to front-ends.

The EPMC boards

The DAQ system is interfaced to the front-end by two *EDRO Programmable Mezzanine Cards* (EPMC). Each EPMC mounts a front-panel board at 90° which integrates 4 SCSI front-end connectors and 4 LEMO plugs. Through the 4 SCSI connectors each EPMC can drive 4 telescope strip layers (12 FSSR2 chips) or 2 APSEL4D chips. For debug purposes a dedicated R/W register in the FPGA selects which signal of a predefined set has to be redirected on the LEMO plugs. In the front-panel board is mounted also a JTAG connector for the programming of the FPGA once the main board connector is no more accessible due to the presence of the metallic shield panel. A set of frontal LEDs provides an overview of the system status at a glance.

Basically, the EPMC firmware running on the Cyclone II FPGA, provides an intermediate interface between the front-end readout chips and the EDRO core. In this way it is possible for the core logic to manage the initialization and the acquisition process of two different kind of front-ends with a standardized 16-bit register access port. By means of this, the core simply accesses in read or write mode a set of VME registers residing on the EPMC boards to deal with the front-ends. In this way the knowledge of what use should be done of the registers is relayed to the DAQ software and to the EPMC firmware.

The DAQ software, for the front-end configuration, addresses via VME the registers of the EPMC to write instructions and parameters; the firmware of the Cyclone II FPGA decodes and executes this sequence of instructions, communicating in a suitable way with the front-end chips.

The instruction sets (at EPMC level), for the control of the two kind of chips, are obviously different and, therefore, in contrast with the common firmware that runs on both the StratixII cores, the EPMC CycloneII FPGAs are loaded with dedicated configuration files. Depending on the front-end they are supposed to drive, the *EPMC_MAPS* or the *EPMC_STRIP* firmware will be loaded. Each one implements the control logic for one architecture only, but both are provided with the same *EDRO Register Interface* and the same *EDRO Data Interface*.

The MAPS interface will be described in more details as it has been the subject of my contribution in this collaboration. A schematic view of the *EPMC_MAPS* firmware is shown in Fig. 6.6.

Imparting a command to a front-end chip appears to the DAQ PC as standard VME R/W operation on a specific register of the *Register File* component. A set of instructions has been developed to implement all the basic slow control operations on the chips. More complex automated routines have been implemented in firmware as well, preventing long and slow transactions on the VME BUS. For example, in the specific case of *EPMC_MAPS* firmware, a full sensor calibration routine was implemented; the user is supposed only to specify the calibration-run parameters and then to start the process.

The register appointed to be the *Instruction Register* is the one with index number `0x06`. Refer to Tab. 6.1 as a reference of all the accessible registers on the *EDRO Register Interface*. The specific use if these registers will be explained during the firmware description.

Each time a write operation is performed on the instruction register, the finite state machine of the *APSEL Interface* decodes the instruction

VME add.	Type	Idx	Name
0x200980	r/w	0x00	Calib. Config. A
0x200984	r/w	0x01	Calib. Config. B
0x200988	r/w	0x02	Enables
0x20098C	r/w	0x03	BCO Calib. delay
0x200990	r/w	0x04	Monitor selector
0x200994	r/w	0x05	DAC threshold
0x200998	r/w	0x06	APSEL Command
0x20099C	r/w	0x07	data to APSEL LSW
0x2009A0	r/w	0x08	data to APSEL MSW
0x2009A4	r/w	0x09	End-Event timeout
0x2009A8	r/w	0x0A	SC-clock prescaler
0x2009AC	r/w	0x0B	unused
0x2009B0	r/w	0x0C	unused
0x2009B4	r/w	0x0D	unused
0x2009B8	r/w	0x0E	unused
0x2009BC	r/w	0x0F	unused
0x2009C0	r.o.	0x10	Version
0x2009C4	r.o.	0x11	BCO counter LSW
0x2009C8	r.o.	0x12	BCO counter MSW
0x2009CC	r.o.	0x13	Busy spy reg
0x2009D0	r.o.	0x14	PLL lock
0x2009D4	r.o.	0x15	APSEL FSM monitor
0x2009D8	r.o.	0x16	Calib. DAC step
0x2009DC	r.o.	0x17	DATA from APSEL 0
0x2009E0	r.o.	0x18	DATA from APSEL 1
0x2009E4	r.o.	0x19	Last Time Stamp
0x2009E8	r.o.	0x1A	End-Event wrap flags
0x2009EC	r.o.	0x1B	Error codes
0x2009F0	r.o.	0x1C	Dummy matrix pointer 0
0x2009F4	r.o.	0x1D	Mask pointer
0x2009F8	r.o.	0x1E	Dummy matrix pointer 1
0x2009FC	r.o.	0x1F	Rate Monitor

Table 6.1: **EPMC_MAPS register file map**. The VME addresses refer to EDRO 2 - EPMC 2 location, where MAPS chips used to be mounted. Idx is the firmware internal index of the registers.

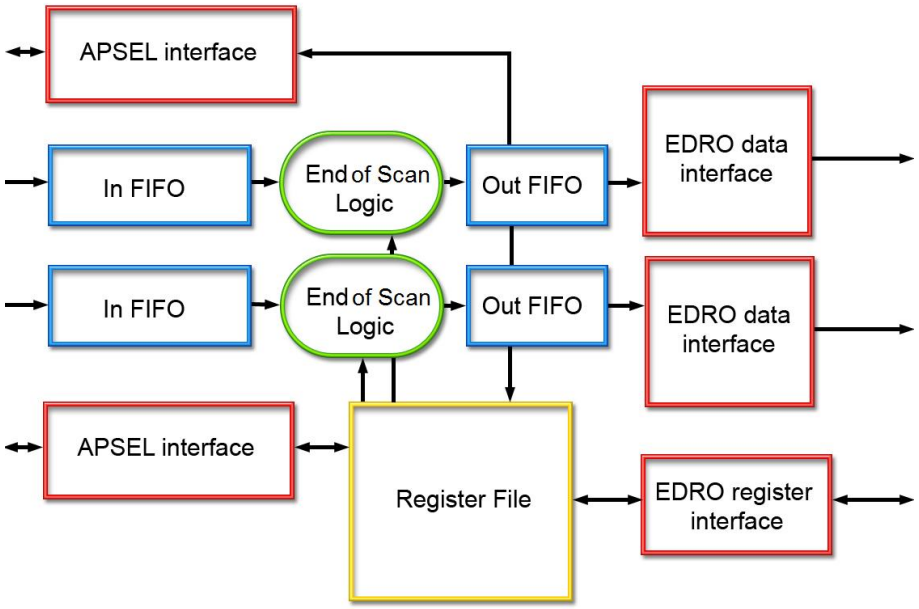

Figure 6.6: **The EPMC firmware scheme**. Since the FPGA must drive two MAPS chips, two *APSEL Interfaces* and two data channels are implemented. The *Register File* instance is shared among the chips and it is accessed in R/W mode by the *EDRO Register Interface*.

and enters an operation cycle. The instruction register is one and hence it is shared among the two front-end chips. The two most significant bits of each instruction encode the destination of the command: `10` refers to chip 1, `00` refers to chip 0. `11` is interpreted as a broadcast command, in this case both the APSEL Interfaces are supposed to perform the requested operation simultaneously.

The list of the defined EPMC instruction set is reported below:

- `0x101` *Load mask* : It shifts into the APSEL4D mask memory the 32 bits stored in the two registers *Data to APSEL* from the LSB of the LSW to the MSB of the MSW. As the APSEL mask memory is 288 bit long (36 bytes), the full masking of the matrix takes 9 successive *Load mask* operations. In the register `0x1D`, bits [5:0] count the number of bytes shifted in the APSEL 0 chip, while bits [11:6] refer to chip 1. In this way it is always possible to know how many bytes have been shifted in.

- `0x102` *Hard reset* : It forwards to the chip a hard reset on the dedicated pin while cycles 10 SC clock pulses at system clock

speed.

- `0x103` *soft reset* : It sends a software reset to APSEL4D which simply resets the time counter register, leaving unaltered the other logic.

- `0x104` *Push dummy* : It shifts in the digital matrix of APSEL4D the 32 bits stored in the two registers *Data to APSEL* from the LSB of the LSW to the MSB of the MSW. As the full digital matrix is 512 bytes long, the full dummy matrix is loaded with 128 *push dummy* operations. Also in this case pointer monitors are present; the number of shifted bytes is stored for chip 0 and chip 1 respectively in the r.o. registers `0x1C` and `0x1E`.

- `0x105` *BCO read* : It reads back the value running in the APSEL4D BC counter. The read value can be found in the LSB of r.o. registers `0x17` and `0x18` respectively for chip 0 and chip 1.

- `0x106` *Set dummy reg.*: It configures the APSEL4D to work with the the dummy matrix. (default option of the chip).

- `0x107` *Reset dummy reg.*: It configures the APSEL4D to work with the the real sensor matrix.

- `0x108` *DM bit rolling* : It forwards this command to the APSEL chip that directly implements this function inside itself. It basically gives a byte shift to the 4096-bit memory of the dummy matrix. It provides the user of a quick hit-rearranging feature, helpful for fast debugging operations.

- `0x109` *Apply hit* : Another basic APSEL4D command that is forwarded to the chip. When working in dummy mode it makes the hits of the digital matrix visible to the readout logic. After this command the previously loaded hits are supposed to be read and sent back over the readout chain.

- `0x10A` *Go to Run Mode* : Macro command. It formerly performs a matrix selection (equivalent to a *Set* or *Reset dummy reg.* operation), hence it turns on the *RUNisOn* system flag and puts the APSEL4D chip in run mode. Now the FSM of the APSEL Interface is in the RUN state and can accept only two commands from the DAQ system: *Apply hit* and *Stop run.*

- `0x10B` *Stop Run* : It turns the *RUNisOn* flag off and returns the APSEL FSM to the *IDLE* state, awaiting for other commands.

