





UNIVERSITÀ DEGLI STUDI DI NAPOLI
“FEDERICO II”

Dottorato di Ricerca in Fisica Fondamentale ed Applicata
XIX ciclo

V i n c e n z o I z z o

Design and test of the Read-Out driver
for the trigger system of the muon
barrel spectrometer of the ATLAS
experiment

P h D T h e s i s

Advisor:

Prof. S. Patricelli

Prof. A. Aloisio

Coordinator:

Prof. G. Miele

Contents

Introduction

Chapter I - LHC and the ATLAS detector.....	1
The physic goals at the LHC.....	1
The <i>Large Hadron Collider</i>	6
The ATLAS experiment.....	9
The inner detector.....	11
The magnetic system.....	13
The calorimetric system.....	14
The electromagnetic calorimeter.....	14
The hadronic calorimeter.....	16
The Muon Spectrometer.....	17
<i>Monitored Drift Tubes</i> (MDT).....	21
<i>Cathode Strip Chambers</i> (CSC).....	22
<i>Resistive Plate Chambers</i> (RPC).....	23
<i>Thin Gap Chambers</i> (TGC).....	26
The Trigger and the Data Acquisition System.....	27
 Chapter II - The level-1 trigger and the Data Acquisition system of the ATLAS muon spectrometer.....	33
The structure of the spectrometer.....	34
The trigger system in the barrel region.....	34
The principle of the trigger for muons detection.....	35
The read-out electronics.....	37
The synchronization.....	40

The spectrometer segmentation.....	45
The electronics on the RPC detectors.....	46
The Coincidence Matrix.....	47
The PAD Logic boards.....	54
The path of the information.....	57
The optical link.....	60
The RX- Sector Logic board.....	64
The Sector Logic section.....	65
The RX section.....	67
 Chapter III - The Read Out Driver of the Muon Spectrometer of ATLAS.....	71
The ROD in the acquisition structure.....	72
The read-out data in the RX/SL board.....	72
The RODbus backplane.....	74
The design in VHDL.....	75
The ROD.....	77
The VME FPGA.....	81
The serial custom protocol between the FPGAs.....	82
The ROD FPGA.....	85
The main Data and Control paths.....	86
The serializers (SerDes.....	88
The TTCrq.....	89
S-Link.....	93
The <i>FIFO</i> memories.....	96
Clock domains and clock muxes.....	98
The ARM7 microcontroller.....	101
 Chapter IV - The internal logic of the Read Out Driver.....	105
The <i>Event Builder</i> functionality of the ROD.....	106

The Output Frame of the ROD.....	107
The Architecture of the <i>Event Builder</i>	112
The <i>Event Builder</i> engine.....	113
The <i>Event Builder</i> error handling.....	116
The RX frame syntax error.....	117
The Timeout error.....	118
The L1Accept error.....	119
The test of the ROD.....	122
 Conclusions.....	125
 Appendix A	
The VME FPGA registers.....	127
The ROD FPGA internal registers.....	129
 Appendix B	
The LVDS standard.....	137
 Bibliography.....	139

Introduction

This thesis, submitted for the Degree of Doctor of Philosophy, describes the design and the test of the *Read Out Driver* (ROD) of the data produced by the RPC detectors of the ATLAS muon spectrometer. The ROD is a VME board that performs the *online* event building of data produced by the ATLAS spectrometer. The ROD also allows to control some of the critical elements of the data acquisition system, such as the synchronization of the acquisition system and the correct data formatting. In the ATLAS DAQ system, the task of the ROD is fundamental because it acts as the supervisor (or *Master*) in the management of a whole subsystem - made by some acquisition boards - that handles data of one of the 32 sectors that constitute the ATLAS apparatus. The ROD board is based on the FPGA (*Field Programmable Gate Array*) technology, programmable electronics devices that can reach working frequencies of hundred of MHz and allow to create complex logic functions and memory elements on few square centimetres.

ATLAS (A Toroidal LHC ApparatuS) is one of the experiment that will be built at the *Large Hadron Collider*, that should be completed by 2007 at *CERN* (“*Centre Europeenne pour la Recherche Nucleaire*”) in Geneva, Switzerland. LHC will provide proton-proton collisions at the center-of-mass energy of 14 TeV, a value that has never been reached in high energy physics. Such center-of-mass energy and the foreseen luminosity of $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ will allow to confirm the Standard Model predictions on the existence of the Higgs boson. The ATLAS experiment has been designed to search for Higgs signal in a wide mass range (between 100 GeV and 1 TeV) and in its main decay channels. The design characteristics of the apparatus will also allow to perform accurate measurements of the *top* quark physics and of the B mesons and to search for possible SUSY particles.

The muon spectrometer is located in the outer region of the apparatus and plays a key role for many physics program items of ATLAS. The presence of muons with a high transverse momentum (p_T) is a signature that indicates the production of high mass particle at the interaction point. The selection and the precision measurement on muons with high p_T is crucial in studying the decay channels of the Higgs boson or in detecting mesons composed by the b quark. In order to study the Higgs boson decay channels, the channels with a high p_T are the most important because they have a very good signal-to-noise ratio.

In the central part of the spectrometer, in order to produce information for the ATLAS trigger, RPC detectors are used. The information produced by the RPC detectors are collected by on detector electronics and are transferred, over optical fiber, to the counting room *USA15*, located at about 100 meters from the detector. In the counting room *USA15*, data are received by receiver boards (*RX*), are preliminary elaborated and are transferred to the ROD. The ROD has the task to manage and further elaborate such data, performing a syntactical analysis, producing *frames* depending on some detector parameters and sending the produced frames to the next acquisition levels. The ROD also has the task to receive some LHC signals, fundamental for the synchronization of the acquisition system, and to distribute them to the *RX* boards. Such signals are, as example, the LHC 40 MHz *bunch crossing* signal and the signal of level one trigger (*Level1 Accept*), that are distributed on dedicated low-noise connections.

The first chapter is devoted to the description of the main ATLAS experiment features. The main design parameters of the LHC collider and the main items of the ATLAS physics program are presented, together with the different detectors of the ATLAS apparatus. A particular care is taken in describing the main characteristics of the muon spectrometer, where the designed trigger system uses RPC detectors. A general description of the ATLAS trigger and data acquisition system is given too.

The second chapter describes the structure of the level 1 trigger system of the muon spectrometer: my Ph.D. thesis work is included in this section of the experiment. Moreover, the structure of RPC detectors' acquisition system is presented, paying attention to the data transmission system from the detectors to

the control and elaboration room. Hence, the structure of the *on detector* boards is described, together with the data transmission system over optical fiber; a brief description of the *off-detector* boards, receiving the spectrometer information, is given too.

The third chapter describes the *Read Out Driver (ROD)* and is a part of my original contribution to this Ph.D. thesis work. The chapter presents the environment that hosts the ROD and the experiment specifications to comply with. The *VHDL* language is introduced too: it's an hardware description language that allows the simulation and the synthesis on programmable logic devices (e.g. *FPGA*), used to design most of the ROD functionalities. The board model, described by VHDL language, has allowed to perform a comparison between different implementation solutions, to evacuate the different performances and to start the carrying out of the board.

In the chapter three, an accurate description of the ROD board architecture, together with the block schemes of the different sections of the board itself. Moreover, the *RODbus* custom backplane, designed to guarantee a correct communication between the data acquisition boards managed by the ROD, is described in detail. A brief description of the different devices used on the board is given too.

In the fourth chapter a detailed description of the output ROD *frame* format – that will be sent to the next acquisition levels – is presented. The main characteristics of the *Event Builder Engine*, that is at the basis of the functionality of online event building, are presented. Moreover, the logic and syntactic controls – performed by the ROD on the input frames – are described, together with the procedures followed in case of errors. Eventually the results of the tests performed on the ROD board are presented and the integration tests in the ATLAS apparatus are described too.

Finally the conclusions summarize the results of this Ph.D. thesis work and illustrate the perspectives of performing new functionalities implementation and new test procedures in the ATLAS experiment.

Chapter 1

LHC and the ATLAS detector

This chapter describes the general characteristics of the Large Hadron Collider (LHC) [1, 2] and of the ATLAS [3, 4, 5] apparatus, one of the four detectors that will operate at the LHC. The physical goals of the LHC and of the ATLAS detector are briefly explained; then, the different elements of the ATLAS experiment are described: the inner detector, the magnet system, the calorimetric system and the muon spectrometer. A particular care is taken in describing the spectrometer's structure in the barrel region and the detectors used in the trigger system. More details on the trigger and the data acquisition systems of the muon spectrometer will be given in the next chapter.

The physic goals at the LHC

High energy physics is always searching for new discoveries and nowadays one of the most important objectives is the discovery of the evidence of the *Higgs* boson, the last piece of the *Standard Model*.

The *Standard Model* [6] groups three of the four known fundamental interactions: the strong interaction, the weak and the electromagnetic

interaction. The *Standard Model* is a quantum field theory that describes the interaction of spin- $1/2$ point-like fermions (quarks and leptons) as mediated by spin-1 bosons (γ , gluons, W and Z). Specifically, eight bosons (the gluons) are the mediators of the strong interaction between the quarks, one boson (the photon) is responsible of the electromagnetic interaction between electrically charged particles and three (the W^+ , W^- , Z^0 bosons) are the mediators of the weak interaction. The *Lagrangian* of the theory of the *Standard Model* has a global gauge invariance under the symmetry group $SU(3)_C \otimes SU(2)_I \otimes U(1)_Y$ (C =Colour, I =Isospin, Y =Ypercharge) and the interaction is the manifestation of a local gauge invariance, that implies also that all the masses of the fermions and bosons are equal to zero.

The particle masses are introduced through the *Spontaneous Symmetry Breaking* mechanism: the local gauge symmetry $SU(2) \otimes U(1)$ is broken by the existence of a *Higgs* field that implies the existence of a neutral scalar boson H^0 , the *Higgs* boson. The so-far undiscovered *Higgs* boson is thought to be responsible of the masses of all the particles in the *Standard Model*.

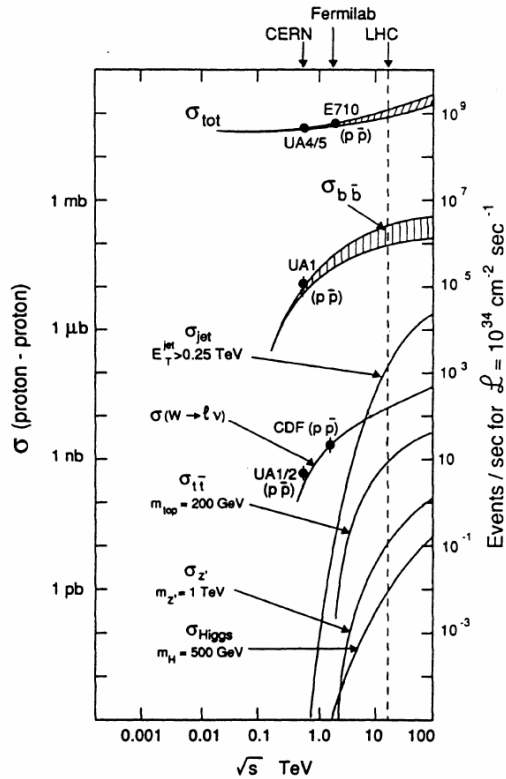


Fig. 1.1 Proton-proton cross section as function of the center-of-mass energy

The *Standard Model* is presently in agreement with all the experimental results[7]: its predictions were confirmed by the discovery of the neutral weak current, in the 70's, and by the discovery of the $W^\pm$, Z^0 bosons at the CERN in 1983, but there is no experimental evidence for the Higgs boson yet. One of the physic goals of the LHC is the discovery of the Higgs boson. Fig. 1.1 shows the cross sections for various processes as a function of the center-of-mass energy; the working line of the LHC is also shown: the cross section of a *Higgs* boson production is also drawn, in the hypothesis of a *Higgs* mass of 500 GeV.

Moreover, due to its highest center-of-mass energy and luminosity, LHC will provide indications of new physics penomena beyond the Standard Model, if they exist.

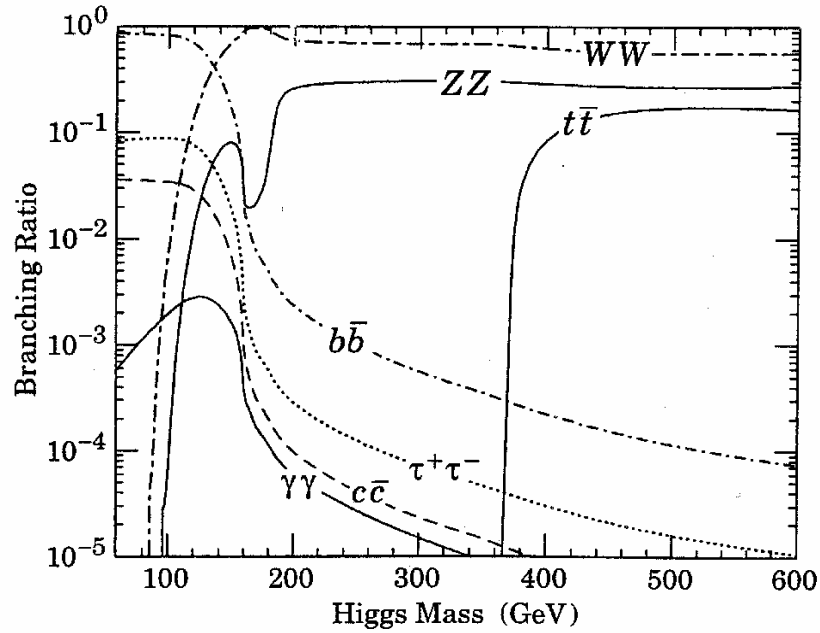


Fig. 1.2 Branching ratio for different decay channels of the Higgs boson

Fig.1.2 shows the *Branching Ratio* of a *Higgs* boson in various decay channels, as a function of the Higgs boson mass. Some processes, because of their decay products, are more difficult to be detected than other processes. For example, processes with neutrinos in the final state will require missing energy studies;

processes with a background of hadronic jets will have a difficult reconstruction procedure. The decays that seem to be the most easily detectable and so the most reliable for the Higgs detection are:

$$H \rightarrow ZZ \rightarrow 4 \text{ leptons (especially with high transverse momentum, } p_T > 7 \text{ GeV/c)}$$

$$H \rightarrow ZW^* \rightarrow 2 \text{ leptons} + 2\nu$$

$$H \rightarrow ZZ^* \rightarrow 4 \text{ leptons}$$

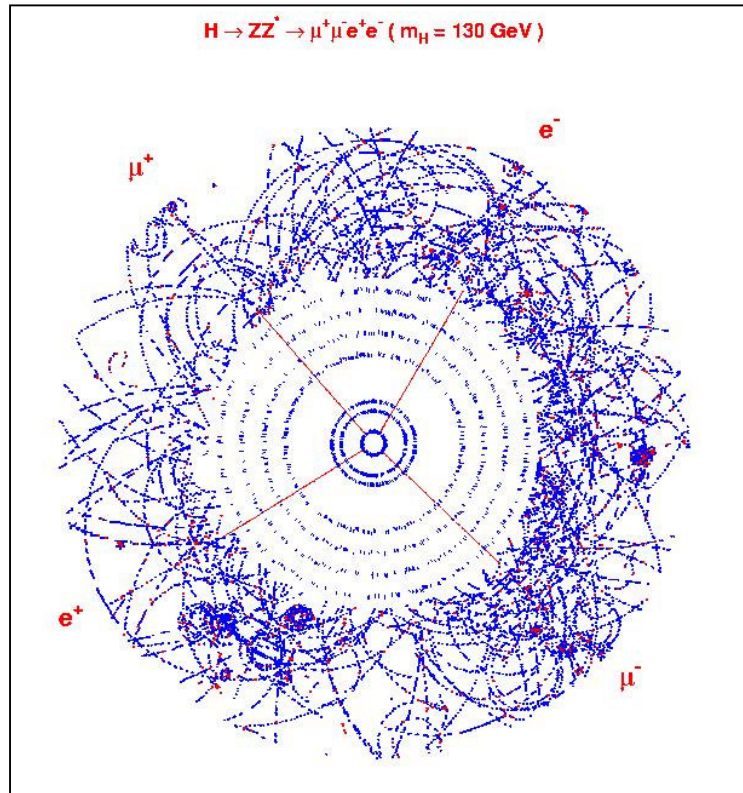


Fig.1.3 Simulation of the decay $H \rightarrow ZZ \rightarrow \mu^+ \mu^- e^+ e^-$

With the hypothesis $m_H = 130 \text{ GeV}/c^2$

To detect as many decay channels as possible, very good performances will be required to every single detector of the ATLAS apparatus. A good energy resolution will be required to identify and discriminate electrons, leptons and jets from the $\gamma\text{-}\gamma$ pairs, and an equally good precision will be needed to reconstruct the interaction vertexes. If we think that this will happen in an environment exposed to high fluxes of ionizing radiation, we can easily understand that the entire ATLAS

experiment is a real challenge to the performances of the present days' technologies.

Other still partly unexplored topics of the modern physics are the systems of heavy quarks. One of the studies that will be performed at the LHC is the determination of the quark top mass with a resolution less than $0.05 \text{ GeV}/c^2$. The most interesting decay to reach such purpose is

$$t \bar{t} \rightarrow b \bar{b} W(jj) W(l\nu)$$

so the ATLAS apparatus will have to be able to detect leptons in the final state and reconstruct secondary decay vertexes. These characteristics will be also used to investigate the B^0 meson physics, e.g. the rare decay study

$$B_d^0 \rightarrow \mu^+ \mu^- (X)$$

or the decay

$$B_d^0 \rightarrow J/\psi K_s^0$$

useful in CP violation studies.