When the *RUNisOn* flag is off no more data are accepted in the FIFOs.

- `0x10C` *Go to Calib. Mode* : Macro command. It starts the full calibration process. The user should set before the correct parameters in the following three dedicated registers:

Register Idx	Range	Field name
`0x00`	[15:8]	Threshold DAC steps
	[7:0]	DAC max end (the 8 MSbits)
`0x01`	[15:8]	BCO cycles
	[7:0]	DAC min end (the 8 MSbits)
`0x01`	[15:0]	BCO prescaler

A calibration consists basically of a threshold sweep, from a DAC minimum value to a DAC maximum value with a certain stepping. The DAC used for threshold voltage generation is an Analog Device AD7390 with 12 bits of resolution (4096 DAC counts). The min. or max. value, stored in `0x00` and `0x01` registers, represents the 8 most significant bits of the whole 12-bit unsigned-integer value. The sweep step is instead encoded at full resolution and thus it can represent from 1 to 255 DAC counts.

The threshold sweep is then subdivided into a column-based sweep. This is not due by real calibration needs but it derives from a limit imposed by the the DAQ system. When the threshold sweep goes through the range of values where all the pixels get always fired, the whole matrix would produce, for each BC clock, a block of 4096 hits. Such a big event would never reach the acquisition storage as the DAQ system was tuned to give best performance in normal conditions, which are far from these. The maximum number of hits referring to a single BC clock period (same time stamp) that are allowed to enter the event builder of the EDRO core is 128. 128 pixels correspond to a full-fired macro-column and hence, in order to work with less stringent constraints, each threshold step is subdivided into a sweep of macro-column halves (2 pixel columns). The double column sweep is realized by the EPMC firmware that rejects those hits whose coordinates fall outside the investigated area.

Finally, for each macro-column half, a certain number of BC clock pulses is sent to the front end in order to increase the statistics. This number is encoded as an unsigned integer in the *BCO cycles* field of reg `0x01`.

The APSEL Interface FSM can be programmed to forward only one BC pulse every N BC received, in order to be sure that the readout chain has sufficient time to deliver the whole event without going into a busy state. Letting the system go into a busy state implies hit losses that are not allowed during calibration. The *BCO prescaler* field individuates, once again as an unsigned integer, the number of system BC edges that are skipped between each BC pulse sent to the front-end (a value of 127 in the prescaler means one pulse every 128).

With this calibration mode a *hard reset* is sent to the front-end before each new DAC cycle, and a *soft reset* between each column step. When the threshold step exceeds the DAC max value the calibration process ends.

- `0x10D` *Go to Calib. Mode 2*: This procedure is identical to the previous one, except for the soft reset which is not sent between each double column sweep.

- `0x10E` *Shift EPMC pix mask*: It is a firmware implemented feature that is used to provide additional fine-grain masking. It is in addition to that implemented on the APSLE4D chip which kills entire macro-pixels or whole rows. It must be said that this additional feature does not avoid the congestion of the APSEL4D readout logic due to super-noisy pixel since it is implemented outside the chip. It is basically a programmable filter located at the input of the incoming data FIFO killing all those hits that are found on a black list. Even if it can't completely replace the on-chip masking, used for super-noisy pixels (occupancy above 80%), on the other hand it was very effective with those medium and low noisy pixels that can't saturate the APSEL4D readout but waste DAQ transmission bandwidth and memory resources. The effectiveness of this feature is evident as a fine masking avoids an unnecessary pixel overkill that drastically lowers the sensor efficiency.

  Operatively, for each front-end chip was allocated in the EPMC firmware a 12-bit wide shift register that can store up to 20 undesired coordinates. When this command is executed, the 12 least-significant bits of the *Data To APSEL LSW* register are shifted into the black list array. Each incoming hit is then compared to the whole list in a single system clock cycle, and if any match is found the hit is rejected.

We discussed the control and monitoring of the readout chips and then we saw how these operations are transparent to the EDRO core which needs no particular information about the kind of connected front-ends. Now we will see that it is almost the same for what concerns the hit collection and delivering. In both architectures the hits are coded by the EPMC in a 24-bit word. The leading 8 bits of the word are reserved to the time-stamp field. The main difference in the hit format of the two architectures concerns the spatial coordinates as they refer to different topologies. Anyway this incoherence is not an issue for the EDRO core since the spatial information is not processed nor for the event building nor for the trigger algorithm. The event builder and the trigger are based only on the hit time tags which are found in the same field for both data structures.

The EDRO core receives from each EPMC four parallel hit-busses: the *EPMC_STRIP* firmware exploits all of them since each stream corresponds to a telescope layer. The *EPMC_MAPS*, instead, uses only two busses as it can drive only two front-end chips. The non-used hit busses are thus kept inactive and from the EDRO point of view these are sipmly channels with a null data rate.

Each hit-stream is made of a 24bit-wide bus running at system clock (40 MHz) plus a *Data valid* flag and a *Hold* bit. When *Data valid* is active it means that the EPMC is sending a valid datum on the bus and it must be stored in the core for processing. On the other side the core can halt the transfer by activating the *Hold* line. This simple handshaking is characteristic of a data driven architecture where the data flow is halted only in harming situations where the whole acquisition run can be compromised (e.g. FIFO overflow and so on). The 24 bits of a MAPS hit are encoded as shown in Tab. 6.2.

Field name	**Range**	**Length**	**Use**
Time stamp	[23:16]	8	BC counter
Data parity	[15]	1	parity of the hit
Chip N	[14:13]	2	`00` for chip 0; `01` for chip 1
Pixel row	[12:8]	5	see APSEL4D chapter
MP relative column	[7:6]	2	see APSEL4D chapter
Macro column	[5:1]	5	see APSEL4D chapter
End of Scan word	[0]	1	`0` = not an EoS word

Table 6.2: **MAPS hit format**.

Another particular feature of the firmware which runs in the FPGA of the EPMC is the implementation of an *End of Scan Logic*. The

concept of scan is associated, for the pixel chip, to a whole readout cycle of the matrix. The EPMC firmware is then provided with a processing logic that analyzes the incoming hits and interposes in the outgoing stream a key word, called *End os Scan Word* each time it infers that no more hits, related to the previous sweep cycle, are going to be sent.

This feature must be added foreseeing to use MAPS data with the *Associative Memories*, a triggering unit that will be described later on. Due to the presence of internal parallel barrels in the architecture of the APSEL4D chip, hits belonging to different sweeps can be overlapped on the output stream. The Associative Memories triggers on single scan patterns and therefore they must be informed when a scan is completed.

The End of Scan Logic, in the *EPMC_MAPS* firmware, is implemented on the observation of the column index of the outgoing hits for each set of rows corresponding to a primary barrel shifter of APSEL4D (§ Chapter 5). If a column wrap is observed on all the four corresponding barrels an End of Scan Word is sent over the hit-stream, tagged with a time stamp which is the value of the BC counter at the moment of the E.o.S. individuation. The format of an E.o.S. word is shown in Tab. 6.3. A timer is also implemented to prevent the logic to wait forever. The timeout value is written, during the startup configuration phase, in the register `0x09` and it is expressed in *Processing clock* cycles.

Field name	**Range**	**Length**	**Use**
Time stamp	[23:16]	8	BC counter
Parity	[15]	1	parity of the EE word
Chip N	[14]	1	`0` for chip 0; `1` for chip 1
Timeout	[13:2]	12	value reached by TO counter
End of Scan word	[0]	1	`1` = it is an EoS word

Table 6.3: **MAPS End of Event Word format**. In the Timeout field is reported the value reached by the Time Out Counter when the E.o.S. word was generated. In case this value is less than the superimposed limit it means that a wrap has been individuated on all the corresponding barrels.

The *Processing clock* is one of those used by the *EPMC_MAPS* firmware logic; in the attempt to make it clearer, a list of all the clocks used is reported below with a brief description for each one.

- *System Clock* : It is the synchronous, system-wide 40MHz clock

delivered to the EPMC by the EDRO core. This is the master clock from which all the others are derived (except for the BC clock). The hit bus towards the EDRO core is synchronous on this clock.

- *Processing Clock* : It is a 80 MHz clock used to process the hits in the End of Scan logic in order to speed up the transition of hits towards the EDRO core.

- *BC Clock* : It is basically a clock asynchronous respect to all the others and it is received on a dedicated line from the EDRO. It is the Bunch Crossing clock that usually marks the time of interactions in a collider experiment. It is used to increment the front-end time counter and, in the particular case of the FSSR2 chip, it is shared with the slow control communication interface.

- *Readout Clock* : The entire test beam went on with the choice of a 20 MHz readout clock. This clock is delivered to the front-ends, and it is used to send back the hits. Due to the length of cables (30 m forward, and 30 m backward) a phasing problem arises when the EPMC tries to read the incoming hits. We solved the problem by introducing a set of four read clocks with same frequency but different phases: 0, 90, 180 and 270 degrees. The Read Clock, which is sent forward to the front-end chips, is generated with a null phase by dividing the system clock of a factor 2. Instead, the clock phase to be used on the input FIFOs of EPMC can be chosen by the user. The bit field [13:12] of register `0x02` selects the phase of the FIFOs write clock with the following encodings:

  - `0x0` : 0 deg. phase
  - `0x1` : 90 deg. phase
  - `0x2` : 180 deg. phase
  - `0x3` : 270 deg. phase

  Monitoring the outgoing clock and the incoming hits with an oscilloscope on the EPMC connector, we found that best *setup* and *hold* conditions were achieved using the 270 deg. phase.

- *Slow Control Clock* : It is the clock used by the APSEL Interface to send instructions to the chip and to set the thresholds on the external DAC. Its frequency can be set by the user in a configuration register. The unsigned integer value stored in register `0x0A` is the divisor of the system clock. It was observed that the DAC

interface can be correctly operated with a maximum frequency of 10 MHz, hence the minimum division factor is 4.

Now that we have discussed how data is received, pre-processed and delivered to the EDRO core, we will describe the main acquisition processes performed by the central StratixII FPGA. In particular these are *event building* and *triggering*.

Event building

The event building process has, in first place, the role of sorting the incoming hits which can arrive in a disordered way depending on the readout architecture of the front-end chips. As was showed in the APSEL4D section, there are 4 parallel barrels, each one receiving hits from 8 pixel rows. We have already discussed this characteristic in the EPMC E.o.S. logic, but there the goal was simply to assert if the hits, belonging to a certain scan, were no more going to be sent; no actions were taken in order to time-sort the hits. Referring to the telescope front-end, the hits coming from a single strip are read by three different readout chips and this can cause temporal overlapping as well on the hit-stream.

Every hit is marked with the time stamp during the sparsification within the front-end chips. The time stamp is a modulo-32 counter incrementing on the BCO clock. The *Bunch Crossing clock* derives its name foreseeing the employment of these chips in a collider experiment; it is a global clock that reach all the front-ends and it is used to give the desired time granularity to the events. Hence, in order to build up a coherent event, when the hits reach the EDRO core, they need first to be time-sorted. This is done with a set of circular buffers implemented in the StratixII FPGA block RAM. Each hit stream is associated to a dedicated circular buffer, made up of 32 time slots (5 bits of time stamp). Every BCO slot has three possible states that last as follows (see Fig. 6.7):

- Hit collecting: 16 BCO.
- Trigger waiting: 8 BCO.
- Readout/cleaning: 8 BCO.

During the *hit collection* phase no action is taken, hits are simply stored in the proper time-slot buffer. During the *trigger waiting* slots, events are closed as the end-event timeout has been reached, and they

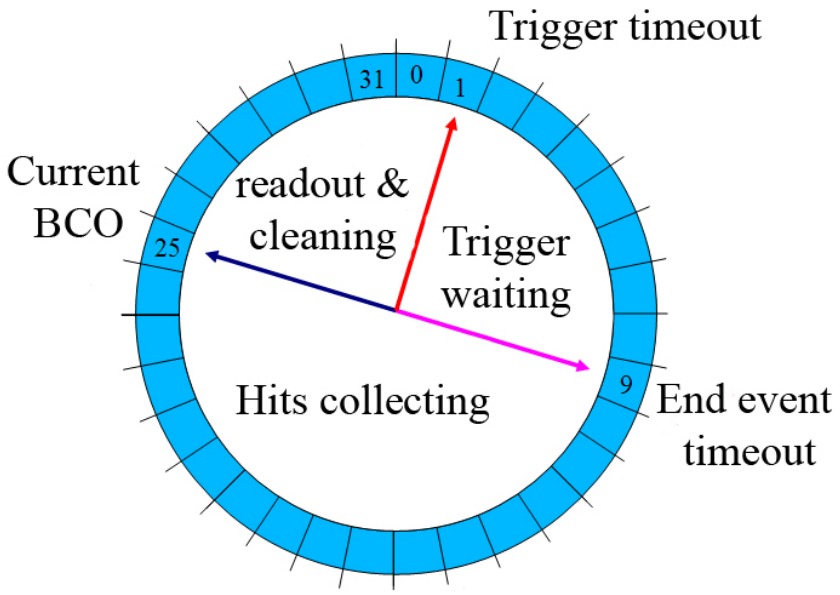

Figure 6.7: **Circular event buffer**. Each hit-stream has an associated circular buffer. BCO increments clockwise.

wait until the trigger logic decides which slots should be acquired. Finally, in the *readout* phase, all the hits of a triggered slot are read out. If no trigger is generated for a certain slot, the whole hit queue is cleared.

Having 8 front-ends connected on each EDRO board, the EDRO core FPGA is provided with 8 circular buffers, and each circular buffer is made up of 32 slots that can store up to 40 hits each. Every hit is encoded with a 24-bit word, thus each circular buffer requires more than 30 Kb of memory. The triggered hit-sets of a buffer are then stored into a local FIFO. Now the hits are time-sorted and wait to be read by the central *event builder* which will fill a central FIFO with the formatted events. All the unused memory of the Stratix II FPGA (4 Mbits) has been allocated for this FIFO in order to maximize the global DAQ rate.

The event building process that formats the events, basically puts together the hits collected with some additional information of the run. Here starts to form the event structure as a field-defined frame of 32-bit words. A schematic of the EDRO core logic is given in Fig. 6.9 and Fig. 6.9.

The inner event building processes runs at 160 MHz, in order to

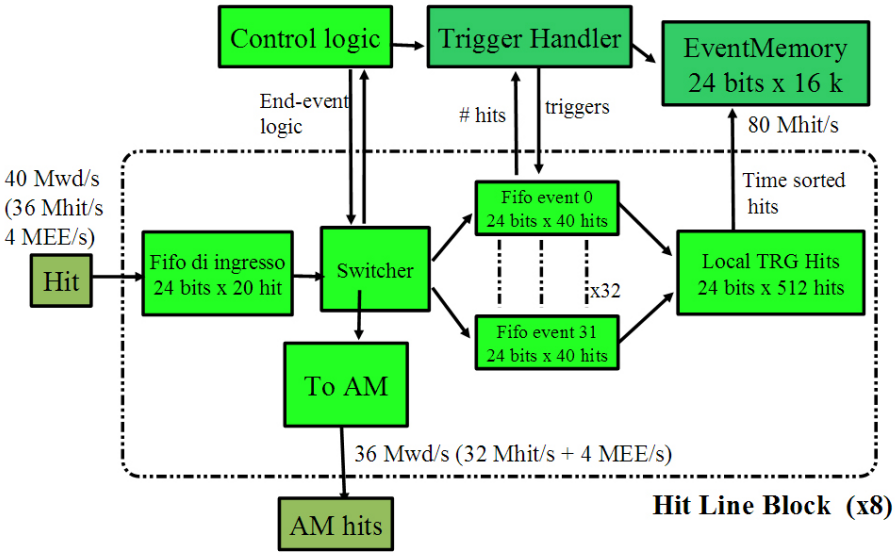

Figure 6.8: **Hit Line Block details**

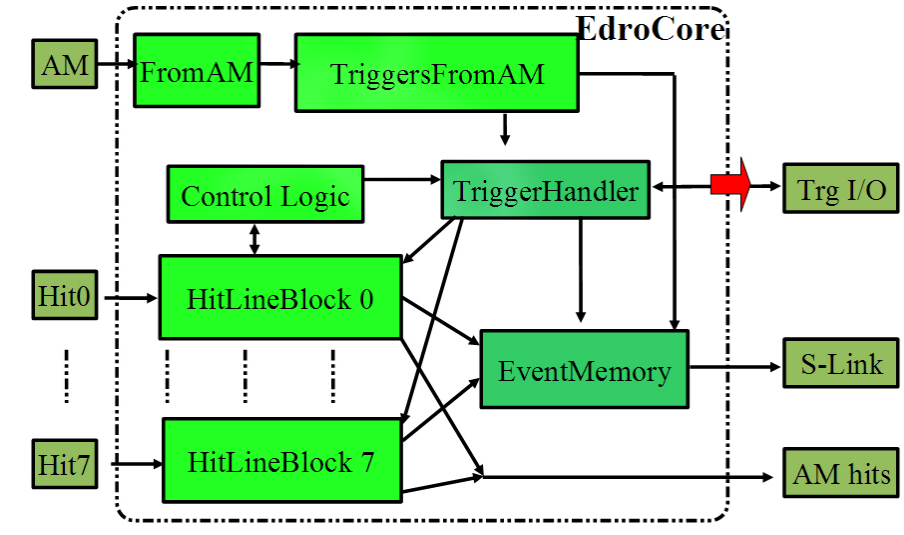

Figure 6.9: **EDRO core logical blocks.**

speed up the data processing, while the EPMC data interface still runs at 40 MHz.

With the optical interface provided by the *S-link*, these event frames are transferred to a PCI board called *FILAR Card*. This board is mounted inside the DAQ PC and makes acquisition data available to the system processes. Must be said that, even if the FILAR card receives data from two independent channels, it collects events that were generated synchronously. Thus the DAQ PC can univocally pair couples of event-halves that present the same global time stamp. The so builded event is then stored in the hard drive as a binary file. A deeper look will be given now at the event structure.

The boxed structure of the stored events is showed in Fig. 6.10 and 6.11. This is the standard output of a C++ program called *EventController* that processed the test beam run 3995. The EventController libraries were meant to provide the decoding algorithms for the binary event structures and the main program is used to print on screen an ASCII report of the analyzed run. With a high verbosity execution of the main program one can get printed on screen the whole run decoded word by word. In each numbered row is decoded a 32-bit word with its binary and hexadecimal representation. In the rightmost column a text comment explicitly decodes the content of each word allowing the user to easily inspect the desired information.