In summary, the LHC collider has been designed to investigate elementary particle physics in an energy range not yet explored, so it will allow to investigate a wide number of the still unsolved problems in high energy physics:

- the search of an experimental evidence of the Higgs boson and the origin of the spontaneous electroweak symmetry breaking;
- the study of direct CP violation and the measure of the parameters of the Cabibbo-Kobayashi-Maskawa matrix;
- the heavy quarks physics, especially the B quark and the T quark;
- the existence of a possible composite structure of the quarks;

Moreover, at the energies reachable at the LHC, signals for new physics such as SUSY particles or unexpected discoveries could be seen.

The Large Hadron Collider

The *Large Hadron Collider* has been designed to produce proton-proton collisions at a center-of-mass energy of 14 TeV.

The *LHC* should be completed by 2007 at *CERN* (“*Centre Europeenne pour la Recherche Nucleaire*” in Geneva, Switzerland) in the tunnel of 27 Km where the *LEP* (*Large Electron-Positron collider*) was previously accommodated. In order to reach beam energy of about 7 TeV, a network of different subsystems or pre-accelerators gradually accelerate the particles, before injecting them in the *Large Hadron Collider*, as shown in the fig. 1.4.

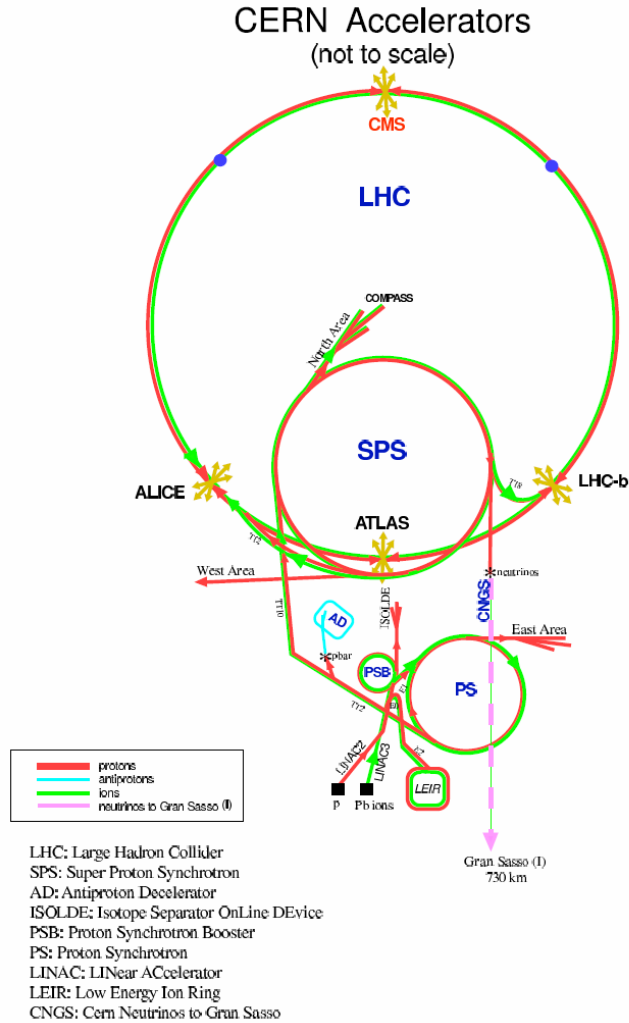


Fig. 1.4 The LHC proton injection system

After production, protons are accelerated up to 50 MeV in the linear accelerator LINAC, then up to 1 GeV in the “*Proton Synchrotron Booster*” and then to 26 GeV in the *PS (Proton Synchrotron)*; finally, the *SPS (Super Proton Synchrotron)* accelerates protons up to 450 GeV and inject them in the *LHC*.

The high luminosity of the LHC is one of the most interesting characteristics of the collider, because it will allow studying events with very low cross sections. The rate R of production of a given process is dependent by the physics of that process, i.e. by the cross section σ , and by the technology of production of very concentrated particle beam, i.e. by the luminosity L . From the relation $R = \sigma L$, if we have high luminosity, it is possible to observe events with very low cross sections σ .

The LHC beam is a succession of proton bunches, each one containing N protons, spaced by 25 ns equivalent to a collision frequency f of 40 MHz. The LHC’s luminosity can be expressed as follows:

$$L = \frac{N^2 k_b f_{\text{rev}} E}{m_p 4\pi \beta^* \epsilon} F$$

where N is the number of protons per bunch, k_b is the number of bunches in the beam, f_{rev} is the bunch collision frequency, F is a factor including the not-exact beam collimation ($F \sim 0.9$), m_p is the proton mass, β^* is a factor depending upon the beam focalization, E is the beam energy and ϵ is the normalised transverse emittance, a conserved quantity depending by the transverse beam dimensions.

In the first three years of run, LHC will run at the so-called low luminosity ($10^{33} \text{cm}^{-2} \text{s}^{-1}$) allowing to study the heavy quarks physics and a preliminary search for the Higgs boson; after three years, LHC will run at high luminosity regime ($2.5 \cdot 10^{34} \text{cm}^{-2} \text{s}^{-1}$), intended to compensate the low Higgs production cross section.

Tab. 1.1 shows the most important design parameters of LHC [8].

Beam energy (E)	7 TeV
Dipolar magnetic field	8.28 Tesla
Instantaneous luminosity	$10^{34} \text{ cm}^{-2} \text{ s}^{-1}$
Accelerator length	26.66 Km
Injection energy	450 GeV
Number of protons per bunch (N)	1.1×10^{11}
Bunch spacing	7.48 m
Number of bunches (k_b)	2835
Circulating current	0.56 A
Distance between bunches (Δt)	25 ns
Normalized transverse emittance (ϵ)	$3.75 \mu\text{m rad}$
β function at interaction points (β)	0.5 m
Beam lifetime	22h
Luminosity lifetime	10 h
Energy loss per turn	6.9 KeV
Total radiated power per beam	3.7 KW

Tab. 1.1 The design parameters of the *Large Hadron Collider*

The high luminosity will grant a high production rate of interesting events and will balance the selection made to reject the background events.

In the LHC beam, protons are grouped in *bunches*, crossing every 25 ns: as the estimated proton-proton cross section σ_{p-p} at 14 TeV is about 110 mb, at the luminosity of $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ there will be 27 interactions per *bunch crossing*, from the formula

$$N = L \cdot \sigma_{p-p} \cdot \Delta t \sim 27$$

corresponding to $\sim 1.1 \times 10^9$ Hz event production frequency.

If we consider interactions with a *Higgs* boson in the final state, the cross section will be very little, i.e. for the production via gluon fusion of a *Higgs* boson with a presumed mass of 200 GeV, $\sigma (gg \rightarrow H) \sim 10 \text{ pb}$; in this case, the rate of production of a *Higgs* boson via gluon fusion will result to be

$$\sigma (gg \rightarrow H) \cdot L \sim 10 \text{ pb} \cdot 10^{34} \text{ cm}^{-2} \text{ s}^{-1} = 0.1 \text{ Hz}$$

so a Higgs boson should be produced every 10 seconds.

The ATLAS experiment

ATLAS is a detector that will be built at one of the four LHC beam interaction point. It has a cylindrical symmetry along the beam axis and the cylinder has a length of 46 m and a diameter of 22 m.

The ATLAS structure is shown in fig. 1.5. Closest to the interaction point there is a tracking detector and, from inner to outer radius, there are: an electromagnetic calorimeter, an hadronic calorimeter and a muon spectrometer. All these sub-detectors are in a magnetic field in order to bend the charged particle trajectories and measure their momentum.

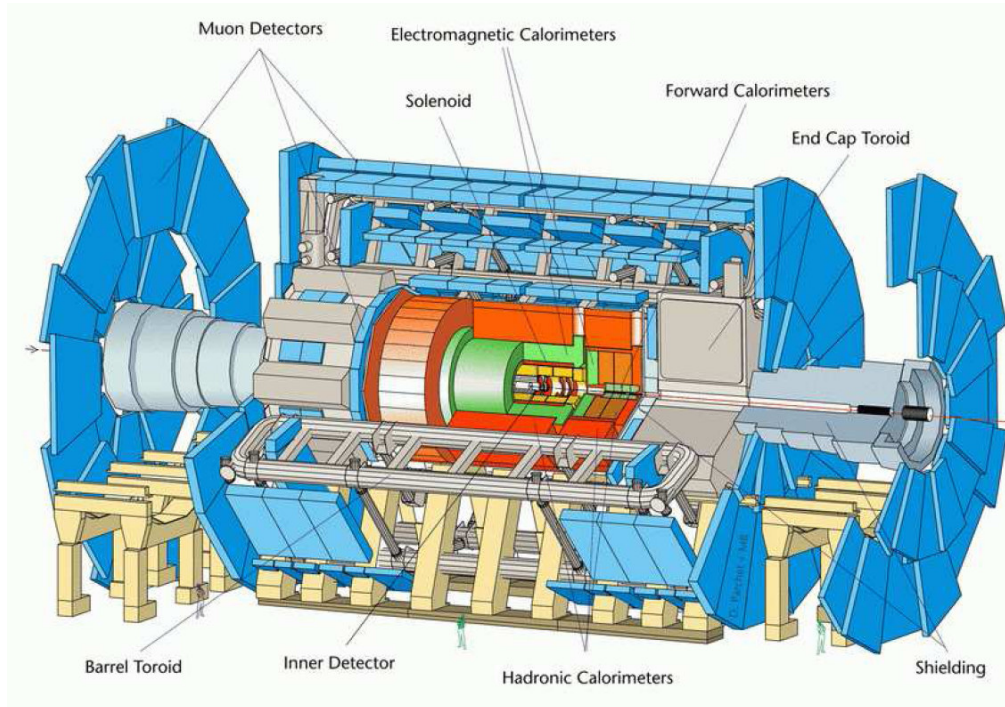


Fig. 1.5 The ATLAS detector layout

Fig. 1.6 shows the ATLAS coordinate system. The system is right-handed with the z-axis along the beam direction and the x-axis pointing toward the center of the LHC circumference; the y-axis points from the interaction point upward. Because of the detector's symmetry, a cylindrical (z, ϕ, θ) coordinate system is

used. The azimuthal angle ϕ is referred to the x-axis, in the x-y plane; the polar angle θ is referred to the z-axis, in the r-z plane.

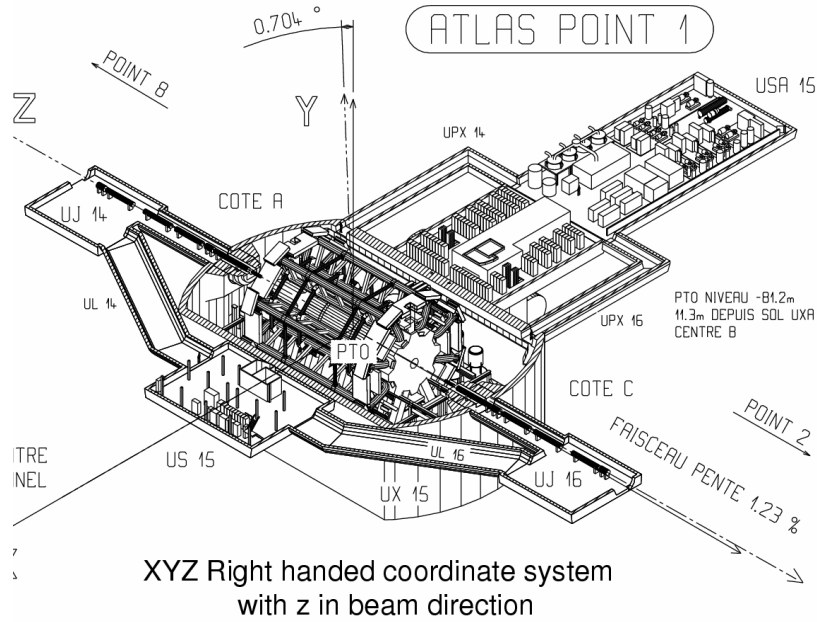


Fig. 1.6 The ATLAS coordinate system

Instead of the polar angle θ , the pseudorapidity variable η is used. The η variable is defined as: $\eta \approx -\ln(\tan(\theta/2))$, and allows to identify four regions in the detector, as shown in fig.1.7: *Barrel*, *Transition*, *End-cap* e *Forward*.

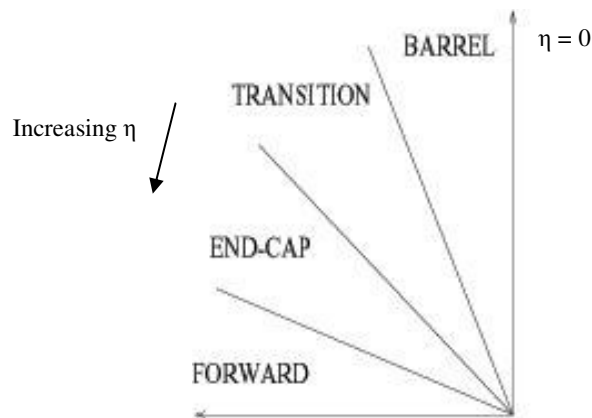


Fig. 1.7 The detector's regions defined by the pseudorapidity variable η

Given the high performances required, the ATLAS detector will have to satisfy some basic design requirements:

- A tracking system with a very high resolution, allowing the reconstruction of secondary decay vertexes of heavy quarks
- A very good energy and direction resolution of the electromagnetic calorimeter, to identify correctly electrons and photons.
- A good hadronic calorimeter to achieve accurate energy measurements in order to perform *jets* identification and missing energy measurements.
- A muon spectrometer able to perform momentum measurements with a very high precision.
- Large acceptance in pseudorapidity (η) and almost total coverage in azimuthal angle (ϕ).
- High trigger efficiency for all interesting processes.

The inner detector

The *Inner detector* [9, 10] is located around the beam's interaction zone and has a cylindrical symmetry, with a length of 6.8m and an external diameter of 2.3m.

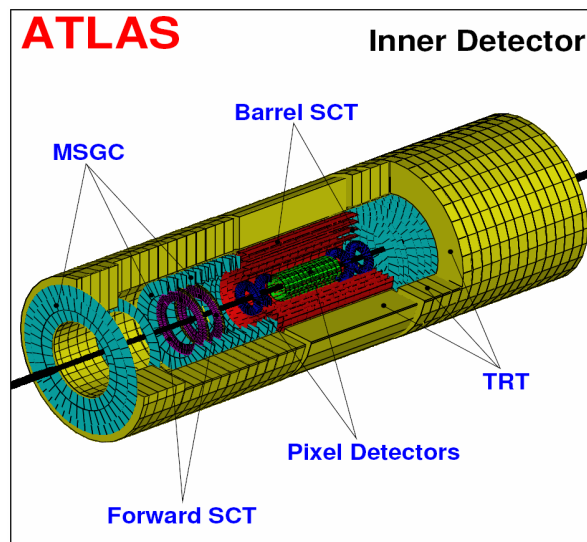


Fig. 1.8 The Inner Detector

The whole inner detector is in a magnetic field of 2 Tesla, parallel to the beam axis, generated by a superconducting solenoid.

The first objective of the inner detector is to reconstruct particles' tracks and to perform measurements of momenta of the charged particles in the apparatus. Moreover, the inner detector will give the position of the primary and secondary vertexes of decays of some short-lifetime particles, such as $B_d^0 \rightarrow J/\psi K_s^0$ or $J/\psi \rightarrow e^+e^-$, allowing to identify the produced particles, such as K or Λ .

The inner detector is made up of two parts: the inner part, with a very high spatial resolution and composed by discrete elements, and an outer part, made by continuous tracking elements. The inner part consists of two concentric structures, one made by silicon pixels and the other made by silicon microstrips, the SemiConductor Trackers; the outer part consists of tube detectors, the Transition Radiation Trackers.

- Pixel detector [11]: is the closest detector to the collision point. Pixels are narrow rectangular silicon regions, with a dimension of $50 \times 300 \mu\text{m}^2$. Pixels are arranged on three cylindrical surfaces around the beam axis, in the *barrel* region, and on five disks centered on the beam axis, in the *end-cap* regions.

Every time a charged particle crosses one of these surfaces, a signal is produced, identifying the fired pixel. The fine granularity of the silicon-pixels ensures an high resolution, $\sigma_{R\phi} \sim 12 \mu\text{m}$ e $\sigma_z \sim 66 \mu\text{m}$.

- *Microstrips detectors* [10]: in this detectors, the elements revealing charged particles' tracks are silicon strips, with a width of $80 \mu\text{m}$. The surfaces with the strips are arranged in the same geometric disposition of the *pixel*, with eight cylindrical strip surfaces in the barrel region and nine disks in the *end-cap* region. The strips are arranged along the beam's axis, in the *barrel* region, and radially in the *end-cap* region. The spatial resolution is $\sigma_{R\phi} \sim 16 \mu\text{m}$ and $\sigma_z \sim 580 \mu\text{m}$.

- *Transition Radiation Tracker* [10]: gas detectors based on the transition radiation produced by electrons tracks. They are tubes with a 4mm diameter and

up to 144cm length, containing a particular gas mixture and with thin wires mounted along the tube axis. There is a constant voltage between the tube surface and the internal wire, and the space between two adjacent tubes is filled with Polypropylene, that emits X photons when interacting with electrons. The gas mixture is ionized when a particle crosses the tube, giving timing information about interesting events.

The front-end electronics of the transition radiation detector tubes has two programmable thresholds. In this way is possible to distinguish signals generated by charged particles from the ones generated by the photons emitted as transition radiation by electrons. This allows a good efficiency in the separation e/π . The tubes are arranged along the beam's axis in the barrel region and radially in the end-cap region, and the spatial resolution is $\sigma_{R-\phi} \sim 170 \mu\text{m}$.