The EventController standard output has been presented in order to explain the structure of the events:

An event starts with a header that basically contains the information related to the current run, for example the run number is decoded in decimal notation in the info column. Each event is then subdivided into two *Read Out Buffers* (ROB). Each ROB contains one *Read Out Data* (ROD) field which is the data frame received from an EDRO core. In the ROD header are encoded several fields of information, between these:

- The format version of the EDRO event frame.
- The EDRO board identification number that generated the ROD.
- The BCO counter at the moment of the event building (BX field).
- The BCO counter when the trigger occurred (BCO field, typically 16 steps forward respect to those contained in the hits).
- The trigger type adopted.

```
53) 1100 1100 0001 0010 0011 0100 1100 1100     cc1234cc       Event header
54) 0000 0000 0000 0000 0000 0000 0111 0011     00000073       +
55) 0000 0000 0000 0000 0000 0000 0000 1011     0000000b       +
56) 0000 0011 0000 0001 0000 0000 0000 0000     03010000       +
57) 0000 0000 0000 0000 0000 0000 0000 0001     00000001       +
58) 0000 0000 0000 0000 0000 0000 0000 0001     00000001       +
59) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       +
60) 0000 0000 0000 0000 0000 0000 0000 0011     00000003       +
61) 0000 0000 0000 0000 0000 1111 1001 1011     00000f9b       + Run Number    3995
62) 0000 0000 0000 0001 0100 0100 0010 0110     00014426       +
63) 0000 0000 0010 0111 0000 1010 1101 0011     00270ad3       +
64) 1101 1101 0001 0010 0011 0100 1101 1101     dd1234dd       + ROB header
65) 0000 0000 0000 0000 0000 0000 0010 0110     00000026       + *
66) 0000 0000 0000 0000 0000 0000 0000 1010     0000000a       + *
67) 0000 0011 0000 0001 0000 0000 0000 0000     03010000       + *
68) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + *
69) 0000 0000 0000 0000 0000 0000 0000 0011     00000003       + *
70) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + *
71) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + *
72) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + *
73) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + *
74) 1110 1110 0001 0010 0011 0100 1110 1110     ee1234ee       + * ROD header
75) 0000 0000 0000 0000 0000 0000 0000 1001     00000009       + * Format version      9
76) 0000 0011 0000 0001 0000 0000 0000 0000     03010000       + *
77) 1110 1101 1010 0000 0011 0000 0000 0000     eda03000       + * Source ID: Slave
78) 0001 1110 1100 1110 1000 1010 1010 0111     1ece8aa7       + * BX: 516852391
79) 0000 0000 0000 0001 0100 0100 0010 0110     00014426       + * Lvl1ID: 82982
80) 0000 0000 0010 0111 0000 1010 1101 0011     00270ad3       + * BCO: 2558675
81) 0000 0000 0000 0001 0000 1010 0000 1101     00010a0d       + * Trigger type: EXT   TS: 13
82) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + * Event type: 0
83) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       + * N Hit Words 0-3:    (1, 1, 1, 1, )
84) 0000 0000 0000 0000 0000 0010 0000 0000     00000200       + * N Hit Words 4-7:    (1, 3, 1, 1, )
85) 1010 0000 1010 0000 0000 0000 0000 1101     a0a0000d       + * Start trigger      0 trks End Trigger
86) 1101 1000 0000 0000 0000 0000 0000 0000     d8000000       + * Hit Block for line 0
87) 1101 1001 0000 0000 0000 0000 0000 0000     d9000000       + * Hit Block for line 1
88) 1101 1010 0000 0000 0000 0000 0000 0000     da000000       + * Hit Block for line 2
89) 1101 1011 0000 0000 0000 0000 0000 0000     db000000       + * Hit Block for line 3
90) 1101 1100 0000 0000 0000 0000 0000 0000     dc000000       + * Hit Block for line 4
91) 1101 1101 0000 0000 0000 0010 0000 0000     dd000200       + * Hit Block for line 5
92) 1101 1101 1100 1101 1111 1111 1111 0010     ddcdfff2       + * Hit Block for line 5
93) 1101 1101 1100 1101 0110 0100 1001 0110     ddcd6496       + * Hit Block for line 5
94) 1101 1110 0000 0000 0000 0000 0000 0000     de000000       + * Hit Block for line 6
95) 1101 1111 0000 0000 0000 0000 0000 0000     df000000       + * Hit Block for line 7
96) 1011 0001 1110 1011 0001 1110 0000 1111     b1eb1e0f       + * End ROD
97) 0000 1111 0001 0001 0100 1111 1101 1110     0f114fde       + * End ROD;  Checksum ok
98) 0000 0000 0000 0000 0000 0000 0000 0000     00000000       +
99) 0000 0000 0000 0000 0000 0000 0000 0001     00000001       +
100) 0000 0000 0000 0000 0000 0000 0000 1111    0000000f       +
101) 0000 0000 0000 0000 0000 0000 0000 0001    00000001       +
102) 1101 1101 0001 0010 0011 0100 1101 1101    dd1234dd       + ROB header
103) 0000 0000 0000 0000 0000 0000 0100 0010    00000042       + *
104) 0000 0000 0000 0000 0000 0000 0000 1010    0000000a       + *
105) 0000 0011 0000 0001 0000 0000 0000 0000    03010000       + *
106) 0000 0000 0000 0000 0000 0000 0000 0001    00000001       + *
107) 0000 0000 0000 0000 0000 0000 0000 0011    00000003       + *
108) 0000 0000 0000 0000 0000 0000 0000 0000    00000000       + *
109) 0000 0000 0000 0000 0000 0000 0000 0000    00000000       + *
110) 0000 0000 0000 0000 0000 0000 0000 0000    00000000       + *
111) 0000 0000 0000 0000 0000 0000 0000 0000    00000000       + *
112) 1110 1110 0001 0010 0011 0100 1110 1110    ee1234ee       + * ROD header
113) 0000 0000 0000 0000 0000 0000 0000 1001    00000009       + * Format version      9
114) 0000 0011 0000 0001 0000 0000 0000 0000    03010000       + *
115) 1110 1101 1010 0000 0011 0000 0000 0001    eda03001       + * Source ID: Master
116) 0001 1110 1100 1110 1000 1010 0011 0111    1ece8a37       + * BX: 516852279
117) 0000 0000 0000 0001 0100 0100 0010 0110    00014426       + * Lvl1ID: 82982
118) 0000 0000 0010 0111 0000 1010 1101 0010    00270ad2       + * BCO: 2558674
119) 0010 0001 0000 0100 0000 1010 0000 1101    21040a0d       + * Trigger type: AM    TS: 13
120) 0000 0000 0000 0000 0000 0000 0000 0000    00000000       + * Event type: 0
121) 0000 0011 0000 1001 0000 0010 0000 0010    03090202       + * N Hit Words 0-3:    (3, 3, 10, 4, )
:
```

Figure 6.10: **EDRO event format**. Decoded event from RUN number 3995. A binary and hexadecimal representation for each 32-bit word is given in the first two columns. In the rightmost column an comment is added to each decoded information.

```
160) 1101 1111 1010 1101 1011 0111 0001 1010    dfadb71a        + * Hit Block for line 7
161) 1101 1111 1010 1101 1011 0101 1010 1101    dfadb5ad        + * Hit Block for line 7
162) 1011 0001 1110 1011 0001 1110 0000 1111    b1eb1e0f        + * End ROD
163) 0010 1000 0010 1000 0010 1001 0010 0110    28282926        + * End ROB;  Checksum ok
164) 0000 0000 0000 0000 0000 0000 0000 0000    00000000        +
165) 0000 0000 0000 0000 0000 0000 0000 0001    00000001        +
166) 0000 0000 0000 0000 0000 0000 0010 1011    0000002b        +
167) 0000 0000 0000 0000 0000 0000 0000 0001    00000001        +
168) 0001 0010 0011 0100 1100 1100 1100 1100    1234cccc        + End of Event
Event 0 length = 115
Dump of event 82982
        Blocks 53 to 168
        Status: (3) Completed
        Warnings: 0
        Trigger: type: EXT, status: Completed
        Timestamp: 13   (13,13)

        BCO: 2558675, Lvl1ID: 82982, BX: 516852391
        ROBs found: 2
                Number of hits (ROB 0): 2 (N_AM: 0)
                Number of hits (ROB 1): 28 (N_AM: 2)

        /-------------------------------/
```

Figure 6.11: **EDRO event format**. Closure of an event packet and some statistic retrieved by the decoding program.

- The number of words received in each of the 8 hit-streams.

The sequence of hits received from each stream follows the header. The closure of the ROD field is followed by the checksum of the ROB.

Triggering

The triggering unit runs in parallel to the event builder. It is a remotely-configurable unit that ponders if certain conditions are achieved for the generation and expedition of an event, but it can also relay to external devices the triggering decision. Associative Memory is one of the possible external trigger sources (see next section), but also dedicated LEMO connectors are integrated in the front panel of the EDRO board in order to receive external TTL trigger pulses. Scintillators can be used in this way to provide a raw trigger.

In the developed DAQ architecture, only one of the two EDRO boards is responsible for the trigger generation/management and it is called *master*. The *slave* board is set in external trigger mode and receives the triggering information from the master. A dedicated EDRO-EDRO connection BUS is used for this delicate inter-operation of the two boards.

Data are then simultaneously acquired and buffered in both the EDRO boards, but only the front-ends connected to the Master are capable to generate a trigger. For this reason the standard configuration during the test beam have foreseen the telescope mounted on the Master board.

In case of internal triggering, several option can be configured at

startup. Trigger generation can require a minimum number of layers that should be hit, and/or a minimum number of hits per layer. The triggering policy is programmed in the EDRO core during the configuration phase which takes place before each run. All the trigger modes are listed below:

1. Burst Mode: trigger on N sequential events.
2. Pre-scaled trigger: select an event every N.
3. Sample Filled 1: select events having hits on N layers at least.
4. Sample Filled 2: select events having at least N hits.
5. Get external trigger.
6. AM Trigger with N tracks (N$\geq$0).
7. Mode 2 or Mode 6.
8. Mode 3 or Mode 6.
9. Mode 4 or Mode 6.

6.2.2 The Associative Memory

All the previously described trigger techniques are based on multiplicity, which means that they simply count the number of touched layers, the number of hits present in a layer and the overall count of hits. No spatial information is taken into account and therefore, together with particle track patterns, a lot of fake events are triggered with no physical meaning.

In presence of a high noise background, for example, many events can be produced where the hits are not aligned along a path. These generate an useless amount of data that steals transmission bandwidth, space in the data storage and processing time during data analysis.

A smarter trigger would be implemented if the spatial information could take part to the decision even in the fast level-0 trigger of a detector. At the moment there is no high-energy particle physics experiment that makes use of such a level-0 trigger, but the CDF collaboration (*Collider Detector at Fermilab*, another famous particle accelerator) is developing a system that can realize this goal. Looking forward for a potential utilization of the APSEL architecture, the collaboration decided to test the MAPS technology together with this innovative triggering facility.

The *Associative Memory* system aims to generate a trigger only if the hit pattern resembles a particle track. This is quite a complicated operation to do in hardware as it needs to deal with not sharp-edge conditions, and it needs to be very well calibrated to keep the efficiency high. Also practically speaking this is not trivial to achieve as this implies a very fast comparison between a bank of all the possible patterns and the incoming events. Realizing a pattern bank is only a matter of resource cost, but performing the parallel comparison is a more challenging task.

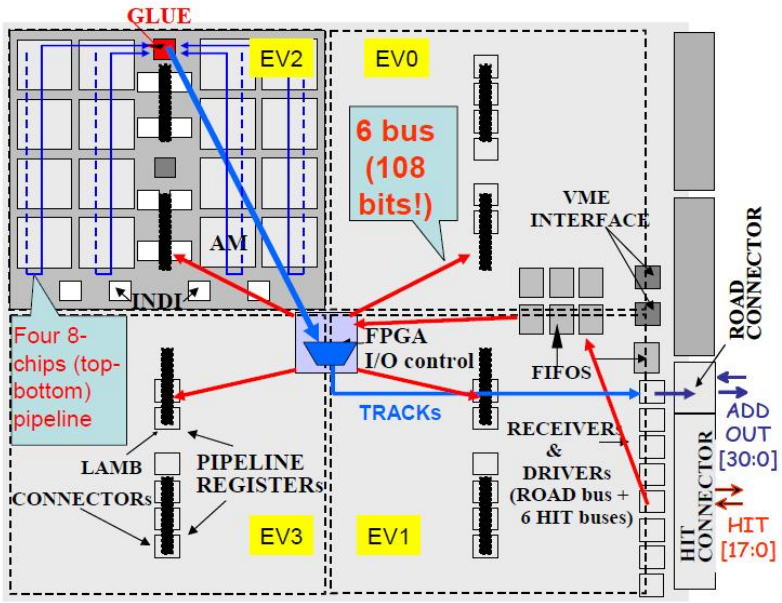

Figure 6.12: **SlimAMboard layout scheme**. Taken from [27].

A dedicated *Associative Memory* (AM) chip [9], developed inside the CDF collaboration, is the core responsible for the pattern matching. The AM chip, called *AM03*, is produced with a CMOS 0.18 μm process using a standard-cell design kit. It has an overall area of 9.8 $\times$ 9.8 mm^2 occupied at the 80% by the pattern bank. Each chip can store up to 5120 patterns. The time consuming pattern recognition problem, generally referred to as the "combinatorial challenge", is beat by the AM exploiting parallelism to the maximum level: it compares the event to precalculated "expectations" (pattern matching) at once. This approach reduces to linear the typical exponential complexity of the CPU-based algorithms. The problem is solved by the time data

are loaded into the AM devices.

Each pattern in the bank is provided with its own comparator, in this way each time an event is shifted into the AM it is compared in parallel to all the patterns present, with a consequent low-latency trigger response.

For the SLIM5 Test Beam, this *Associative Memory* enhanced triggering system has been implemented on an external 9U VME board, the AMBSlim. The AMBSlim board holds 4 mezzanine boards called LAMB (*Local Associative Memory Banks*) each one containing 32 AM chips.

The whole AMBSLim board can receive hit data at a rate of 4 Gbit/s on 6 parallel bus, 18-bit wide and it is able to search for matching patterns in four different pattern banks having 20k different patterns of tracks. Using the AM terminology, each recognized pattern returns a *Road*.

A central Xilinx Virtex II FPGA with 1696 pins constitute the control center and the main data switching network to interface towards the Master EDRO. This system provide a pattern based trigger with a latency <800 ns.

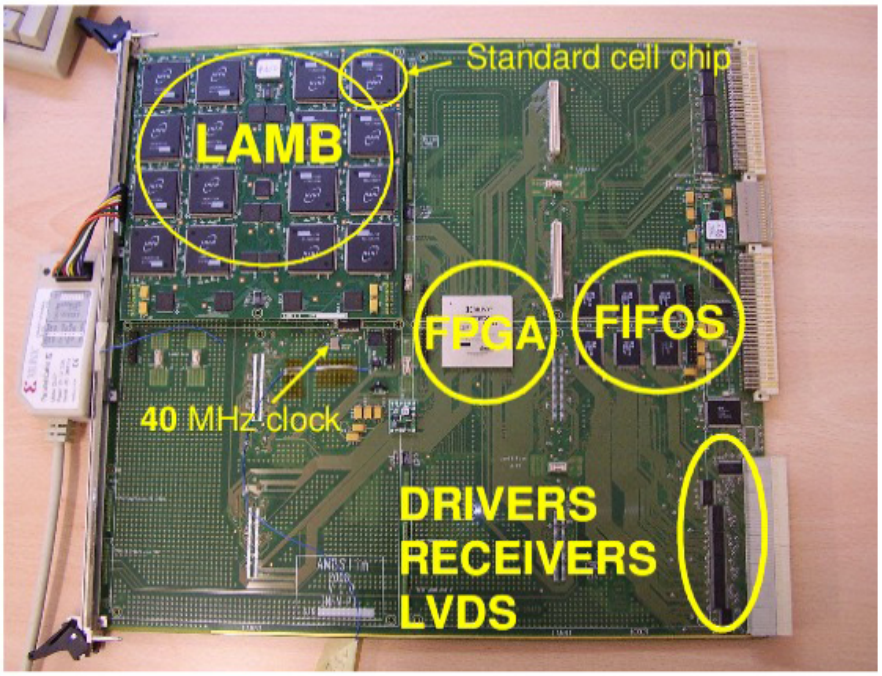

Figure 6.13: **SlimAMboard**.

6.2.3 The DAQ Software

The run controller, managing the whole telescope data taking procedure, was implemented using a customization of the ATLAS DAQ software framework. This program runs on the main DAQ PC where the optical receiver PCI card is housed. There data are stored in a hard drive exploiting *Direct Memory Access* technology in order to increase the acquisition performance. This main machine, that for the Test-Beam was called *PCSLIM*, was remotely operated by a secondary PC, connected over an ethernet network, called *PCSLIM2*. The VME CPU shared the same network and it was installed inside the crate acting as an interface for PCSLIM towards the VME BUS.

The main run-controller application working on PCSLIM is the *ATLAS TDAQ* software. This is a flexible framework highly customizable which in addition includes some debug features like an event-dump display and an on-line root histogramming monitor.

In particular the TDAQ framework consists of a *Graphical User Interface* (GUI) realized in JAVA, an XML description of the acquiring structure, and a set of C++ routines that handle data. Once again XML technology is used for the storage of configuration information which is read and sent to the front-ends at run time by the start-up routines. Basically the configuration routines access the VME BUS to write in the proper registers the configuration information which was read in the XML files.

Once power is provided to the front-ends with a dedicated infrastructure and software control, the acquisition can be started using the run control interface. The first step consists of filling in some global settings for the run, like the kind of data acquisition (physics, calibration...), specifying the maximum number of events to be acquired, customizing the text of the data file names and so on.

At this point the acquisition can be started by following a predefined sequence of steps on the Run Control interface. Three buttons allow to perform, step by step the required operations:

- **Boot**: it performs the bootstrap of all the hardware infrastructure, creates the connection between these and the run-control software.

- **Configure**: the system is reset, programmed and prepared for the acquisition. In this step the front-end and the readout are programmed with the information stored in the XML configuration file.

- **Start**: In this state the whole system is in run mode, a data file named with the current run number and other information is created and it starts to be filled by events.

To each of these steps corresponds the execution of a macro script that performs some transactions on the VME bus, mostly to write configuration registers but also to read back some registers information. These routines are written in C++ code and are compiled together with the run control program.

A screenshot of the run control GUI is presented in Fig. 6.14.