The magnetic system

Magnetic fields are essential in the ATLAS detector, in order to bend the trajectories of charged particles. There are two different structures that generate such fields[12]:

- a superconducting solenoid, producing a 2 Tesla magnetic field for the inner detector;
- an external toroidal superconducting magnet, producing a variable magnetic field from 3 to 6 Tesla (depending from the pseudorapidity) for the muon spectrometer.

The external magnet is made of three toroids, one in the barrel region and two in the end-cap regions, and each one is made of eight independent coils, arranged with an octagonal symmetry. It has an open structure to minimize the contribution of multiple scattering to the momentum resolution; most of the coils is cooled by liquid Helium at 4.5 Kelvin.

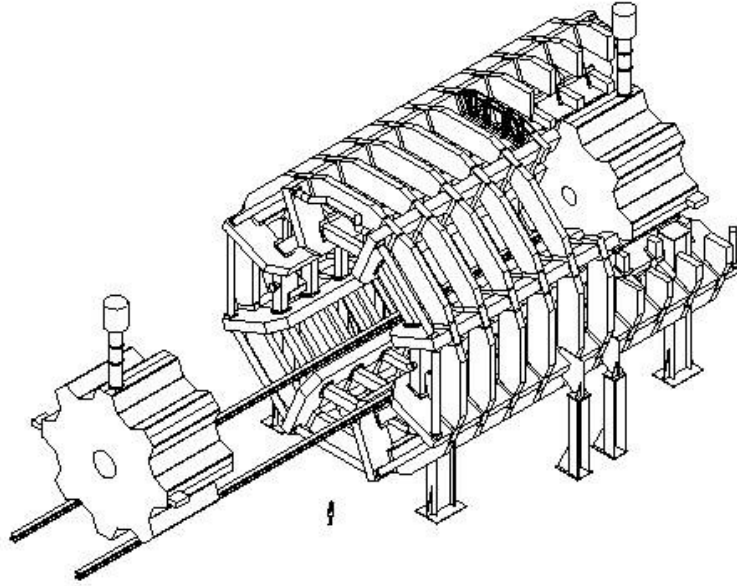


Fig. 1.9 The magnetic system of ATLAS

The calorimetric system

The calorimetric system of ATLAS is made of an electromagnetic (ECAL) calorimeter [13] and of an hadronic (HCAL) calorimeter [14], as shown in fig. 1.10. The ECAL covers a pseudorapidity interval $|\eta| < 3.2$, with a part in the barrel region and a part in the endcap region; the HCAL covers a pseudorapidity interval up to $|\eta| < 4.9$, so up to the forward region of the detector.

The electromagnetic calorimeter

The electromagnetic calorimeter is a detector made of liquid Argon (LAr) and lead (Pb), with a cylindrical structure in the barrel region and two disks in the endcap region. The lead layers and the electrodes used to collect the charges are shaped with an accordion geometry: this configuration allows to reduce the noise of the apparatus and to contain the overlap of signals induced by different events.

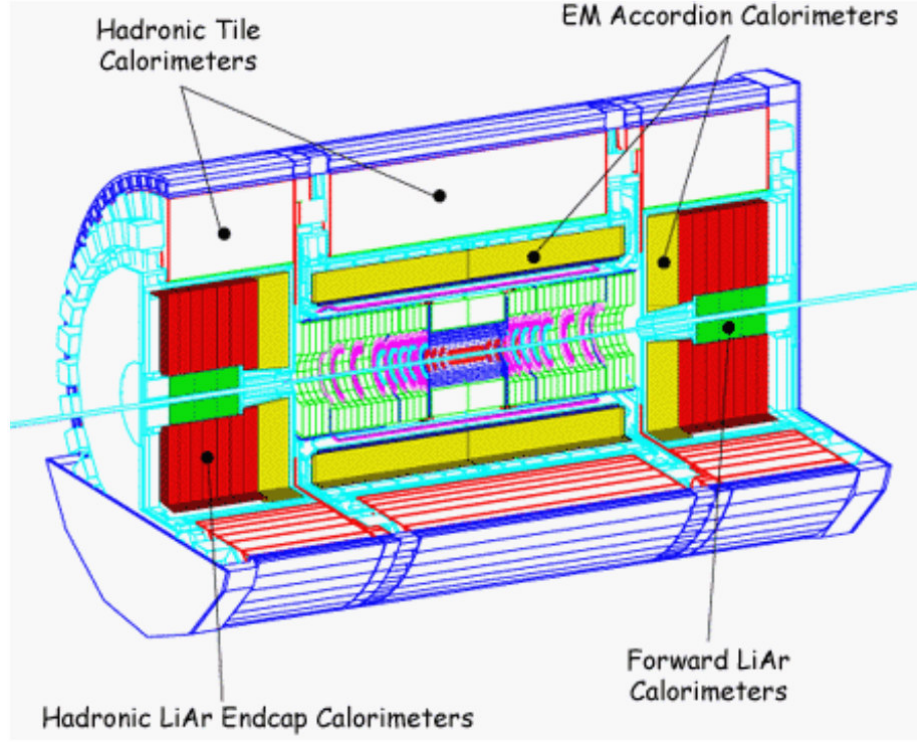


Fig. 1.10 The calorimetric system

The total thickness of the ECAL is greater than $24 X_0$ in the barrel part and greater than $26 X_0$ in the endcap region. The energy resolution of the electromagnetic calorimeter can be expressed as the quadratic sum of three independent terms:

$$\sigma(E)/E = a/\sqrt{E} \oplus b/E \oplus c$$

where a describes the statistic fluctuation, b is the noise term and c takes into account the systematic effects.

At energies between 2 GeV and 5 TeV, typical of events

$$H \rightarrow ZZ \rightarrow 4 \text{ leptons}$$

$$W' \rightarrow l \nu$$

useful in the Higgs boson's search and of supersymmetrical particles, the energy resolution is about $10\%/\sqrt{E}$, with E expressed in TeV.

The hadronic calorimeter

The hadronic calorimeter is made of three parts, one in the barrel region and two in the endcap regions. It covers a pseudorapidity interval more extended than the ECAL, up to $|\eta| < 4.9$.

The discovery of a high mass Higgs boson, decaying into a high p_T

$$W \rightarrow jet - jet$$

requires a $jet - jet$ mass reconstruction and a jet forward tagging. For low mass Higgs search, mass reconstruction via the channel $H \rightarrow bb$ and jet spectroscopy is also important. The different background in the barrel and in the endcap regions impose the use of different technologies: in the interval $|\eta| < 1.7$ iron is used as absorbing material, and scintillating tiles are used as sensible materials; in the interval $1.5 < |\eta| < 4.9$, where a greater particle flux is expected, a technology similar to the one of the ECAL is used, alternating layers of liquid Argon and copper.

The total thickness of the HCAL is 11 interaction lengths. The energy resolution of the hadronic calorimeter can be parameterized as a quadratic sum of three independent terms:

$$\sigma(E)/E = 50\%/\sqrt{E} \oplus 3\% \quad (E \text{ in GeV}) \quad |\eta| < 3$$

and

$$\sigma(E)/E = 100\%/\sqrt{E} \oplus 10\% \quad (E \text{ in GeV}) \quad 3 < |\eta| < 5$$

By this we can see that the energy resolution improves with increasing energy, making them suitable for use in high energy colliders. Moreover, in the design of the calorimeters, the following physics requirements have been considered:

- A good energy resolution in the whole interval covered in η .
- A good linearity in response, on an interval spanning from GeV to TeV.
- Uniformity of performances both in η and in ϕ .

The Muon Spectrometer

In the outer part of the ATLAS apparatus there is the muon spectrometer [15]. It has a volume of about 16000 m^3 and plays a double role: to perform accurate measurements of position and momentum and to produce the muon trigger for the experiment. The muon spectrometer requirements are:

- A momentum resolution lower than 10%, up to values of transverse momentum $p_T \sim 1 \text{ TeV}/c$.
- A spatial resolution of $\sim \mu\text{m}$ in the measurement of the particles' position in the η direction and lower than 10 mm in the ϕ direction.
- A good performance in reconstructing events whose final states are characterized by the presence of 2 or 4 muons, that are events that can be related to a Higgs boson decay.
- A trigger system selectivity up to $p_T > 20 \text{ GeV}/c$.

Tab. 1.1 shows some of the parameters of the muon spectrometer design.

Parameter	Event	Objectives	Comments
$\Delta p_t/p_t$ a 20 GeV/c	$H \rightarrow ZZ^* \rightarrow 4l$	1-2 %	Resolution limited by energy losses and multiple scattering
$\Delta p_t/p_t$ a 75 GeV/c	$H \rightarrow ZZ \rightarrow 4l$	1 -2 %	
$\Delta p_t/p_t$ a 100 GeV/c	$Z \rightarrow \mu^+\mu^-$	< 10 %	Resolution limited by costs optimisation

Tab. 1.1 Parameters of the muon spectrometer design

The basic principle of the spectrometer is that the trajectories of charged particles moving into a magnetic field are curves. The magnetic field lines in the spectrometer are parallel to the ϕ direction: hence, the component of the particles' momenta in the ϕ direction is parallel to the magnetic field lines. If we define a cylindrical coordinates system, particles' trajectories are arcs of circumference in the r - z plane. The radius and the direction of the curvature of the trajectory in the r - z plane are depending, given the magnetic field value, by the momentum and by the charge of the particle. By reconstructing the trajectory, the charge and the

momentum of the particle can be calculated. An important parameter is the *bending power* of the magnetic field, the integral $\int B \cdot dl$ of the azimuthal component. Fig 1.11 shows the integrated azimuthal component of the magnetic field versus the pseudorapidity coordinate η .

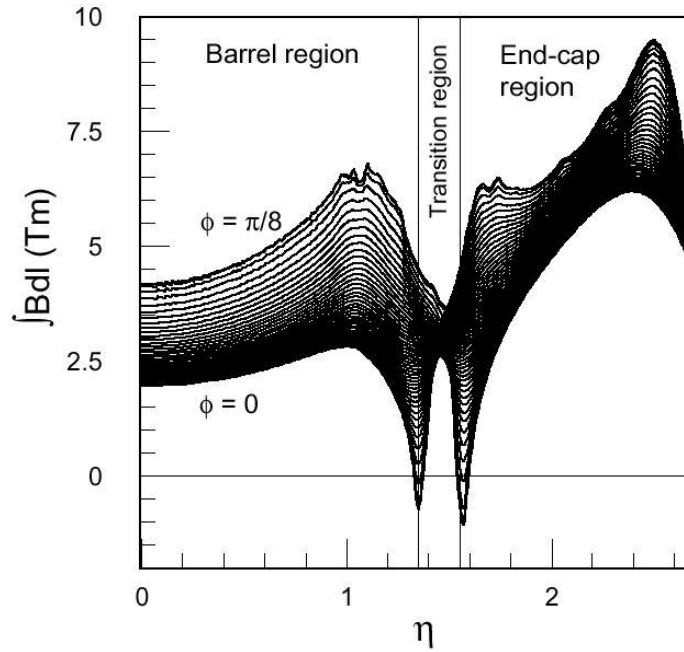


Fig. 1.11 The integrated azimuthal component versus the pseudorapidity coordinate η

In order to achieve high momentum resolution, up to energies ~ 1 TeV, there is the need of measuring the sagitta of the muon trajectory very accurately; the trajectory is sampled in three measuring stations with a resolution lower than ~ 50 μm . Moreover, the surface that must be covered by detectors is very large, up to ~ 5000 m^2 , so reducing the costs of the detectors is crucial.

In order to satisfy the different muon spectrometer requirements, such as the high spatial resolution of the precision chambers, the low cost of production, the radiation hardness, the low response time of the trigger detectors, four different kinds of gaseous detectors have been designed, both for precision measurements and for trigger chambers. The disposition of the detectors is shown in fig.1.12.

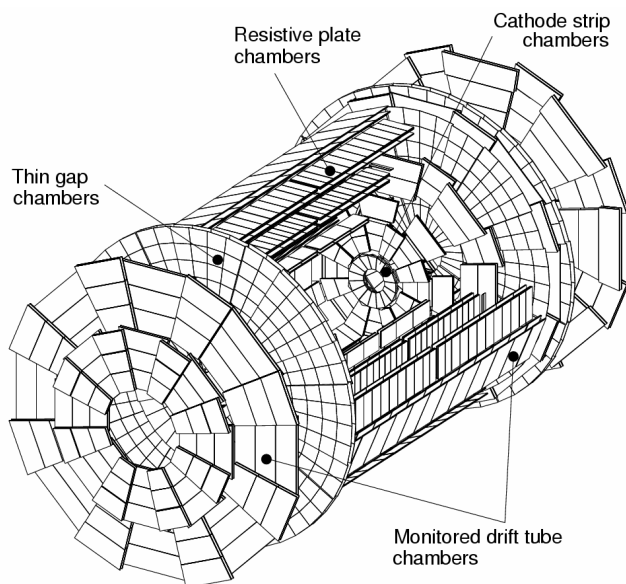


Fig. 1.12 The disposition of the muon spectrometer chambers

In the barrel region, both precision chambers and trigger detectors are arranged in three cylindrical concentric surfaces, mounted at a distance of about 5m, 7.5m and 11m from the beam axis, as shown in fig. 1.13; in the two endcap regions the detectors are located on four vertical stations, at a distance of ~ 7 m, 10m, 14m and 22 m from the interaction point.

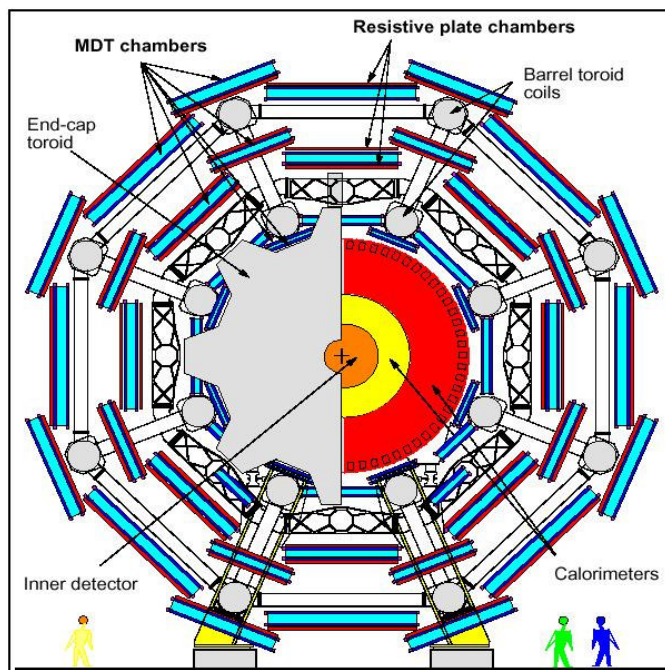


Fig. 1.13 The spectrometer structure in the *barrel* region (ϕ projection)

For position measurements, *Monitored Drift Tubes* (MDT) are used in the barrel region and *Cathode Strip Chambers* (CSC) are used in the region with high pseudorapidity ($|\eta|$ up to 2.7).

For trigger measurements, stations of *Resistive Plate Chambers* (RPC) are used in the region with $|\eta| < 1.05$, and *Thin Gap Chambers* (TGC), multiwire proportional chambers, are used in the transition and in the endcap regions.

Fig. 1.14 shows the momentum resolution of the apparatus at $|\eta|=0$. It is influenced by the intrinsic resolution of the detector, by the multiple scattering, by the statistic fluctuations of the energy loss and by the chambers alignment. For example, for momenta up to 250 GeV/c, the resolution is affected mainly by the multiple scattering and is about 2%; for momenta below 10 GeV/c, the fluctuations of the energy loss of the muons in the calorimeters limit the resolution to about 6-8%. The momentum resolution is lower than 10%, also at energies of ~ 1 TeV.

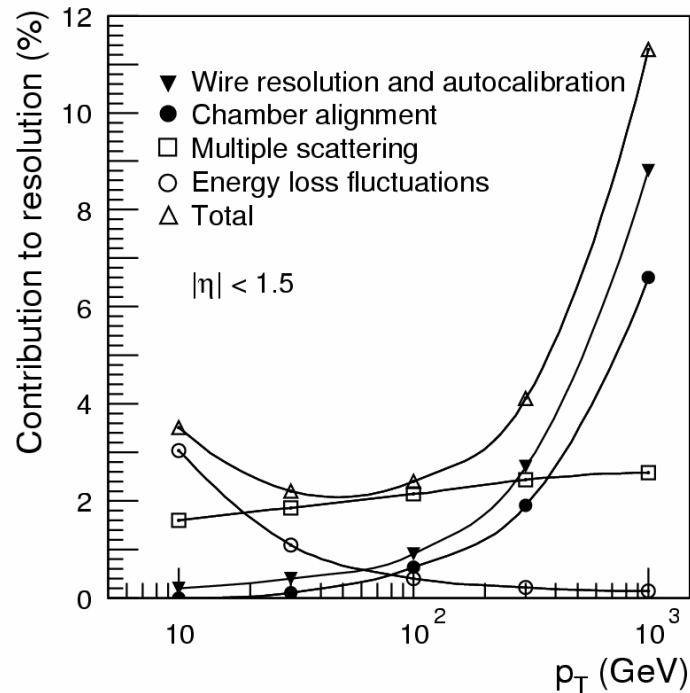


Fig. 1.14 Momentum resolution of the spectrometer at $|\eta|=0$.

Monitored Drift Tubes (MDT)

MDT detectors [16] are cylindrical drift tubes with a 30 mm diameter and with a 50 μm W-Rn wire (tungsten e wrenio); the gas is a non flammable mixture of Ar(91%)-CH₄(4%)-N₂(5%). The drift time is lower than 480 ns, and the average spatial resolution for a single tube is 80 μm .

In order to enhance the spatial resolution, each MDT chamber is made of two tracking planes, each made by a superposition of 3 or 4 layers of drift tubes. Fig. 1.15 shows the structure of a single MDT chamber. The readout electronics is made of a differential amplifier, a shaper and a discriminator.