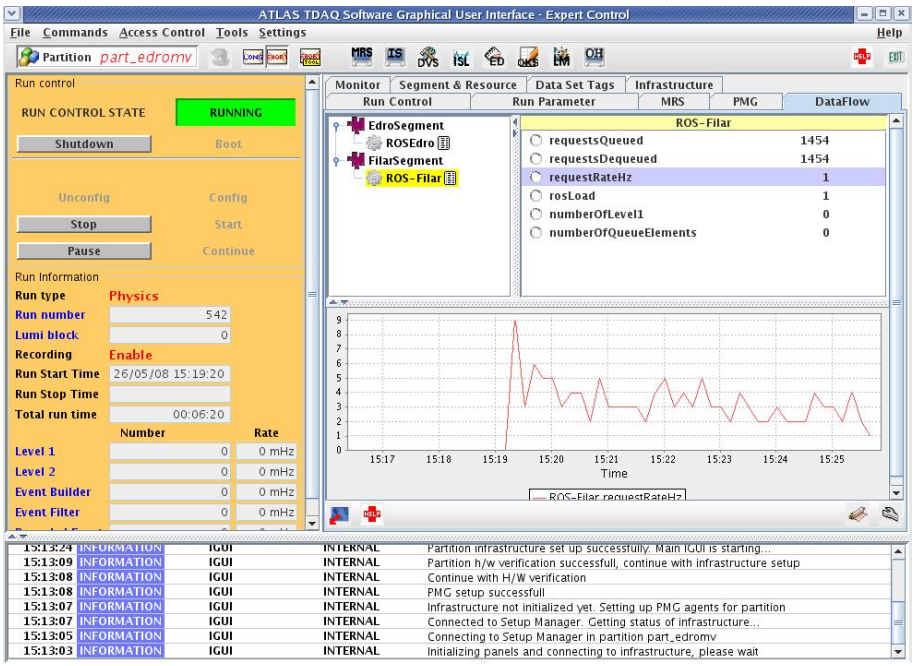

Figure 6.14: **TDAQ display screen-shot**.

6.2.4 The SlimGUI configuration software

The whole telescope configuration is performed by sending the proper information to the front-end chips via the VME bus and the front-end interfaces. All the configuration data are written, register by register into all the front-end interfaces during the *Configure* phase of the run control FSM.

Being the configuration information stored in a complex XML file, it became very important to have a versatile interface to set, store and

recall this great amount of data (we worked with up to 38 front-end chips). For these purposes a dedicated a graphical user interface was developed. This application allows the user:

- To set graphically and easily all the required parameters for the front-end chips and the DAQ system just filling in an organized form.
- To access directly the VME BUS with read/write operations in order to set or read quickly these registers.
- To save the configuration settings into an XML file ready to be read by the routines of the run control program.
- To revert a previously saved XML file in order to apport eventual modifications.

The name of the application is *SlimGUI* and two screen-shots of it are presented in the following figures.

In Fig. 6.15 is shown the section regarding the MAPS read-only registers residing in the MAPS EPMC. Within this tab it is possible to monitor the busy state of the FIFOs, the locking status of the PLLs, the current state of the APSEL Interface FSM and so on. In Fig. 6.16 is presented instead the trigger configurations interface. From this interface it is possible to set the multiplicity factor of the trigger conditions listed in the previous section.

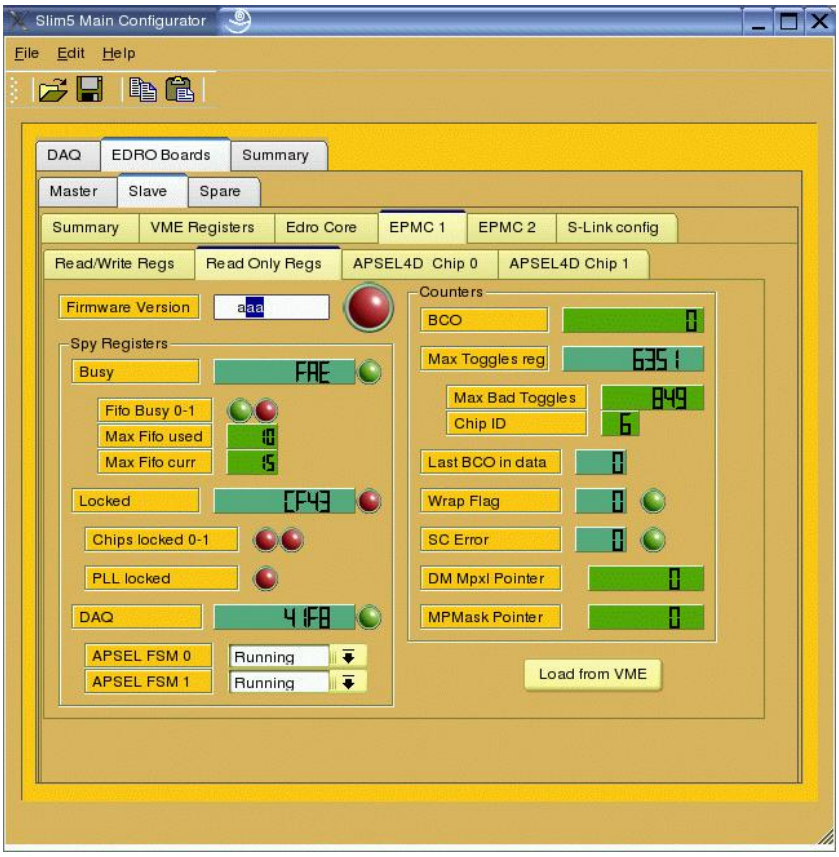

Figure 6.15: **SlimGui**.

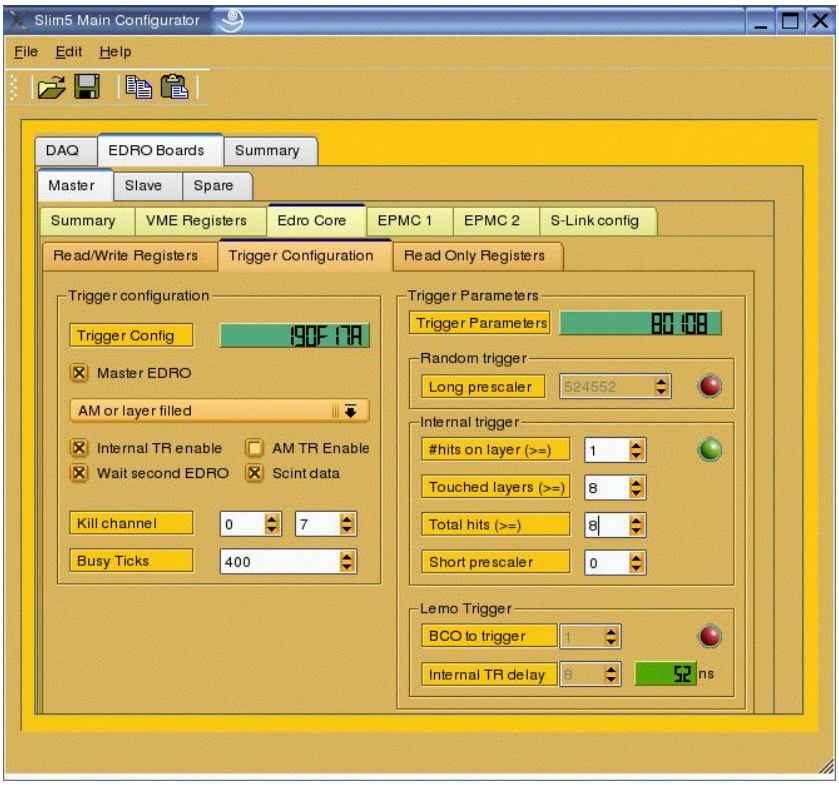

Figure 6.16: **SlimGui**.

Chapter 7

Test Beam Data Analysis and Results

In this chapter are shown the results of the analysis performed on the Test Beam data. Over the 15 days of data taking, the collaboration have collected 90 million events with a remarkable DAQ (and beam) live time fraction of 46%. The BCO period, controlling the time-tagging, was usually set to 5 μs to allow for a correct hit recording of the N-side micro-strip signals. In the typical running conditions we had spills of 30-35k tracks on a 2x2 cm^2 region lasting for 480 ms, the inter-spill duration was variable from 20 to 120 s. DAQ rates observed during the data taking reached routinely up to 40 KHz, with a typical event size of 500 bytes. Final data storage on disk could bear rates up to 20 MB/s but for higher data-rate spikes the Stratix and the Filar card FIFOs played a fundamental role of data buffering (they could buffer up to 2000 events).

The data-push architecture of the detector allowed to implement fast and smart triggers on the on-line acquisition system. This feature is seldom used in test beams and even rarely in large experiments, but in our case it was extremely useful to improve the efficiency of the data collection. The layer multiplicity trigger allowed to select events with easily reconstructible tracks. With the condition of $N_L \geq 6$ ($N_L =$ number of touched layers), we required events having hits both in the front, and in the rear telescope layers, which tuned out an easy and effective way to select tracks traversing completely the apparatus.

Using the AM allowed to increase furthermore the collection efficiency of reconstructible tracks, taking full advantage from the complete spatial information which is found in the data. By the way, in our case the AM were subject of study as well and thus not involved in the performance characterization of the DUTs.

Data analysis starts with the reconstruction of tracks using the telescope detectors. Clusters of fired strips are formed using an algorithm that simply associates adjacent fired strips. Typically a "true" cluster (one generated from a passing particle) contains one or two fired strips. The position of the cluster is calculated by weighting each fired strip with its measured deposited charge. For each telescope plane, clusters in measured on the U-coordinate (horizontal) side of the detector are combined with the V-coordinate (vertical) clusters to form space points.

A simple tracking algorithm has been implemented using the set of space points on the telescope detectors as input. All possible combinations requiring one space point are formed from each telescope layer. We also required that the space points lie along a three-dimensional line, within some tolerance. The resulting combinations are fit to a three-dimensional line using a basic χ^2 minimization. In the analysis phase, we typically require that an event have only one reconstructed track and that its χ^2 probability is greater than 0.10.

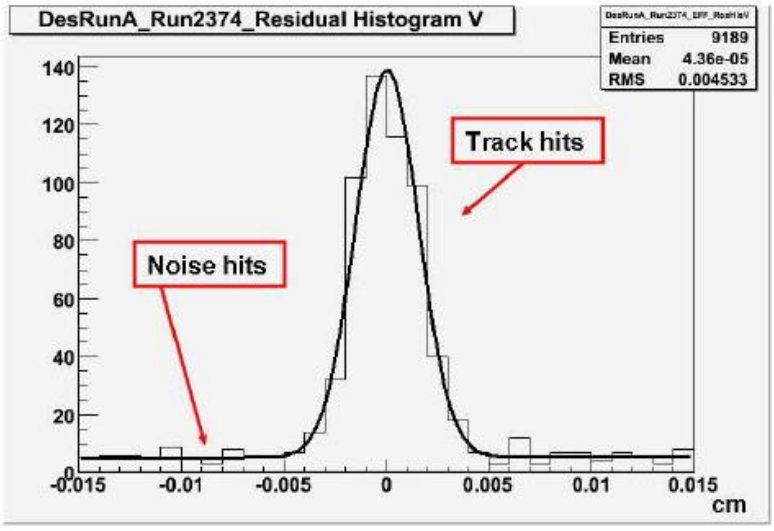

Figure 7.1: **Example of residuals fit**. Residual distribution after the alignment. Real hits contribute to the peak while fake noise hits are uniformly distributed.

Once a set of tracks has been reconstructed using the telescope detectors, we considered intersection point of the fitted track with one

or more *Devices Under Test* (DUTs) which may be either the APSEL MAPS chip or a striplets module. Depending on the sensor efficiency (that we are meant to measure), a space point will be probably generated on the DUT (signal hits). Noise hits can be present as well, but are distinguished from signal hits statistically by fitting the residuals distribution, where a residual is the position of the hit found in the DUT minus the position of intersection of the extrapolated track.

A simple function, consisting of a biased Gaussian is used to fit the distribution of residuals associated to passing tracks. The Gaussian peak is produced by signals hit, while the background contributes as a uniform bias to the fit.

An example of such a fit is shown in Fig. 7.1.

In this way the DUT resolution is determined by the width of the fitted residual distribution. The contribution to this width of the track extrapolation uncertainty and multiple scattering effects, both typically around 5 microns, are subtracted (in quadrature) to yield the intrinsic resolution.

The hit efficiency is then evaluated as the ratio between the number of real hits, extracted from a fit to the residuals distribution, and the number of tracks extrapolated on the DUT fiducial region.

7.1 APSEL4D results

The MAPS chips efficiency has been investigated in several configurations, varying the threshold as well as the incident angle of the tracks. We tested several APSEL chips characterized by a different thinning of the die, and the results presented concern the chip 22 and 23

7.1.1 Efficiency

The measured hit efficiency is shown in Fig. 7.2 as a function of threshold for two MAPS devices: chip 22 and 23 respectively 300 and 100 microns thick.

At the lowest thresholds we observe a maximum efficiency of approximately 92%, and we see the expected general behavior of decreasing efficiency with increasing threshold. The low efficiency, observed for Chip 22 at the lowest threshold, appears to have been caused by a readout malfunction. Investigations have shown that a small localized area on the detector had very low efficiency, while the rest of the detector behaved normally with good efficiency. Other efficiency scans with lower statistics showed efficiencies that were roughly 2-5% lower than

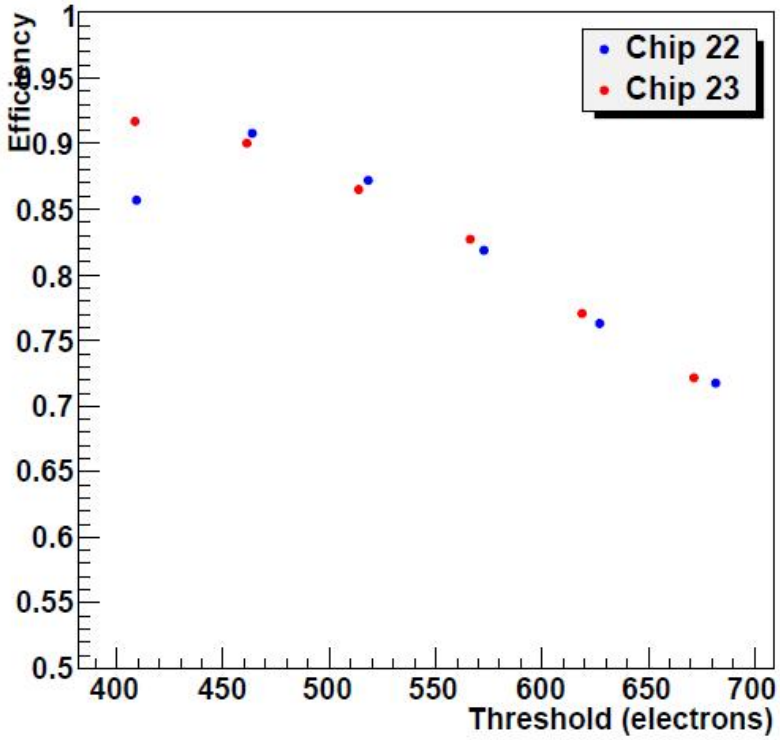

Figure 7.2: **Efficiency results for two MAPS detectors**. The points are measured over a single threshold scan and the statistical uncertainty on each point is smaller the graphical plotting symbol. Chip 22 is 300 μm and chip 23 is 100 μm thick.

those shown in Fig. 7.2. While the reasons for this difference are still under study, one possible cause is attributed to a significant difference in the operating temperature of the devices during the different scans.

Furthermore we have studied the efficiency for detecting hits as a function of the track extrapolation point within a pixel. Since the pixel has internal structure, with some areas less sensitive than others, we expect the efficiency to vary as a function of position within the cell. The uncertainty on the track position, including multiple scattering effects is roughly 10-15 microns, or about one-third of the pixel dimension. We have divided then the pixel into nine square sub-cells of equal area and measured the hit efficiency within each sub-cell.

The efficiencies thus obtained are "polluted" in some sense, due to the migration of tracks among cells. We obtain the true sub-cell

efficiencies by unfolding the raw results, taking into account this migration, which we characterize using a simple simulation. The result can be seen in Fig. 7.3, where we show the efficiency measured in each sub-cell.

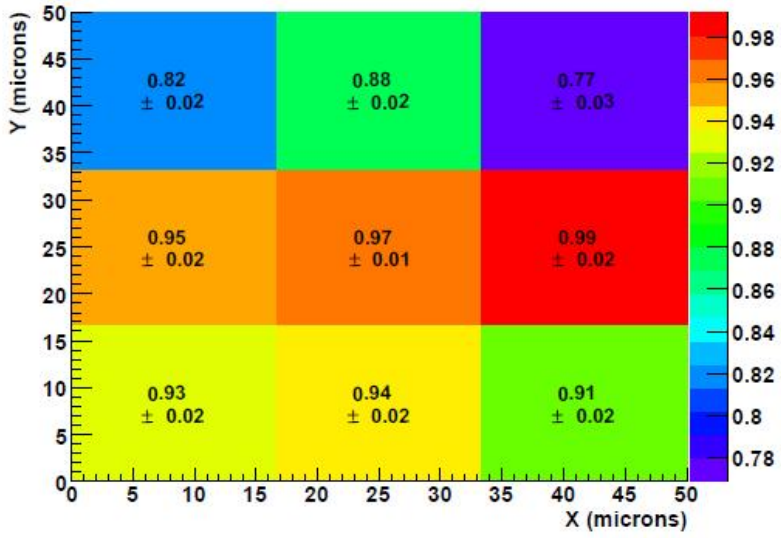

Figure 7.3: **Hit efficiency measured within the pixel**. The picture, which is not in scale, represents a single pixel divided into nine sub-cells. The efficiency and uncertainty values shown, are obtained taking into account track migration among cells.

We have also investigated the efficiency in dependence of the position within the MAPs matrix. Significant differences in efficiency as a function of position could indicate inefficiencies caused during readout. We have generally observed uniform efficiency across the area of the MAPs matrix and a preliminary plot of the study is presented in Fig. 7.4

7.1.2 Resolution

We measure the intrinsic resolution of the MAPS devices from the width of the residual distribution. The intrinsic detector resolution is obtained by subtracting the contributions from track extrapolation uncertainty and multiple scattering effects:

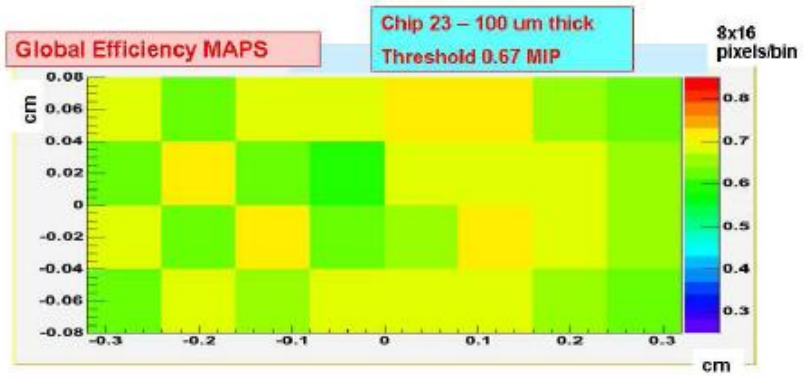

Figure 7.4: **Map of efficiency over the sensor area of chip 23**.

$$\sigma^2_{hit} = \sigma^2_{residual} - \sigma^2_{track} - \sigma^2_{MS}$$

The multiple scattering contribution is calculated for each unique configuration and is typically about 4-6 microns. The track extrapolation uncertainty has been calculated by propagating the track covariance matrix to the point of intersection with the DUT. It typically has a value of about five microns.

The value obtained with the formula above is consistent with the analytic prediction pitch$/\sqrt{12} = 50/\sqrt{12} = 14.4\mu m$.

Conclusions

Several activities were conducted during my PhD activity.