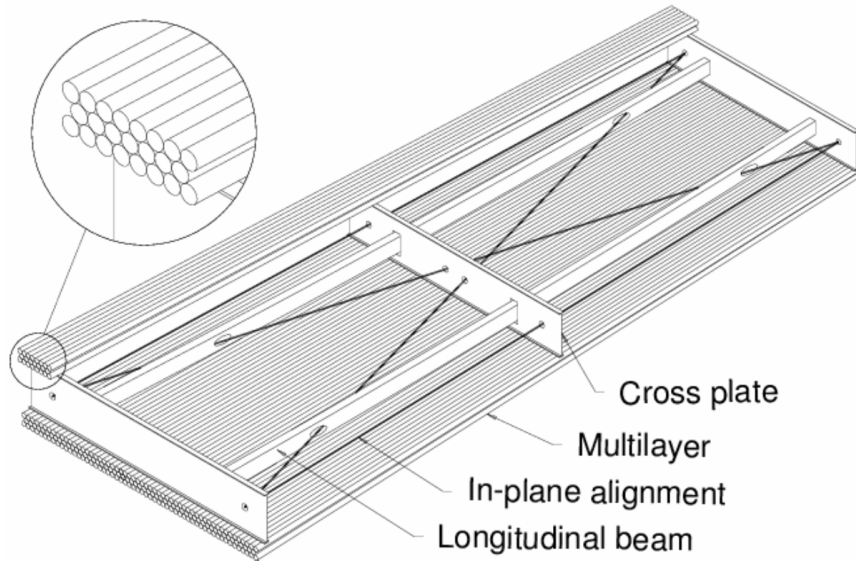


Fig. 1.15 the disposition of tubes in a single MDT chamber

The geometry and the position of the detectors is *Monitored* by an optical system called RASNIK [17] with a precision better than 30 μm . According to the azimuthal symmetry of the magnet, the spectrometer is divided in 16 sectors in the ϕ projection and every sector is completely covered by a chamber. The drift tubes are installed perpendicularly to the beam axis: so the chambers can measure muon's position in the η projection, reconstructing the track in the r-z plane. There is a superposition of 200 mm between two chambers of adjacent sectors, to avoid dead regions in the apparatus due to the presence of reinforcement structures and cables.

Cathode Strip Chambers (CSC)

Cathode Strip Chambers [18] are proportional multiwire chambers with cathode strips readout: are used to achieve a better spatial resolution, because they have to reconstruct tracks of particles with higher transverse momentum. A precision measure is obtained by evaluating the charge on the cathode (made by several strips), induced by the avalanche formed on the anode wire. The gas is a mixture of Ar(30%)-CO₂(50%)-CF₄(20%), the wires are supplied by 2,6 kV and the gas gain is about 10^4 . The cathode strips are 5.08 mm wide and are arranged orthogonally to the anode wires, having a pitch of 2.54 mm; the layout is shown in fig.1.16. The spatial resolution obtained with this readout scheme is lower than 60 μm . As the MDT chambers, a CSC chamber is made of two planes, each made of four layers of detector.

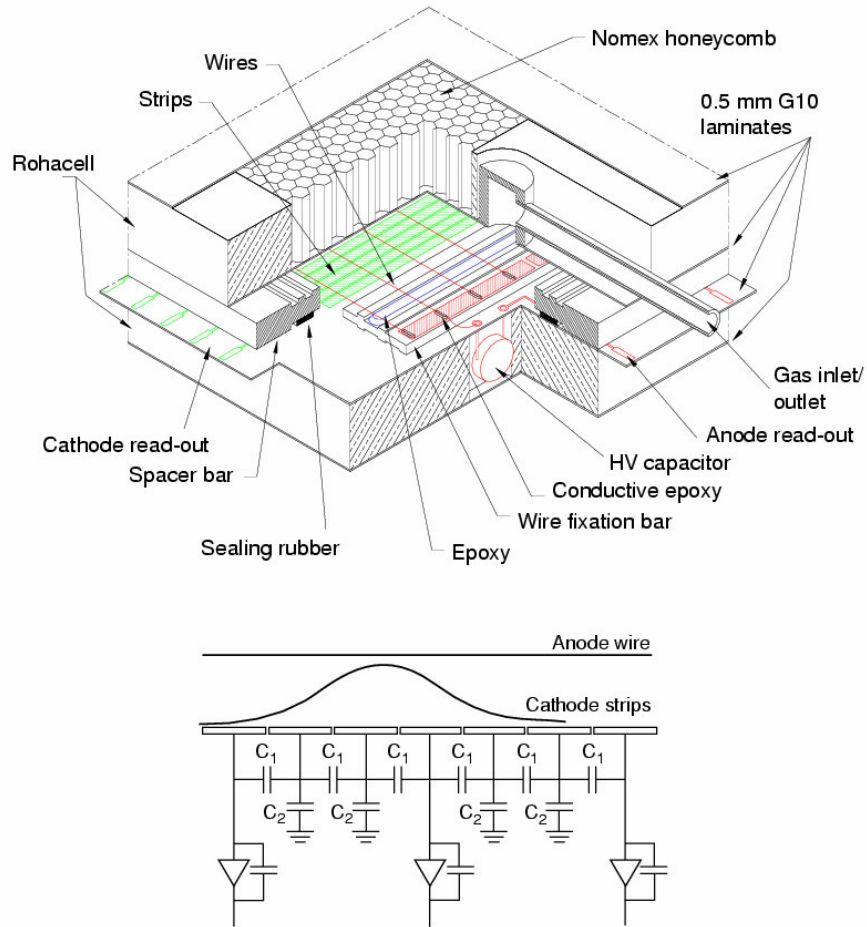


Fig. 1.16 The layout of a CSC detector and a simulation of the charge distribution

CSC detectors are used at about $\sim 7\text{m}$ from the interaction point, in regions with high pseudorapidity, where a high flux of particles is expected. The main characteristic of the CSC detectors are a small drift time of the electrons (30 ns), a good timing resolution (7 ns) and a low sensitivity to the neutrons. Moreover, using strips orthogonal to the cathode strips, a measure of the second coordinate can be also performed.

Resistive Plate Chambers (RPC)

In the barrel region of the apparatus, for trigger purposes, *RPC* (*Resistive Plate Chambers*) detectors [19, 20, 21] are installed. RPCs are detectors that guarantee high performances. Their typical timing and spatial resolution are respectively 1 ns and 1 cm with an intrinsic detection efficiency greater than 98 %. The Research Group in Subnuclear Physics of the Physics Department of the “*University of Naples Federico II*” and the Group I of the *I.N.F.N.* Section of Naples are involved in the development of the ATLAS RPC [22] chambers.

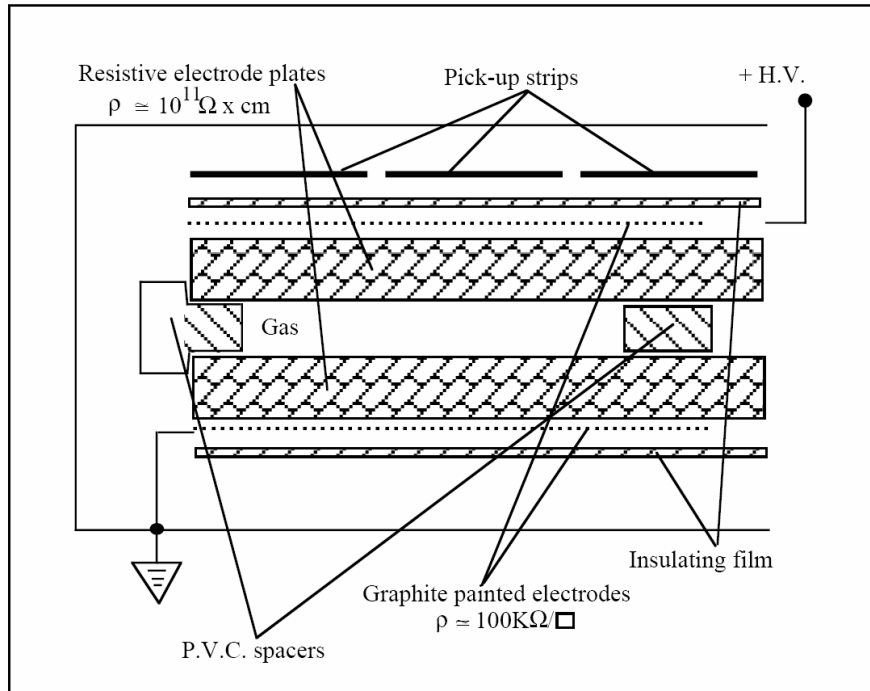


Fig. 1.17 An RPC detector scheme

RPCs are ionization detectors whose working principle is based on the discharge inside gas. Fig. 1.17 shows the cross section of an RPC detector. Two bakelite

planes delimit a gas gap in which a uniform electric field is applied, whose intensity is about 4.5 kV/mm. The bakelite planes, also called resistive plates because of their high resistivity ($\rho = 10^{10} \Omega \times \text{cm}$), are externally coated with two thin graphite layers, connected respectively to high voltage and to ground.

A thin layer (few hundred microns) of PET is glued on the graphite electrodes, to insulate the high voltage electrodes from the read-out strips, oriented in the X and Y directions. The graphite electrodes, because of their high resistance, are transparent to the electrical pulse created inside the gas gap, when a charged particle crosses the gap itself. For this reason, the signal can be read by induction on the read-out metallic plates, usually copper strips.

The RPC just described has the structure of a two dielectric planar capacitor. The working model of the chamber is shown in fig. 1.18. The capacitors C e C_{gas} and the resistors R e R_{gas} describe the capacitance and the resistance of the bakelite planes and of the gas gap, respectively.

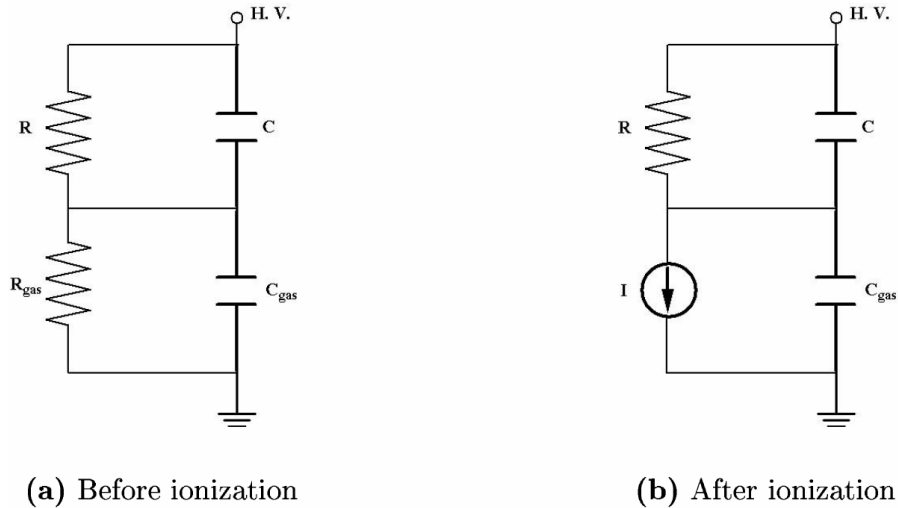


Fig. 1.18 Working scheme of an RPC detector, in the steady condition (a) and in the case of a particle crossing the gas gap (b)

In steady condition, when there is no ionization, the gas has infinite resistance ($R_{\text{gas}} = \infty$). So the high voltage HV is entirely applied on the gas gap. When there is an interaction with charged particles, there is the production of electrons by ionization and the gas gap behaves like an ideal current generator. This current

generator discharges the “gas capacitor” C_{gas} and some of the HV is gradually transferred to the resistive plates, described by the C capacitance. The system goes back to the initial conditions following an exponential law with a time constant: $\tau = R_{el} (C + C_{gas}) = \rho \epsilon_0 (\epsilon_r + 2d/g)$, where ϵ_r is the dielectric relative constant of the bakelite, d is the thickness of the bakelite plates and g is the thickness of the gas gap. For $\rho = 10^{10} \Omega cm$, the previous relationship gives $\tau \approx 10 ms$. This value has to be compared with the time of production of the discharge, that is only ~ 10 ns. In this time interval, the electrode plates behave like insulators and the voltage on the gas gap is too low to feed the discharge. This quenching mechanism is at the basis of the working principle of the detector.

In a simple model [23] a chamber with resistive electrode plates can be divided in a large number of small “discharge cells”, each one independent from the others. An estimation of the extension of the area where the discharge is located is given by the formula $S = Q g / \epsilon_0 V$, where Q is the charge delivered in the gas. Therefore, the detector is “blind” only in a region of extension S , for a time τ . By this, the total charge produced inside the detector, and so the intensity of the electric field, have a direct influence on the rate capability of an RPC. A small value of Q has two important effects: it allows to have small currents in the detector (a crucial point to prevent detector ageing) and to have a high detecting efficiency, also when there is an high flux of ionizing particles.

The current produced by the discharge in the gas induces a signal on the pick-up electrode. The pick-up electrodes can be shaped as strips or square pads. The strip have the advantage to behave as signal trasmission line of well defined impedance, allowing the trasmission at large distances with a small loss of amplitude and timing information. Two ortogonal sets of strips are used; the pitch of every strip is 2-3 cm. Two contiguous strips are separated by a 0.3mm wire, connected to ground to reduce the capacitive coupling of two adjacent strips.

The equivalent circuit (see fig. 1.19) of the read-out electrode can be described as a current generator, charging a capacitor C in parallel to a resistor R , where C is the electrode’s capacitance and R is the resistance between the electrode and the ground.

Typical values for C and R are: $C \sim 1$ pF and $R \sim 25 \Omega$; the time constant of the circuit results to be $\tau' = 25$ ps, much smaller than the rising time of the signal. So the read-out signal is not integrated, but the current circulating in the read-out electrode is, every instant, proportional to the current produced by the discharge inside the gas.

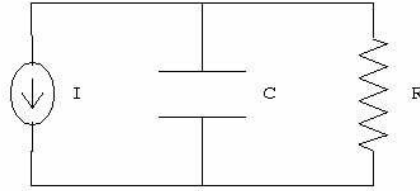


Fig. 1.19 The equivalent circuit of the read-out electrode of an RPC strip

Thin Gap Chambers (TGC)

Thin Gap Chambers [24, 25] are used for trigger purposes in the endcap regions because of their very good rate capability and their ageing characteristics. TGC are, like CSC detectors, multiwire proportional chambers, with a smaller (1.4 mm) distance between cathode and wire plane compared with the distance between wires (1.8 mm), as shown in fig. 1.20. The gas mixture is made of 55% CO_2 and of 45% n-pentane ($\text{n-C}_5\text{H}_{12}$) and the chambers operate at an high voltage of about 3 kV. The small distance between wires and the electric field configuration ensure a short drift time and a good timing resolution of about 4 ns.

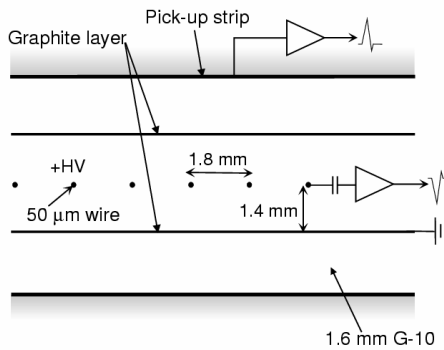


Fig. 1.20 A scheme of a TGC detector

The anode wires, with a 50 μm diameter, are arranged parallel to the MDT wires and produce the trigger signal; read-out strips are orthogonal to the wires and are used to measure the second coordinate. So, TGC chambers have both a trigger purpose, in the endcap region, and a precision chambers purpose (for the measure of the second coordinate) in the forward region, in the inner and middle stations of the spectrometer. In particular, in the endcap region, the detectors are installed at a distance of about 14 m from the interaction point and placed on three different planes to form, as shown in fig. 1.21, a triplet (M1) and two doublets (M2 and M3) of chambers.

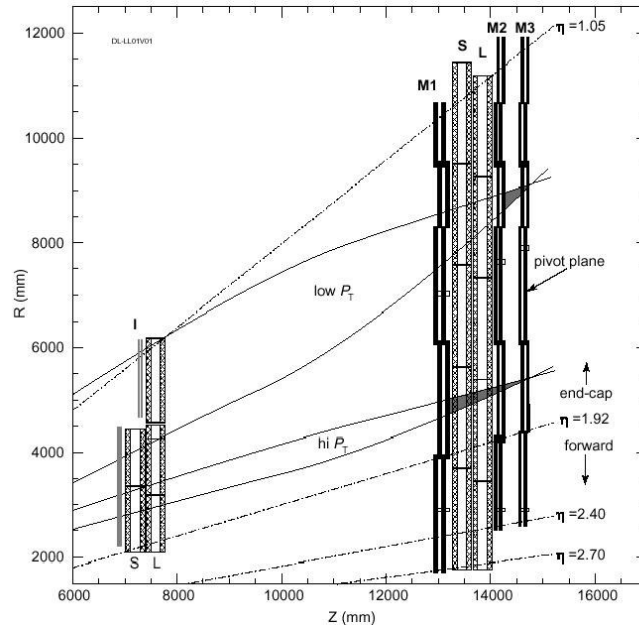


Fig. 1.21 The TGC chambers disposition in the endcap region

The Trigger and the Data Acquisition System

Because of the enormous expected background, the design of an extremely selective trigger is needed. The interesting physics processes must be accepted with high efficiency, while the not interesting ones must be very efficiently rejected. For example, at the nominal luminosity of LHC, searching for a Higgs boson with a mass of $\sim 80 - 100$ GeV, one interesting event is expected, every 10^{13} produced.

This is achieved by defining three different trigger levels (LVL1 [26], LVL2 and LVL3 [27], also called *Event Filter*). The first level receives data only from the detector, whilst each of the two following levels elaborates data from the detector and also the information collected at the previous level. The architecture of the trigger system [28] and the three levels of the trigger are shown in fig. 1.22.

The first level trigger must identify, without ambiguity, the bunch crossing of the interesting event, using only data coming from the calorimeters and from the spectrometer, and uses synchronous parallel processors working at the bunch crossing frequency, 40 MHz. Programmable logic devices (*FPGA* e *CPLD*) and *ASIC* (*Application Specific Integrated Circuit*) are used.

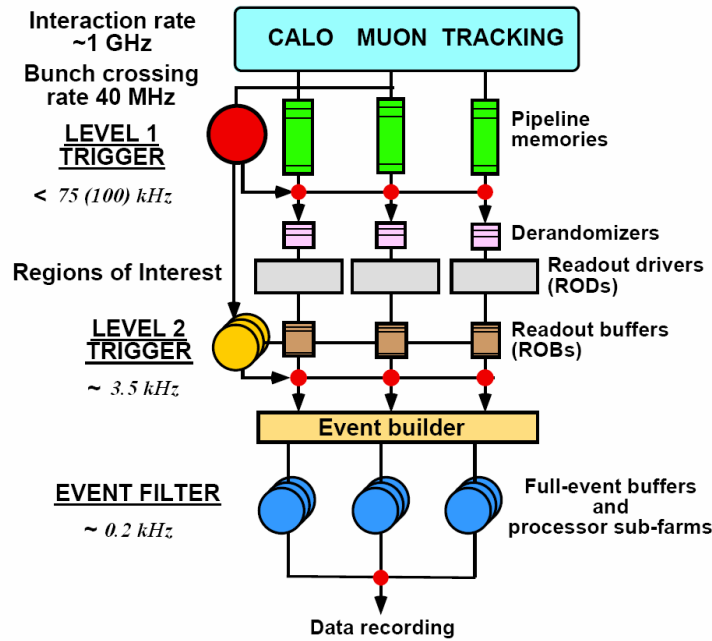


Fig. 1.22 The three levels of the trigger system

Data produced by the detector are elaborated by the level-1 trigger, with a latency lower than 2.5 μ s; during this time, information are stored in “*FIFO*” (*first-in first-out*) memories; the output frequency of data is 75 KHz, increasable up to 100 KHz.

In particular, the level-1 trigger structure, as shown in fig. 1.23, is composed by four subsystems:

- the *Muon Trigger Processor* receives data from TGC and RPC, respectively from the *end-cap* and the *barrel* regions; it detects events that have in the final states muons with a transverse momentum greater than a programmable threshold and identifies the spatial region the muon comes from.
- the *Calorimeter Trigger Processor* identifies electrons, photons, hadrons and *jets*, and measures the possible missing energy.
- the *Central Trigger Processor* collects and elaborates information from the trigger processors.
- the TTC system (*Timing Trigger Control*) generates and distributes the control signals to the whole apparatus.

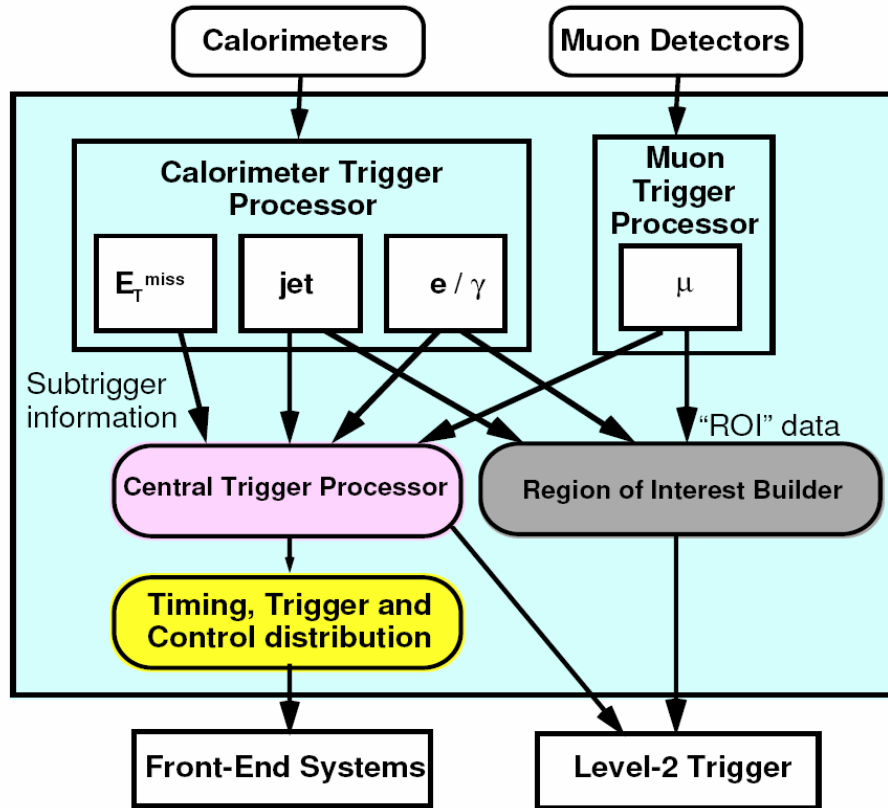


Fig. 1.23 The structure of the level-1 trigger

When an event is accepted by the LVL1, a level-1 trigger signal is generated by the trigger processor: in this case, data stored in the *LIFO* buffer are transferred to the next elaboration levels. Data are read out from the front-end electronics

systems of the detectors into *Read Out Drivers* (RODs) and then forwarded to *Read Out Buffers* (ROBs). Fig. 1.24 shows the path of data from the detector to the elaborators, for the RPC detectors.

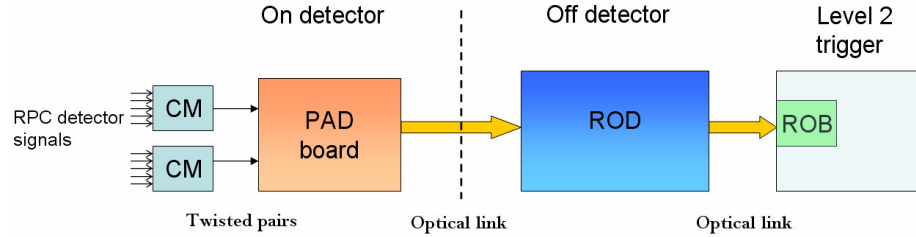


Fig. 1.24 The transmission of data from the RPC detectors to the counting room

The trigger system is divided into Regions of Interest (*ROI*), with a granularity $\sim \Delta\eta \times \Delta\phi = 0.1 \times 0.1$. For example, in the barrel region, the trigger system is divided into 1664 *ROI*. The level-1 trigger generates and transmits to the level-2 trigger processors the coordinates of the ROIs where an interesting event occurred. In this way, the level-2 trigger accesses only to that ROIs, reducing the flux of data to be elaborated. At this stage, full granularity and full precision data from most of the sub-detectors are available and are used by the LVL2 to select interesting physics events. Moreover, even if it is possible to access all the data of the event, only the information needed to decide whether accept or reject that event are acquired.

Data for the bunch crossing selected by the LVL1 trigger are stored in the ROBs during the LVL2 trigger latency (expected in the range of 1-10 ms). Then, the LVL2 trigger promotes data to the following trigger level, if they are related to an interesting event, or deletes them. The data transfer from the *ROBs* to the *Event Filter*, that is the third trigger level, is called *event building*.

Whilst the level-1 trigger is based on an hardware designed for the specific application, both level-2 and Event Filter use commercial processors; moreover, the structures of computing and communication are quite similar. The differences between the level-2 and Event Filter are that level-2 trigger needs simple and fast algorithms, and the level-3 uses elaboration algorithms similar to the ones used for the *off-line* analysis.

The level-2 trigger system is made of a *sub-farm* of commercial processors and the expected trigger frequency is ~ 3.5 KHz. An event accepted by the level-2 trigger is built by the *Event Filter*. The *Event Filter* uses offline algorithms, based on up to date calibration and alignment information, such as the magnetic field map. Most of the rejection power of the *Event Filter* comes from the use of complex algorithms and criteria that cannot be performed by the LVL2 trigger, because of processing time limits. The Event Filter sends data to the mass memories with a bandwidth of $\sim 10 - 200$ Mbyte/s; a single event size should be ~ 1 Mbyte, so the output frequency of the Event Filter should be of ~ 200 Hz.

Chapter 2

The level-1 trigger and the Data Acquisition system of the ATLAS muon spectrometer

In this chapter the architectures of the first level trigger and of the Data Acquisition system (*DAQ*) of the ATLAS muon spectrometer are described. The design of the ROD, that has been developed in this PhD work, is deployed in the trigger system and in the data acquisition system architecture of this section of the ATLAS apparatus. Details will be given on the synchronization system of the ATLAS apparatus with the LHC machine and on the distribution system of the control signals. Moreover, the data acquisition system and transmission from RPC detectors to the counting room *USA15* will be described.

The signals of the RPCs of the muon spectrometer undergo several elaboration steps before to be sent on optical fibre to the counting room *USA15* and used by the next levels of acquisition and processing. The algorithms of the coincidence matrix, for the generation of the trigger, and the structures of the other acquisition boards are described. The transmission system on optical fibre is presented, whose

receivers will produce data in input to the ROD, that is described in this thesis work.

The structure of the spectrometer

The spectrometer for the detection of muons is built in the outer region of the ATLAS apparatus.

The spectrometer is a tracking system in a toroidal magnetic field, composed by precision chambers and trigger detectors. In the barrel ($|\eta| < 1,05$) region of the apparatus, the magnetic field is generated by a system of coils radially assembled. The field has variable intensity, depending by pseudorapidity, in the 0.5-2 T interval.

In order to fulfill the different requirements of the muon spectrometer, such as high spatial resolution, low cost of production, radiation hardness, low response time of the trigger detectors, four different kinds of gaseous detectors have been deployed, both for precision measurements and for trigger chambers.

For position measurements, *Monitored Drift Tubes* (MDT) are used in the barrel region and *Cathode Strip Chambers* (CSC) are used in the region with high pseudorapidity ($|\eta|$ up to 2.7).

For trigger measurements, stations of *Resistive Plate Chambers* (RPC) are used in the region with $|\eta| < 1.05$, and *Thin Gap Chambers* (TGC), multiwire proportional chambers, are used in the transition and in the endcap regions.

The trigger system in the barrel region

As said in the chapter I, a crucial objective for the success of the ATLAS experiment is an extremely selective and efficient trigger system. Due to a remarkable background of events induced by muons in the spectrometer, the requirements are:

- reconstruction of the track and discrimination of the event, in relation to the transverse momentum, corresponding to the highest allowed LVL1 trigger latency of the experiment, which is 2.5 μ s;

- identification of the *bunch-crossing* of an interesting event;
- identification of the Region Of Interest ROI of an interesting event;
- high acceptance in pseudo-rapidity ($|\eta| < 2.4$).

For the physics researches at the LHC, two kinds of interesting events should be taken in account in the design of the trigger system of the spectrometer:

- the production of muons with values of transverse momentum p_T , respect to the beam axis, greater than 6 GeV/c, in low luminosity regime, that will be indicated as “*low p_T* ” events;
- the production of muons with values of transverse momentum p_T greater than 20 GeV/c, in high luminosity regime, that will be labelled as “*high p_T* ” events.

As already said, the precision chambers of the apparatus are not suited to achieve all of the indicated targets. For example, the MDT detectors have high response time (~ 500 ns), much greater than the time separation of the *bunches* of particles at the LHC (25 ns). Therefore, in the design of the spectrometer there is the need for trigger detectors with spatial resolution of about 1 cm and high resolution in time. Moreover, the presence of a significant background in the apparatus, up to 60 Hz cm^{-2} in high luminosity activity, requests at least two planes of detectors in coincidence, to validate the event. The adopted scheme should also allow to produce chambers on large scale, in order to cover large regions, up to 3650 m^2 , at relatively low costs.

The trigger algorithm for muons detection

The aim of the trigger system of the muons spectrometer of the ATLAS apparatus is the detection of particles coming from the interaction vertex having a transverse momentum greater than a certain programmable threshold. Moreover, the trigger latency of level 1 must be lower than $2.5 \mu\text{s}$.

As already said, the trajectories of the muons are deflected in the apparatus by the toroidal magnetic field. The curvature will depend, among other factors, by the momentum of the particle. Let's suppose, as depicted in fig. 2.1, that our purpose is to detect the position of a muon in two different chambers of the

spectrometer. Let $P1$ and $P2$ be the impact points respectively in the first and in the second chamber. Also let $P3$ be the impact point in the first station of detectors of the trajectory crossing $P2$ and coming from the interaction vertex, in the limit $p = \infty$. The distance between the points $P1$ e $P3$ depends from the curvature radius of the trajectory and hence from the momentum of the muon. If we define a cone with vertex in $P2$, axis $P2P3$ and opening angle α , all the particles detected in the first station inside the cone have a momentum greater than a certain threshold p_t , related to the angle α .

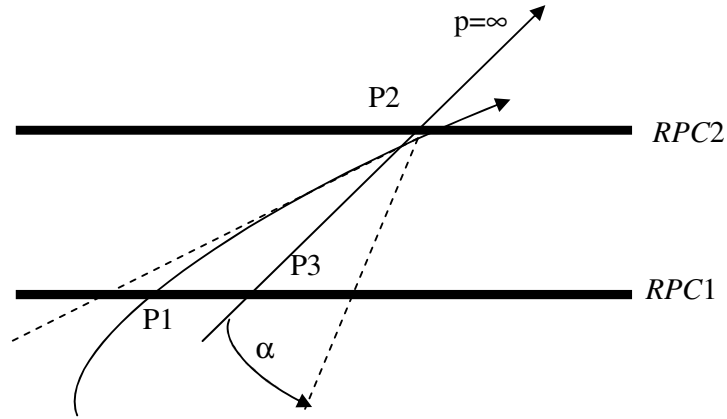


Fig. 2.1 The principle of the trigger system: the radius of curvature of a trajectory depends on the particle momentum

The greater is the angle, the lower is the threshold. In the spectrometer, the position of the particle coincides with the position of the read-out electrode on which the signal is induced. The trigger algorithm is then based upon the signals of the two planes of detectors taken with the opportune timing coincidences (fig. 2.2). The coincidence conditions can be written into a programmable memory that is just a matrix whose row and column indexes represent the positions of the strips on the detectors. The matrix, programmed depending on which trigger condition is chosen, will indicate the validity (or not) of the coincidence between the signals corresponding to the matrix elements. The ability of carrying out in this way the trigger algorithm in *hardware*, allows considerably reducing the elaboration time of the informations. The trigger system of the muon spectrometer of the ATLAS apparatus exploits the idea just described, which will be discussed in the next paragraphs.

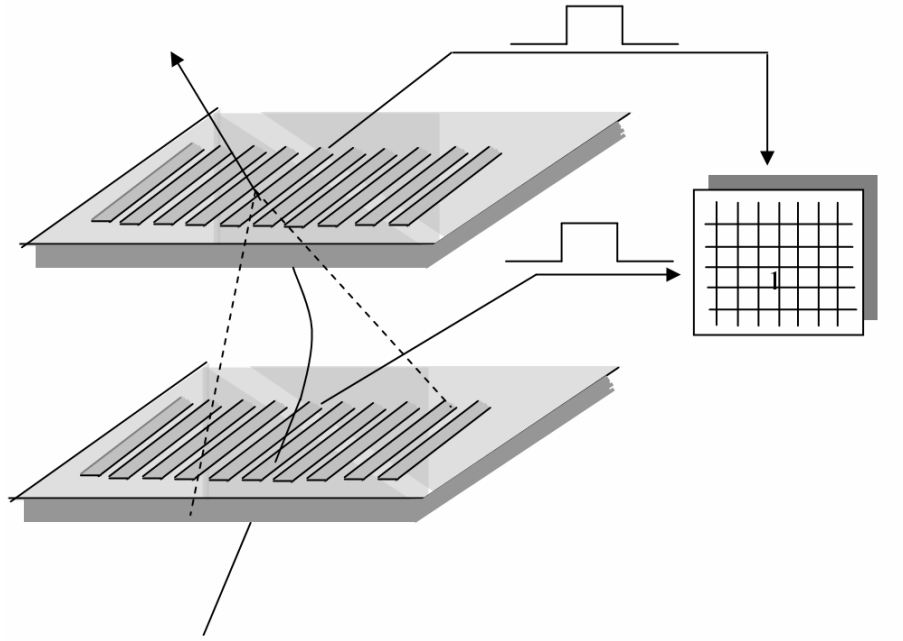


Fig. 2.2 A coincidence matrix between the read-out strips of the RPCs allows to select the muons' trajectories

The read-out electronics

Fig. 2.3 shows the longitudinal section of the spectrometer in the barrel region. The RPC chambers are assembled on three different stations in each of the 16 sectors of the spectrometer, 8 “*Large Barrel Sector*” and 8 “*Small Barrel Sector*”.

The RPC1 and RPC2 stations are installed just on the top and the bottom of the second station of the MDT precision station labelled in fig.2.3 as *BMS* (*Barrel Middle Small*); the RPC3 station is at a greater distance from the beam axis, under the *MDT BOS* (*Barrel Outer Small*) chamber.

The RPC detectors have the same dimensions of the corresponding MDT chamber; so every chamber will cover a whole sector, in the azimuthal direction.

A “*low p_T* ” event ($p_T > 6 \text{ GeV/c}$) is detected if, as described above and in the previous paragraph, a muon is identified in the RPC1 station and a signal is detected in the RPC2 station, in a window with a programmable width depending on the transverse momentum threshold. An “*high p_T* ” event is detected if a “*low p_T* ” coincidence is found and also a signal in the RPC3 station is detected. Both for “*high p_T* ” and “*low p_T* ” events, a coincidence in time between signals is

requested to be in a 20 ns interval. In order to reduce the background in the apparatus, the trigger algorithm is applied in both η and ϕ projections in which the trajectory is sampled.