For the NEMO experiment a collaboration between the INFN/University groups of Catania and Bologna led to the development and production of a mixed signal acquisition board for the Nemo Km^3 telescope. The research concerned the feasibility study for a different acquisition technique quite far from that adopted in the NEMO Phase 1 telescope. The DAQ board that we realized exploits the LIRA06 front-end chip for the analog acquisition of anodic an dynodic sources of a PMT (*Photo-Multiplier Tube*). The low-power analog acquisition allows to sample contemporaneously multiple channels of the PMT at different gain factors in order to increase the signal response linearity over a wider dynamic range. Also the auto triggering and self-event-classification features help to improve the acquisition performance and the knowledge on the neutrino event.

A fully functional interface towards the first level data concentrator, the Floor Control Module, has been integrated as well on the board, and a specific firmware has been realized to comply with the present communication protocols. This stage of the project foresees the use of an FPGA, a high speed configurable device, to provide the board with a flexible digital logic control core. After the validation of the whole front-end architecture this feature would be probably integrated in a common mixed-signal ASIC (*Application Specific Integrated Circuit*). The volatile nature of the configuration memory of the FPGA implied the integration of a flash ISP (*In System Programming*) memory and a smart architecture for a safe remote reconfiguration of it.

All the integrated features of the board have been tested. At the Catania laboratory the behavior of the LIRA chip has been investigated in the digital environment of the DAQ board and we succeeded in driving the acquisition with the FPGA. The PMT pulses generated with an arbitrary waveform generator were correctly triggered and acquired by the analog chip and successively they were digitized by the on board ADC under the supervision of the FPGA.

For the communication towards the data concentrator a test bench has been realized in Bologna where, thanks to a lending of the Roma University and INFN, a full readout chain equivalent to that present in the NEMO phase-1 was installed. These tests showed a good behavior of the digital electronic that was able to receive and to execute command imparted by the PC console and to answer back with a reply. The remotely configurable logic behaved well too and demonstrated, at least in principle, the validity of this technique.

A new prototype board is now under development at the Catania laboratory as an evolution of the one described above. This board is going to be deployed within the NEMO Phase-2 tower in one of its floors dedicated to new front-end proposals. This board will integrate a new analog acquisition chip called SAS (*Smart Auto-triggering Sampler*) introducing thus a new analog front-end but inheriting most of the digital logic present in the current DAQ board discussed in this thesis.

For what concern the activity on high-resolution vertex detectors, I worked within the SLIM5 collaboration for the characterization of a MAPS (*Monolithic Active Pixel Sensor*) device called APSEL-4D. The mentioned chip is a matrix of 4096 active pixel sensors with deep N-well implantations meant for charge collection and to shield the analog electronic from digital noise. The chip integrates the full-custom sensors matrix and the sparsification/readout logic realized with standard-cells in STM CMOS technology 130 nm.

For the chip characterization a test-beam has been set up on the 12GeV PS (*Proton Synchrotron*) line facility at CERN of Geneva (CH). The collaboration prepared a silicon strip telescope and a DAQ system (hardware and software) for data acquisition and control of the telescope that allowed to store about 90 million events in 7 equivalent days of live-time of the beam. My activities concerned basically the realization of a firmware interface towards and from the MAPS chip in order to integrate it on the general DAQ system. Thereafter I worked on the DAQ software to implement on it a proper Slow Control interface of the APSEL4D.

Several APSEL4D chips with different thinning have been tested during the test beam. Those with 100 and 300 μm presented an overall efficiency of about 90% imparting a threshold of 450 electrons. The test-beam allowed to estimate also the resolution of the pixel sensor providing good results consistent with the pitch/$\sqrt{12}$ formula. The MAPS intrinsic resolution has been extracted from the width of the residual plot taking into account the multiple scattering effect.

Bibliography

[1] *Km3NET: Conceptual Design Report*, www.km3net.org.

[2] *M.A. Markov, On high energy neutrino physics in: Proceedings of the 1960 Annual International Conference on High Energy Physics. p. 578*, Rochester, NY, 1960.

[3] J.A. Aguilar. *Astropar. Phys.*, 26(314), 2006.

[4] G. Anassozontis and P. Koske. *Sea Technology*, 44(7), 2003.

[5] M.C. Bouwhuis. The data acquisition system for the antares neutrino telescope. *Nuc. Instr. Meth. A*, 570(107-116), 2007.

[6] Alice Collaboration. The alice experiment at the cern lhc. *J. Inst.*, 3 S08002, 2008.

[7] E. de Wolf, editor. *Proceedings of the workshop on Technical Aspects of a Very Large Volume Neutrino Telescope in the Mediterranean Sea*, Amsterdam, October 2003. vlvνT.

[8] M. Bonori e F. Ameli. NEMO electronics report. Ver. 0.9 β, Luglio 2004.

[9] A. Annovi et Al. *IEEE Trans. Nucl. Science*, 53(1726-1731), 2006.

[10] A. Capone et Al. *Nuc. Instr. Meth. A*, 487(423), 2002.

[11] A. Gabrielli et Al. On-chip fast data sparsification for a monolithic 4096-pixel device. *IEEE Trans. Nucl. Science - approved (2008).*

[12] D. Allard et Al. *Astro-ph*, 0605327.

[13] D. Lo Presti et Al. *Nuc. Instr. Meth. A*, 567(548-551), 2006.

[14] D. Lo Presti et Al. *Nuc. Instr. Meth. A*, 596(100-102), 2008.

[15] E. Andrés et al. *Nature*, 410(441), 2001.

[16] E. Migneco et Al. The status of nemo. *Nuc. Instr. Meth. A*, 567(521), 2006.

[17] F. Ameli et Al. *Nuc. Instr. Meth. A*, 423(146), 1999.

[18] F. Ameli et Al. *IEEE Trans. Nucl. Science*, 55(233), 2008.

[19] F. Simeone et Al. *Nuc. Instr. Meth. A*, 588(119-122), 2008.

[20] F.M. Giorgi et Al. *Nuc. Instr. Meth. A*, 596(103-106), 2008.

[21] G. Agourras et Al. *Astropar. Phys.*, 23(377), 2005.

[22] G. Anassozontis et Al. *Nuc. Instr. Meth. A*, 479(439), 2002.

[23] G. Bunkheila et Al. *Nuc. Instr. Meth. A*, 567(559-562), 2006.

[24] G. Rizzo et Al. Development of deep n-well maps in a 130 nm cmos technology and beam test results on a 4k-pixel matrix with digital sparsified readout. *IEEE Nuclear Science Symposium Conference Record*, 2008.

[25] I.A. Belolaptikov et al. *Astropart. Phys.*, 7(263), 1997.

[26] J.R. Hoff et Al. *IEEE Trans. Nucl. Science*, 48(3), 2001.

[27] M. Piendibene et Al. The associative memory for the self-triggered slim5 silicon telescope. *IEEE Nuclear Science Symposium Conference Record*, 2008.

[28] M. Ruppi et Al. *Nuc. Instr. Meth. A*, 567(566), 2006.

[29] R. Abbasi et Al. *Phys. Lett. B*, 619(271), 2005.

[30] S.W. Barwick et Al. *Astropar. Phys. J*, 498(779), 1998.

[31] A. Rovelli for the NEMO coll. *Nuc. Instr. Meth. A*, 567(569-572), 2006.

[32] P. Piattelli for the NEMO Collaboration. The status of nemo. *Nuc. Phys. B. (Proc. Suppl.)*, 165(172-180), 2007.

[33] E. Gandolfi, A. Gabrielli, and P. Ricci. Control board for optical modules of a high energy neutrino experiment. Physics Department - Università di Bologna.

[34] J. Hoerandel. *Astropar. Phys.*, 19(193), 2003.

[35] K. Mannheim J.G. Learned. *Ann. Rev. Nucl. Part. Sci*, 50(679), 2000.

[36] J. Linsley. *Phys. Rev. Lett.*, 10(146), 1963.

[37] Domenico Lopresti. *Optical module front-end VLSI full-custom ASIC for a submarine neutrino detector*. PhD thesis, Università degli studi di Catania, 2002.

[38] C.A. Nicolau. *Nuc. Instr. Meth. A*, 567(552-555), 2006.

[39] Carlo Nicolau. Studio e realizzazione di un sistema programmabile per l'esperimento NEMO. Master's thesis, Università degli studi di Roma "La Sapienza", 2003.

[40] K. Parnell and N. Mehta. *Programmable Logic Design quick start hand book*, 4th edition, june 2003.

[41] Proceedings of first VLVνT Workshop. *G. Riccobene et Al. - Overview over Mediterranean water parameters*, Amsterdam, October 2003.

[42] Pune, editor. *atro-ph/0507150, Proc. 29th Int. Cosmic Ray Conf.*, India, 2005.

[43] M. Sedita. *Nuc. Instr. Meth. A*, 567(531), 2006.

[44] W. Snoeys. Pixel readout electronics development for the alice pixel vertex and lhcb rich detector. *Nuc. Instr. Meth. A*, 465(176-189), 2001.

[45] L. Ratti V. Re, M. Manghisoni. Fssr2, a self-triggered low noise readout chip for silicon strip detectors. *IEEE Nuclear Science Symposium Conference Record N16-1*, 2005.

[46] M. Nagano & A.A. Watson. *Rev. Mod. Phys.*, 72(689), 2000.

[47] A. X. Widmer and P. A. Franaszek. A DC-balanced, partitioned-block, 8b/10b transmission code. *IBM J. Res. Develop.*, 27(25), September 1983.

[48] G.T. Zatsepin and V.A. Kuz'min. *Sov. Phys. JETP Lett.*, 4(78), 1966.