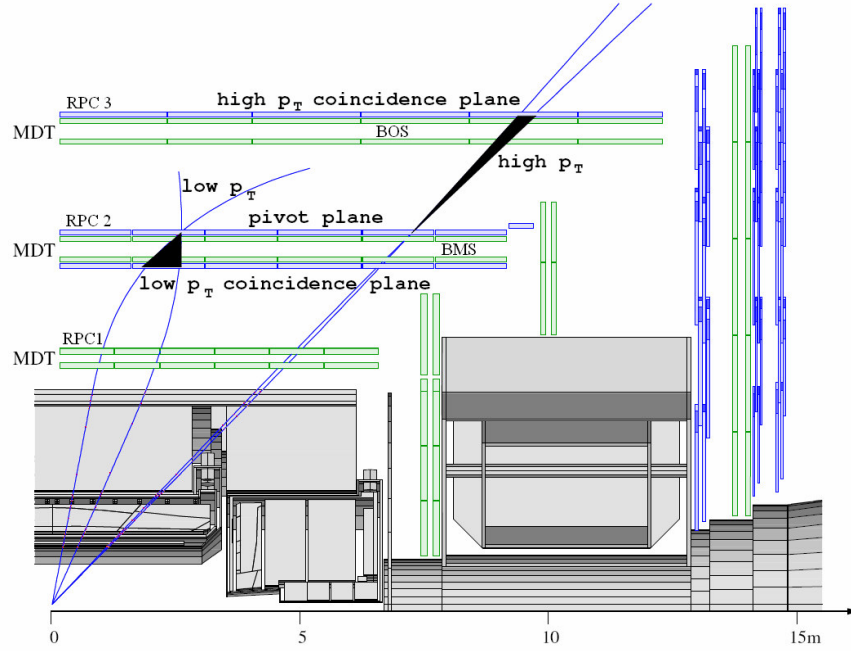


Fig 2.3 Longitudinal section of the trigger system

Each RPC chamber is usually made of two units, longitudinally overlapped, as shown in fig. 2.4. The greatest dimensions of the various units are limited by the propagation delay of the signals on the read-out strips of the chambers. This delay must be lower than 11 ns, to allow the correct identification of the *bunch-crossing* of the event, that is one of the target of the muon spectrometer trigger system.

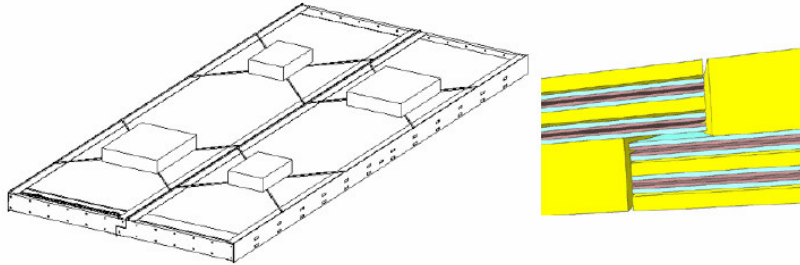


Fig. 2.4 The structure of an RPC chamber, made by two RPC units joined together. On the right a zoom of the superposition region

Each unit is made of a polystyrene panel that supports, as shown in fig. 2.5, two planes of detector made of bakelite layers (2mm thin) separated by insulating polycarbonate spacers. The gas gap is 2mm wide and is sealed with epoxy glue.

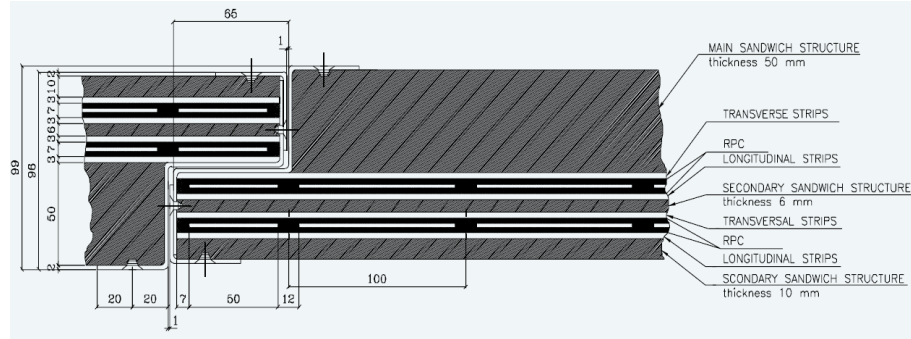


Fig. 2.5 The detailed scheme of an RPC unit

The gas is a mixture of $C_2H_2F_4 - C_4H_{10}$, in the proportions 96,7 - 3 %. Moreover, the presence of a concentration lower than 1% of SF_6 reduces the probability of producing ‘streamer’ in the chamber. By reducing the charge produced by secondary ionization, the detector is able to deal with greater radiation fluxes, up to $\sim 1 \text{ kHz} \times \text{cm}^{-2}$. The bakelite layers are painted with graphite, that creates a thin film connected to the high voltage of the chamber; on the film also the readout planes are installed, isolated by a polyethylene sheet.

Each detector plane is read out by two series of strips, one on every side and mutually orthogonal. The strips oriented in the azimuthal direction, parallel to the magnetic field lines and to the tubes of the MDT chambers, measure the coordinates in the η projection, in the r-z plane. From now on, such strips will be indicated as η strips. The strips oriented in the longitudinal direction, and so parallel to the beam axis, measure the azimuthal coordinates in the ϕ projection, in the x-y plane. Such strips will be indicated as ϕ strips.

The strips are made by copper and have a variable width, between 30 and 40 mm. They are terminated on their characteristic impedance (25Ω) and separated by thin strips tied to ground to decouple the signals on adjacent channels. The structure of the readout system is shown in fig. 2.6. The read-out electronics is made of three stage amplifier and a comparator. The circuit allows detecting

signals with an amplitude of $\sim 50 \mu\text{V}$. Eight acquisition channels are integrated in a single ASD board (*Amplifier Shaper Discriminator*).

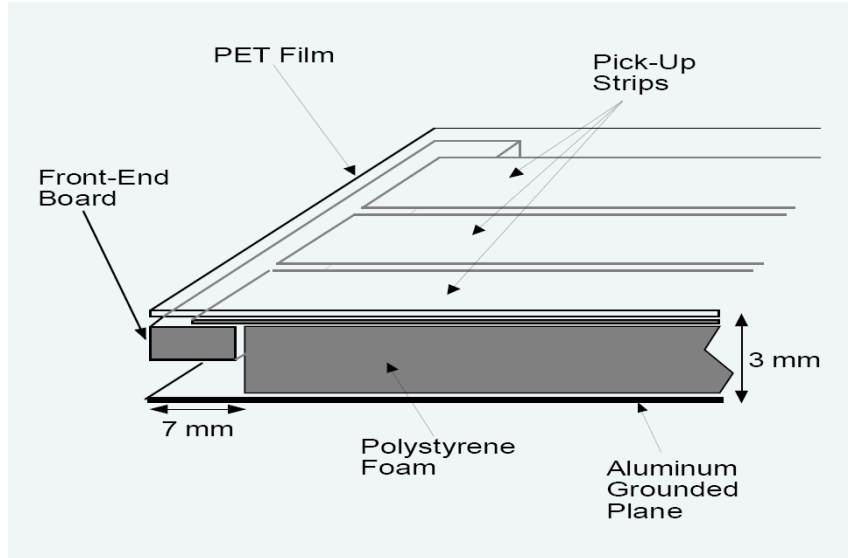


Fig. 2.6 The read-out electronics of an RPC strip

The signals generated by the ASD board are transferred to the trigger electronics. The trigger electronics for the identification of a “*low p_T* ” event is installed on the chambers of the RPC2 station, the trigger logic for “*high p_T* ” events is installed on the chambers of the RPC3 station. The acquisition system and the transmission system of data to the *read-out* system of the spectrometer will be discussed in the next paragraphs.

The synchronization

Before continuing with the description of the spectrometer readout system, a crucial concept as the synchronization must be described in detail. In the beams of the LHC collider [29], protons are grouped in “*bunches*” that interact (*bunch crossing*) every 25 ns in four experimental sites: in one of this sites the ATLAS experiment is installed. From the point of view of the data acquisition system, the ATLAS apparatus is a synchronous system working at the *bunch crossing* frequency of the LHC, i.e. 40 MHz. In order to make the ATLAS apparatus

synchronous with the LHC machine, all the DAQ and trigger system shall be driven by a common clock signal, synchronous to the *bunch crossing* frequency of the collider.

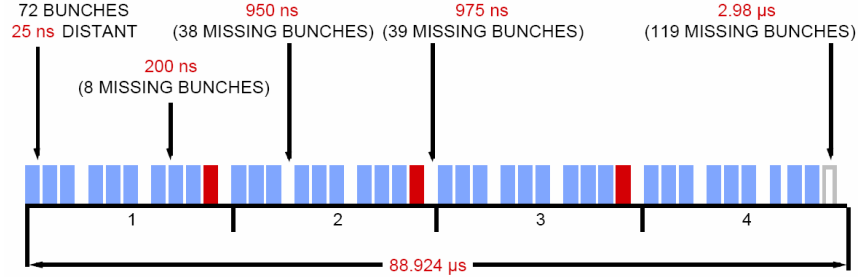


Fig. 2.7 The distribution of the bunches in the LHC orbit

A periodic signal, called orbit, is associated to the revolution of the bunches in the LHC collider. This signal is generated by the control system of the LHC collider, has a period of 88924 ns and is made of 3564 elements; these elements are also called bunches, even if some of them are empty and do not contain any proton. Bunches are grouped in bursts of 72 bunches. Fig. 2.7 shows the structure of a whole orbit. As already said, some “holes” (*missing bunches*) are visible inside the orbit.

In order to make a correct identification of an interesting event, for the physics under study in ATLAS, the elements of the acquisition electronics (*Front End* electronics) have to satisfy two requirements. The front end electronics shall associate to data in each event a unique progressive number (*Event Identifier*, or *EVID*) and a number identifying the bunch crossing that generated the collision (*Bunch Crossing Identifier*, or *BCID*); both these numbers will be managed by counters on the front end boards. Moreover, there is the need to transmit, to all the ATLAS apparatus, two signals that will allow to keep coherence between these two identifiers (i.e. *EVID* and *BCID*) and the events inside the apparatus. These two signals are the *Event Counter Reset* (*ECR*) and the *Bunch Counter Reset* (*BCR*), whose tasks will be discussed later.

In order to achieve such synchronization between the elaboration systems and the machine clock there is the need to build a transmission system, able to distribute the clock signal and all other control signals (such as the level 1 trigger

decision) to all elements of the ATLAS apparatus, keeping under control the *skew* and the *jitter*. In fact, because of the large dimensions of the detector, the reference clock signal must be transmitted to all the entities of the apparatus, over distances up to hundreds meters. This is one of the problems that must be faced to make the entire apparatus working at 40 MHz. Due to the need to reach hundreds of different destinations, the clock phase should be controlled in a subnanosecond range, to enable a correct working environment of the ATLAS apparatus. So there is the essential need to use a system that allows a resynchronization procedure and that guarantees a clock recovery in the event of a loss of synchronization.

Fig. 2.8 shows the control architecture designed for the LHC collider, in particular the *Trigger Control System* and the system *Timing Trigger and Control* [30].

The *Trigger Control System* is in charge of the generation of the timing and trigger signals. The *TCS* receives from the LHC machine the clock and the orbit signals, and receives from the *Central Trigger Processor* (*CTP*) of the ATLAS experiment the signal of validation of the Level 1 trigger (*LVL1Accept*).

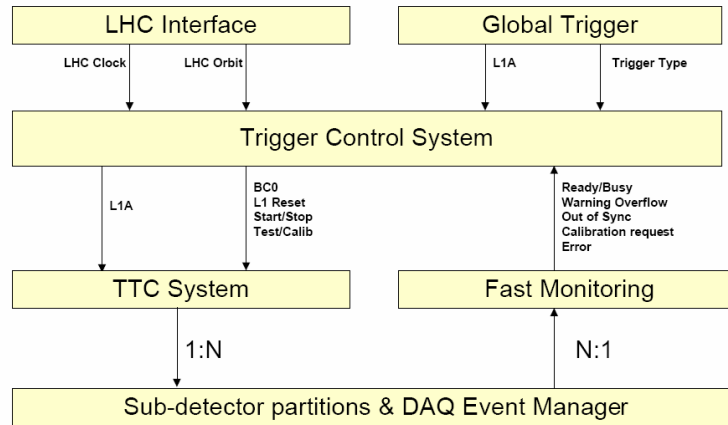


Fig. 2.8 The architecture that manages the control signals

The *TCS* also manages the *Reset* (*ECR* e *BCR*) signals, used to recovery the synchronization in case of a loss of information due to a bad transmission or to a problem in the storage of data. Such reset signals are transmitted periodically or when there is a request by one of the subsystems of the ATLAS apparatus, if a malfunction has occurred. A further task of the *TCS* is the control and the

management of the calibration, synchronization and test signals for the apparatus subsystems. Moreover, the *TCS* manages also some service signals, such as *busy* (indicating that the system is busy and cannot receive a trigger signal), *out of sync* (indicating that fragments of events associated to the same *bunch crossing* are not labeled with the same *EVID* or *BCID* codes) or error signals.

The *Trigger Timing and Control (TTC)* [31] system is responsible of the distribution of the timing and trigger signals to the entire detector and to the different entities of data elaboration. The *TTC* receives from the *TCS* the 40 MHz clock signal, the orbit signal, the *LVIA*, *BCID*, *EVID*, *BCR*, *ECR* and other service signals. All these signals are coded and optically transmitted, by a tree structure, as shown in fig. 2.9, to the elaboration systems and toward the different destinations in the ATLAS apparatus. Signals are reconstructed at the destination and are adapted to the protocols of every subdetector. For example, for the level 1 trigger system in the barrel region, the *TTC* system is divided in three partitions, one for $\eta > 0$, one for $\eta < 0$ and one for the counting room *USA15*, for a total of 912 destinations.

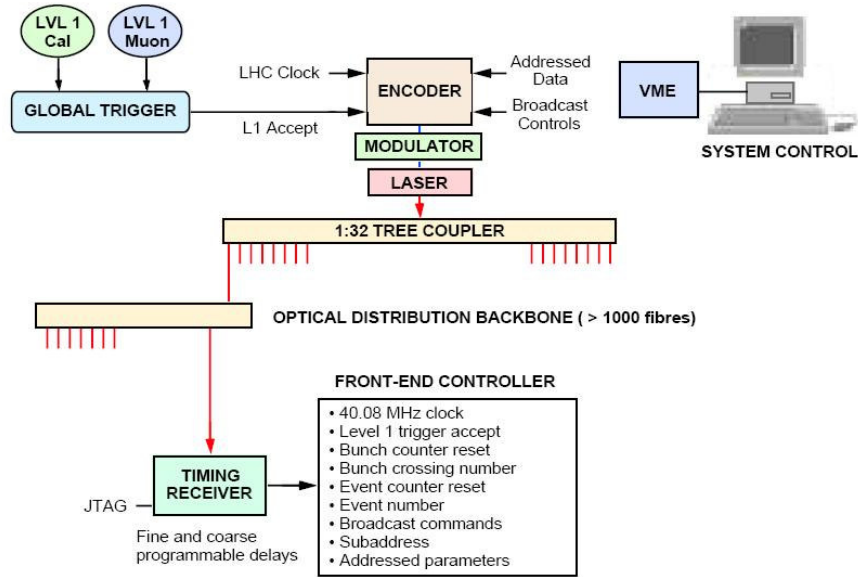


Fig. 2.9 The tree structure of the *TTC* system

The *TTC* incorporates facilities to compensate for particle flight times and detector, electronics and propagation delays. In addition it provides simultaneous

transmission of synchronized broadcast commands and individually-addressed controls and parameters, such as channel masks and calibration data.

Some high-power laser sources will be used to distribute the TTC signals to their destinations through entirely passive all-glass networks composed of a hierarchy of optical tree couplers. At each destination, a special timing receiver ASIC (TTCrx) delivers all the signals required by the electronics controllers.

The tree structure requires considerably less optical fiber than a star configuration of point-to-point links from a central fanout. The optical tree couplers have small size, low mass and wide bandwidth. They require no operating power and are highly reliable. Moreover, as new devices will become available, the architecture adopted allows the TTC system to be easily upgraded.

The distribution to the different units of the apparatus is made by the TTCrq board, shown in fig. 2.10, that reconstructs the signals *Clock*, *LVL1A*, *Event Counter Reset* and *Bunch Counter Reset* and makes them available on four different channels, so they can be used by the different elaboration structures. The reconstruction latency of the TTCrx is about 100 ns, and the clock *jitter* is about 50 ps RMS [32].

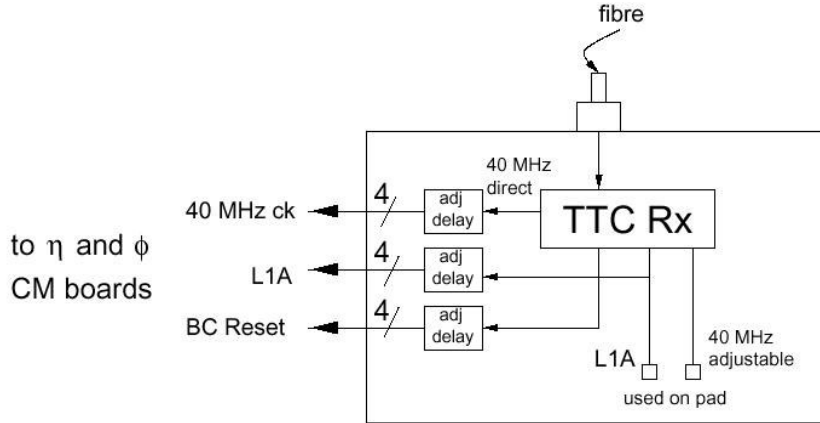


Fig. 2.10 The scheme of the TTCrq board

Starting from the TTCrx signals, the data acquisition boards (*Front End* electronics) can generate the *Event Identifier* and the *Bunch Counter Identifier*. The *EVID* is incremented every time that the *LVIID* signal occurs, is coded by 24 bit and is counted starting from the last *Event Counter Reset*. The *BCID* is

incremented at every *bunch* of the orbit, is coded by 12 bit ($2^{12}=4096$, the power of 2 nearest, in excess, to 3564) and is counted starting from the last *Bunch Counter Reset*. These two identifiers will be indicated also as *FEBCID* and *FELIID*, where *FE* is for *Front End*.

The spectrometer segmentation

The ATLAS muon spectrometer has been segmented, in the barrel region and in the ϕ direction, in 16 different sectors.

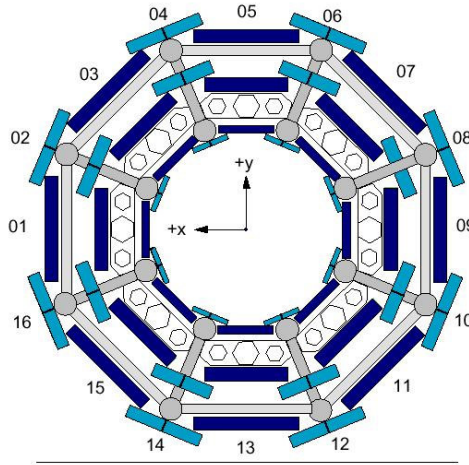


Fig. 2.11 The segmentation in sectors of the spectrometer in the barrel region

Fig. 2.11 shows such segmentation of the spectrometer, made starting from the octagonal structure of the magnetic system. Each sector corresponds to an azimuthal region defined by the coils disposition in the magnet. Two kinds of sectors can be identified: the sectors occupied by the coils (indicated by even numbers and called “*Small Sector*”) and the sectors between two adjacent coils (indicated by odd numbers and called “*Large Sector*”).

Starting from this segmentation, the trigger and data acquisition system is segmented into 64 logic sectors. In fact, dividing each of the 16 sectors (*Large* or *Small*) of the spectrometer by two, 32 sectors for $\eta > 0$ and 32 sectors for $\eta < 0$ are defined. In the boundary regions between two adjacent sectors, there is a partial overlap of the detectors: in such a case, the same muon can be counted twice. To

solve the problem of the overlap of signals in adjacent detector chambers, opportune trigger and acquisition system algorithm have been developed.

The electronics on the RPC detectors

The trigger and data acquisition electronics for the RPC detectors of the spectrometer (*on-detector* electronics) is made of ASD boards, of Coincidence Matrixes CM and of PAD boards. These elements will be indicated as *Front End* electronics. The signals produced by the RPC detectors are received by the ASD (*A*mplifier *S*haper *D*iscriminator) boards: here the signals are amplified, discriminated by using a programmable threshold, and shaped in width. Each ASD board houses eight acquisition channels.

As previously described, the data acquisition and trigger system of the ATLAS apparatus must be synchronized to the clock signal of the LHC collider. These requirement is essential for the coincidence matrixes CM and for the PAD boards, but is not necessary for the elements of preamplification and discrimination of the RPC detector signals. In fact, the process of formation of the event, even if synchronized with the *bunch crossing*, is intrinsically asynchronous, because it depends on the flight time of the muon, on the discharge inside the RPC detector and on the propagation time of the signal along the read-out strip.

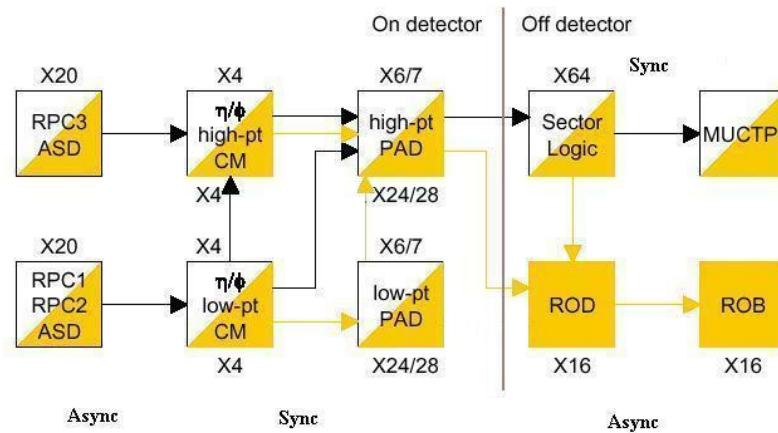


Fig. 2.12 The scheme of the acquisition electronics. The light lines show the data path of the detector, the dark ones show the trigger path

The entire spectrometer has 64 logic sectors, 1664 ROIs, 416 PAD boards for “*low p_T*” trigger and as many for “*high p_T*” trigger. The coincidence matrixes CM use the information produced by the ASD boards to execute the two trigger algorithms, “*low p_T*” and “*high p_T*”. Data produced by the CMs are further elaborated by the PAD boards, that take care also of the transmission to the calculators that form the next trigger and DAQ levels: a simplified path of the *readout* data and of trigger information is depicted in fig. 2.12.

The Coincidence Matrix

The trigger electronics installed on the RPC chamber is schematically shown in fig. 2.13.

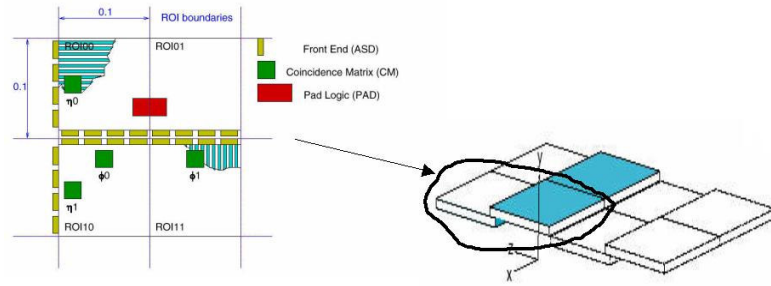


Fig. 2.13 The trigger electronics on the detector

The ASD boards are located at the end of the RPC strip and are connected to the coincidence matrixes, distant few meters and mounted on the PAD logic boards. Fig. 2.14 shows one PAD board. Every PAD board hosts four coincidence matrixes, relative to a region of granularity $\Delta\eta \times \Delta\phi \approx 0.2 \times 0.2$, and also hosts the *TTCrx* and the module indicated as *Link Tx*, responsible of the transmission on optical fiber of the trigger and *readout* data toward the next acquisition levels.

The “*low p_T*” coincidence matrixes in η direction, indicated in fig. 2.13 as η_0 and η_1 , receive, from each of the detector planes shown in fig. 2.15, in both RPC1 and RPC2 station, signals from four ASD boards, each reading eight η strips.

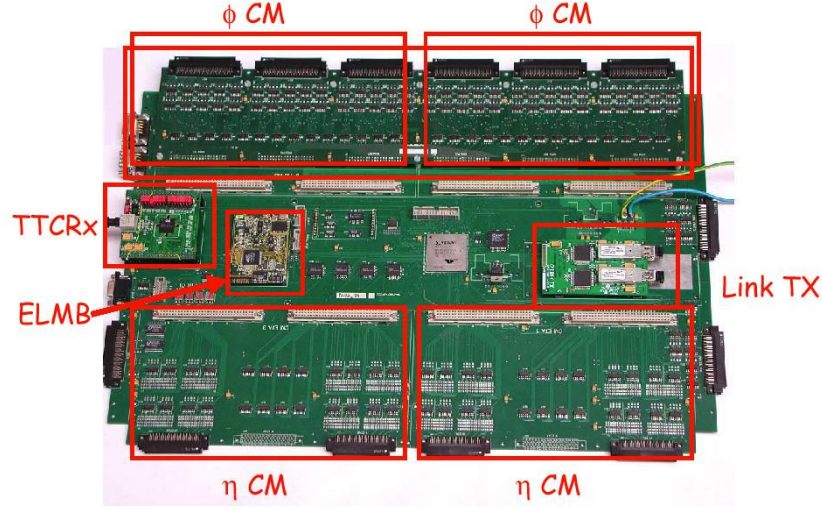


Fig. 2.14 The PAD board and the hosted elements

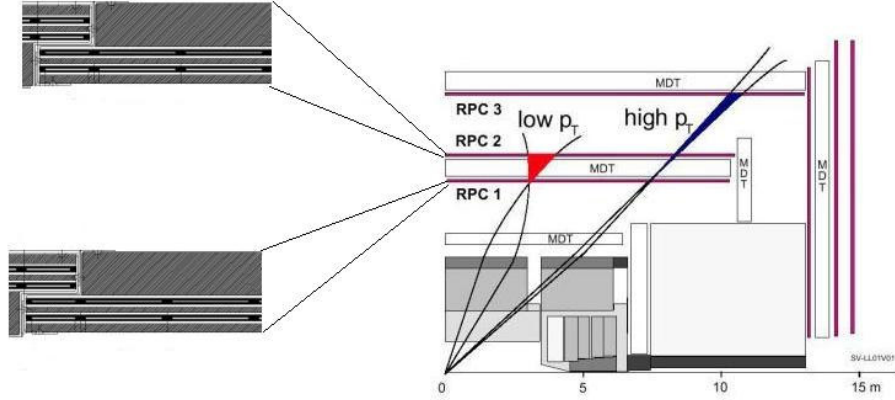


Fig. 2.15 The disposition of the RPC detectors of the trigger system

So, every CM matrix receives a total of $8(\text{strips}) \times 4(\text{ASD}) = 32$ channels from every RPC plane, i.e. $32 \times 4(\text{planes}) = 144$ acquisition channels, in η direction and in “ $low p_T$ ” events. The matrixes cover an area of granularity $\Delta\eta \times \Delta\phi \approx 0.1 \times 0.2$. As discussed in the previous paragraphs, in these regions a “ $low p_T$ ” event is detected resolving the coincidences defined between the signals of the four detector planes. The coincidence is verified in a programmable matrix, depending on the transverse momentum threshold defined. Every CM board allows programming, at the same time, up to three different trigger conditions. The

coincidence can be also programmed with a majority 2/4, 3/4 or 4/4 between the four detector planes. As example, with a 3/4 majority, the trigger condition is valid only if at least 3 of the 4 signals received from the two trigger stations are detected, within the trigger window and in a timing coincidence of 20 ns. This majority algorithm partly allows reducing the background of the apparatus. A further background rejection factor is achieved by the confirmation of the event, with the same procedure previously described, with the signals of the ϕ strips, longitudinally oriented.

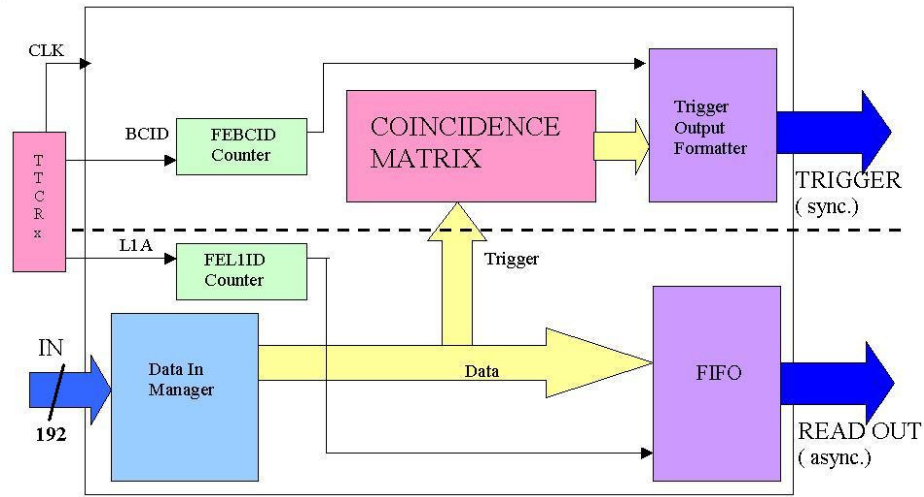


Fig. 2.16 Block scheme of the coincidence matrix CM

Fig 2.16 shows the block scheme of the coincidence matrix CM [33]. The matrix has been designed in VLSI by the research group of the section of the *Istituto Nazionale Fisica Nucleare* and of the Università “La Sapienza” of Roma. The board can receive up to 192 signals (*IN*), generated by the ASD boards, and the signals *BCID*, *LIA* and the 40 MHz clock, recovered by the TTCRx and managed by two counters inside the CM. The trigger and the readout data follow two different paths, after being processed by the input logic to the coincidence matrix. Data of the detector (*Read Out*) are stored in *FIFO* (*First In First Out*) memories and wait to be transmitted (or not) to the next acquisition and elaboration levels, depending on the decision of the level 1 trigger; trigger data are elaborated inside the board in synchronous way and transmitted to the trigger

processors. What is worth underlining is the different principle of the processing of the two data sets: trigger data are elaborated by a rigorous synchronous protocol, clocked by the *Bunch Crossing* of the collider (40 MHz); read out data are processed by an asynchronous protocol, clocked by the frequency of the *LIA* signal (~ 100 KHz).

Fig. 2.17 shows a detailed block scheme of the chip of the coincidence matrix: inputs are labeled by I0 and I1 (32 bits for each doublet of the RPC1 station) and by J0 and J1 (for the doublet of the RPC2 station, or for the RPC3 station, for the “*high p_T*” events: in this case, up to 64 bit are foreseen to be used). The internal frequency of 320 MHz, generated by a DLL, allows to divide each clock period in eight parts and allows a more accurate calibration of the system. At the input, the presence of three different blocks can be noted. A *dead-time* circuit introduces a programmable dead time of few nanoseconds. It makes it possible to avoid that more than an impulse, within the programmed timing window, is detected on the same input channel. There is the possibility to mask, i.e. to exclude the channels that are not useful, for example because are noisy channels. Moreover, there are the “*pipeline*” memories, with a programmable depth, useful to store data and to correctly perform the timing alignment of data, that arrive to the chip in different instants because are generated in different positions.

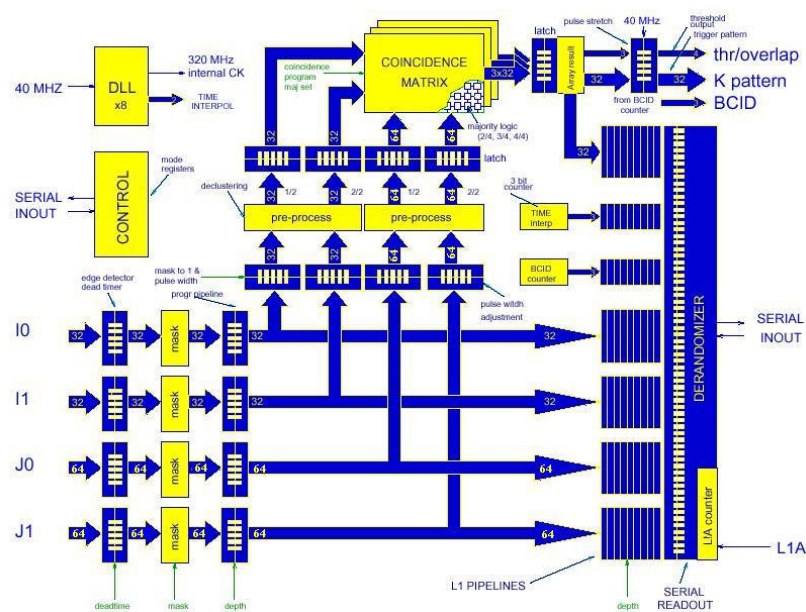


Fig. 2.17 The detailed block scheme of the chip of the coincidence matrix

After the input memories, data are separated and transmitted on two different paths, one for the *Read Out* (the lower part, in fig. 2.17) and one for the trigger. In particular, the output of the trigger block is made by the 4 bit word *thr/overlap*, by the 12 bit *BCID* and by the 32 bit *K-pattern*, intended to go to the J0 and J1 inputs of the “*high p_T*” CM matrix, but also intended to be transmitted, together with the *read-out* data in case of a level 1 validation, toward the next data acquisition levels. The *read-out* data and the *K-pattern* are stored in seven banks of memories, with programmable depth, waiting for the decisions of the level 1 trigger. If the *LIA* signal occurs, data are promoted to the *FIFO* memories (in fig. 2.17 labeled as “*derandomizer*”) that prepare them for the serial output, otherwise data are cancelled. Memories must keep data for a time equal to the latency of the level 1 trigger, i.e. for all the time the trigger elaborates data and decides whether accept or not the event. Such latency must be less than 2.5 μ s and, because the memories work with a clock of 320 MHz (a period of ~ 3.1 ns), the depth of the memories must be greater than 800 locations.

Trigger data follow a more complex path. Data are shaped in width (*digital shaping & pulse width*) in order to perform the signals coincidence with greater reliability. In fact, because the internal clock is 320 MHz, these signals must have a width less than ~ 3.1 ns, in order to avoid false coincidences. Then the *pre-processing* is performed, that executes the 1/2 and 2/2 majority operations, followed by a register that prepares data for the true coincidence matrix logic. The coincidence matrix logic performs the coincidence and the 2/4, 3/4 and 4/4 majority operations. As previously said, the occurrence of a trigger condition is coded in output in the 4 bit word *thr/overlap*, that is transmitted to the PAD board and that contains the highest transverse momentum threshold validated in the event and the possible occurrence of the coincidence in the overlap regions of the chambers. The matrixes produce in output also the 12 bit of the *BCID* counter and, for the “*high p_T*” algorithm, the *K-pattern* from the RPC1 station that generated the trigger signal.

As for the CM η matrixes, the φ_0 and φ_1 coincidence matrixes search, in every φ region, a coincidence between the signals of the 4 detector planes in the “*low p_T*” stations and cover granularity regions of $\Delta\eta \times \Delta\varphi \approx 0.2 \times 0.1$. In φ projection no threshold is imposed on the momentum of the detected muon.. The only

required condition is the existence of a coincidence of the signals with a 2/4, 3/4, 4/4 majority, within a 20 ns time interval. To analyze the “*high p_T*” events, the information are transferred to the corresponding coincidence matrixes $\eta_0, \eta_1, \phi_0, \phi_1$ “*high p_T*”, that are installed on the RPC3 chambers.

The coincidence matrix η_0 “*high p_T*”, for example, as shown in fig. 2.18, receives the signals from the coincidence matrix η_0 “*low p_T*”, and from the two detector planes of the RPC3 station. The coincidence algorithm is the same as before: as for a “*low p_T*” event, the matrix searches a time coincidence between the signals, within a time interval of 20 ns, with a 2/4, 3/4, 4/4 majority and in spatial programmable windows depending from the threshold of the imposed transverse momentum.

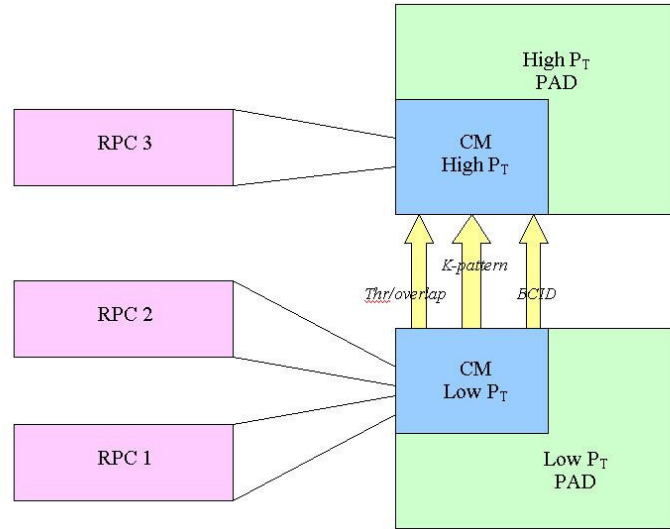
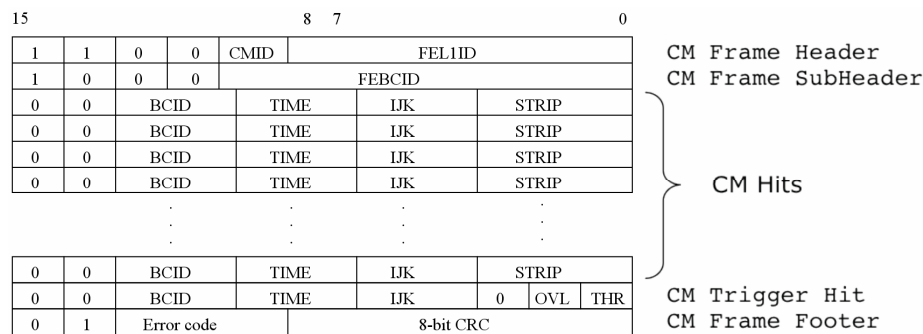


Fig. 2.18 The path for the data of the *High p_T* algorithm

The CM matrix is programmable to resolve 1/2 and 2/2 majority coincidences between the two signals of the RPC3 station only and independently from the occurrence of a “*low p_T*” event. This allows confirming, in the next trigger system logic levels, a “*low p_T*” event and to reduce the noise of the apparatus.

The format of the CM Fragment [34], i.e. the output data of every CM board, is shown in fig. 2.19 and, as can be seen, is arranged in blocks (“*frame*”) made of 16 bit words, whose most significant bits give information on the kind of the transmitted word. The four most significant bits identify the “*header*” and the



<i>OVL</i>	<i>Description</i>
0	No overlap
1	Overlap in low p_T CM channels (low $ \eta $)
2	Overlap in high p_T CM channels (high $ \eta $)
3	$OVL=1$ and $OVL=2$

The PAD Logic boards

The information of two adjacent “*low p_T* ” CM matrixes in η direction, and the corresponding information of the two CM matrixes in ϕ direction, are elaborated by the “*low p_T* ” PAD Logic board. For the “*high p_T* ” trigger algorithm, as shown in fig. 2.18, data arriving to the “*high p_T* ” PAD logic board come from both the CM “*low*” and from the CM “*high*”.

The PAD Logic “*low p_T* ” board and the four CM “*low p_T* ” are installed on the RPC2 station, whereas the PAD Logic “*high p_T* ” board and the four CM “*high p_T* ” are installed on the RPC3 station.

One PAD Logic board covers a granularity region of $\Delta\eta \times \Delta\phi \approx 0.2 \times 0.2$, whereas the dimension of a *Region Of Interest* is $\Delta\eta \times \Delta\phi \approx 0.1 \times 0.1$: so each PAD Logic contains information on 4 ROI. A *Small* sector of the spectrometer is managed by 7 PAD, a *Large* one is managed by 6 PAD.

Fig. 2.21 shows the block scheme of a PAD Logic board [35].

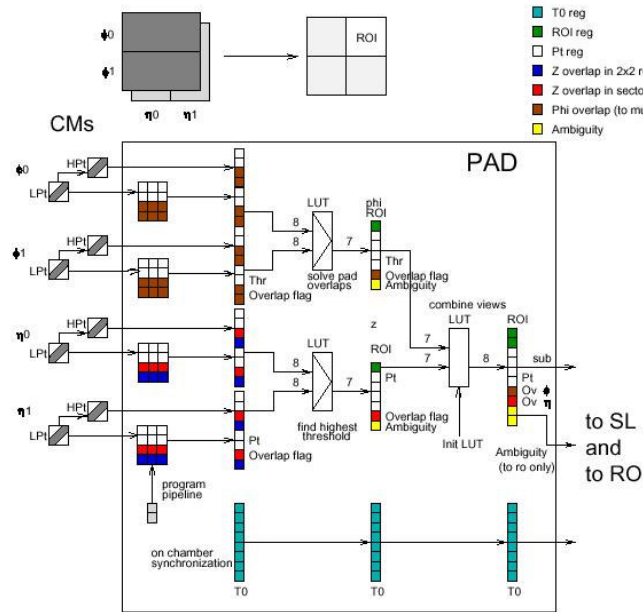


Fig. 2.21 Block scheme of a PAD Logic board

The main task of the PAD board is to perform a further elaboration of *readout* data and trigger data, produced by the CM matrixes. These data are combined and two different frames with different information, for *Read Out* and for trigger, are

produced and should be sent on optical fiber to the counting room *USA15*. While the transmission protocol of the *Read Out* data is asynchronous and clocked by the occurrence of the *LIA* signal (~ 100 KHz), trigger data are transferred with a synchronous protocol, at the *bunch-crossing* frequency (40 MHz) and with a latency that must be little (in order to not saturate the next *pipeline* memories) and fixed (to allow an exact time calibration of the acquisition system). The other tasks of a PAD board are: the identification of the *Region Of Interest* for an event validated by the trigger system, that combines information both in η and in ϕ direction; the transmission of the trigger information (such as the various momentum thresholds imposed), of the *BCID* and of the *flags* of overlap in η and in ϕ . Furthermore, using the TTCrx chip, each PAD board receives the signals *LVLIA*, *LVLIRST*, *BCIDRST* from the *TTC* and distributes them to the four coincidence matrixes and to the PAD board logic.

Fig. 2.22 shows the diagram of the data flow inside the PAD board. The logic is managed by an *FPGA* (*Field Programmable Gate Array*) designed for this purpose: data produced by the CM enter in the *FPGA* as serial inputs, are converted to parallel data and are combined to produce 16 bit words. In order to be transferred to the next elaboration levels, such words are reconverted in a serial format by the devices that manage the optical transmission.

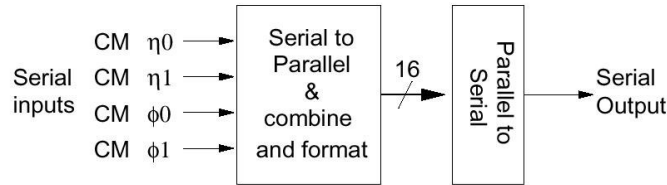


Fig. 2.21 The block scheme of the data flux inside the PAD board

The logic of the PAD Logic board is structured following a “*pipelined*” model: operations are executed in three consecutive steps. With the first step, data coming from the CM matrixes are time aligned; with the second step, the possible overlaps in η and in ϕ are resolved and the highest momentum threshold is calculated; with the third step, data are combined together and the output is produced. The event is confirmed if a trigger condition is found both in η and in ϕ direction.

If an “*high p_T* ” event is not found, the status *flag* OPL is asserted, that indicates the existence of a 1/2 or 2/2 majority coincidence between the signals of the RPC3 station only inside the ROIs checked by the PAD.

As will be described in the next paragraph, data produced by a PAD board will be sent over optical fiber to a receiver board, the RX/SL board. In particular, PAD data are sent to different sections of the RX/SL board, depending on they are trigger or *read-out* data. It’s worth underlining that trigger data are sent to the *Sector Logic* (SL) section of the RX/SL board with a synchronous protocol at 40 MHz; *read-out* data are transferred to the RX section of the RX/SL with an asynchronous protocol, only in occurrence of the L1A signal, which has an expected frequency of ~ 100 KHz.

Data produced by PAD boards have different formats, depending whether they are trigger information (12 bit) or *read out* data: in this case a new frame with 16 bit words is produced, that contains up to eight *CM frames*. A *CM frame* is always transmitted, even if there is no interesting event, to transmit *FELIID* e *FEBCID* (the identifiers that are generated by the counters on the *Front-End* boards).

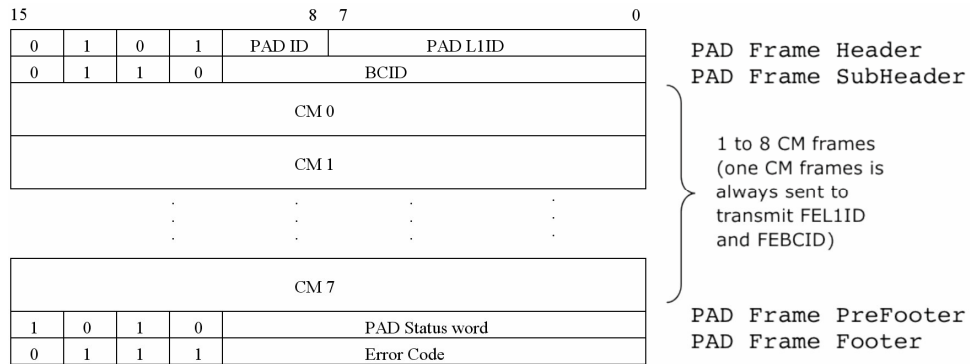


Fig. 2.23 The *Read Out* data format in output from a PAD

A *PAD frame* contains an *header* and a *footer*, that are identified by the values of the three most significant bit, respectively “1 0 1” e “0 0 1”. The *header* field contains a PAD identifier (*PADID*, 4 bit) and a control code (*status*, 8 bit). The *footer* field contains an error code (13 bit). The output data format of a PAD board, for *read out* and trigger data, are shown in fig. 2.23 and 2.24.

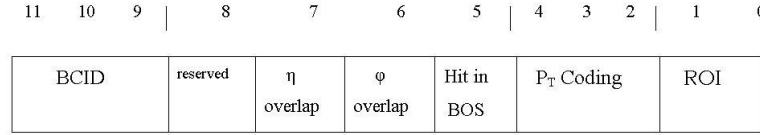


Fig. 2.24 The trigger data format in output from a PAD

The path of the information

Data from a PAD logic board are transmitted over optical fiber to the RX-Sector Logic board and, from these, to the ROD. The ROD (*Read Out Driver*) is the entity that is responsible of the management and the elaboration of the *read out* data coming from the PAD boards of a whole sector (*Large* or *Small*) and of the transmission of data to the next elaboration systems.

A synchronous transmission system with a small and fixed latency is needed for trigger data. The transmission latency must be small and fixed in order to allow to calibrate all the acquisition systems. This latency must also be small, in order to keep small the depth of the “*pipeline*” memories that have to store data in the next acquisition and elaboration systems.

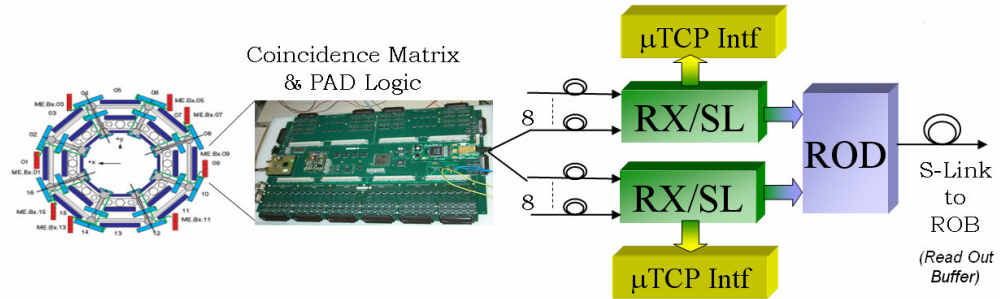


Fig 2.25 the data path from the detector to the counting room

Fig. 2.25 shows the path of trigger and *read out* information. *Read out* data are grouped in 16-bit words (and a strobe bit), trigger data are transferred in 12 bit words. As already said, the trigger data path must be rigorously synchronous, at the 40 MHz of the LHC, but the *read out* data path is asynchronous, because interesting data (and so the LV1A signal that validates data) are not produced in the detector at every *bunch crossing*.

Fig. 2.26 shows the structure of the *ROD crate*, located in the counting room *USA15* at about 80 meters from the beam interaction point, that hosts two *RX-Sector Logic* boards and one *ROD board* (master). Keeping in mind the dimensions of the optical receivers on the *RX-SL* boards and since in every *crate* there are 21 *slot*, is reasonable to create in a single *crate* not more than two replicas of the structure presented in the fig. 2.26. Every structure is controlled by a VME central unit, *CPU*, that is responsible of the management of the whole system. As can be seen, every *ROD* is connected, through custom *bus* (*RODbus*), both to the trigger section (*SL*) and to the read-out section (*RX*) of every *RX/SL* board. The trigger section (*SL*) of the *RX-SL* board receives trigger data over optical fiber from the *PAD* boards, transmits its data to the interface to the trigger processor *MUTCPI* (*Muon Trigger Central Processing Interface*) and communicates with the *ROD* to transfer diagnostic information. The *read-out* section (*RX*) receives *read-out* data over optical fiber from the *PAD* boards, up to a maximum of eight *PADs*, and after an elaboration transmits data to the *ROD*. The *RX-SL* boards transmit data to the *ROD* in words made of up to 48 bit: as an example the 16 bit words *read-out* data are arranged to form 32 bit words, made of 24 data bit and 8 control bit. In the next paragraphs a description of the *RX/SL* board will be given.

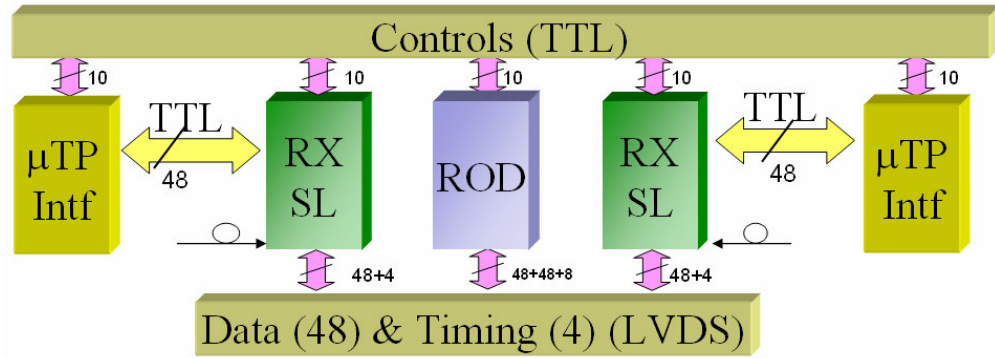


Fig. 2.26 The structure of a *ROD crate*

Every *ROD* can manage the information coming from one of the 32 sectors into which the spectrometer is divided (considering the division in $\eta > 0$ and $\eta < 0$). Each of these sectors corresponds to two logic sectors into which the trigger

system is divided, that are managed each by 7 PADs (if it is a *Small* sector) or by 6 PADs (if it is a *Large* sector). Each RX-SL is dedicated to the management of data of one of the 64 logic sectors of the trigger of the spectrometer, receiving trigger information from 6 or from 7 PADs. The number of the PAD boards connected to a ROD, through the RX-SL boards will be given by $6 \times 2 = 12$ (for a *Large* sector) or by $7 \times 2 = 14$ (for a *Small* sector). In total, 32 structures, similar to the one shown in the fig. 2.26, will be needed to manage the flux of trigger and *read out* information of the whole spectrometer.

The ROD, whose design is the main task of my Ph.D. thesis, will be described in detail in the next chapters. Fig. 2.27 shows the *event frame*, i.e. the “frame” pertaining to the whole event inside the detector, that is built by the *Event Builder* starting from the information of the ROD and elaborated by the following data acquisition systems. The *event frame* is made of 32 bit words.

Event Frame

31	16 15	0
Marker		
Source ID		
Reserved		
LVL1 ID		
BC ID		
LVL1 Trigger Type		
Detector Event Type		
Status words		
RPC Data word		RPC Data word
RPC Data word		RPC Data word
RPC Data word		RPC Data word
# Status Words		
# Data Words		
Trailer		

Fig. 2.27 The *event frame* format

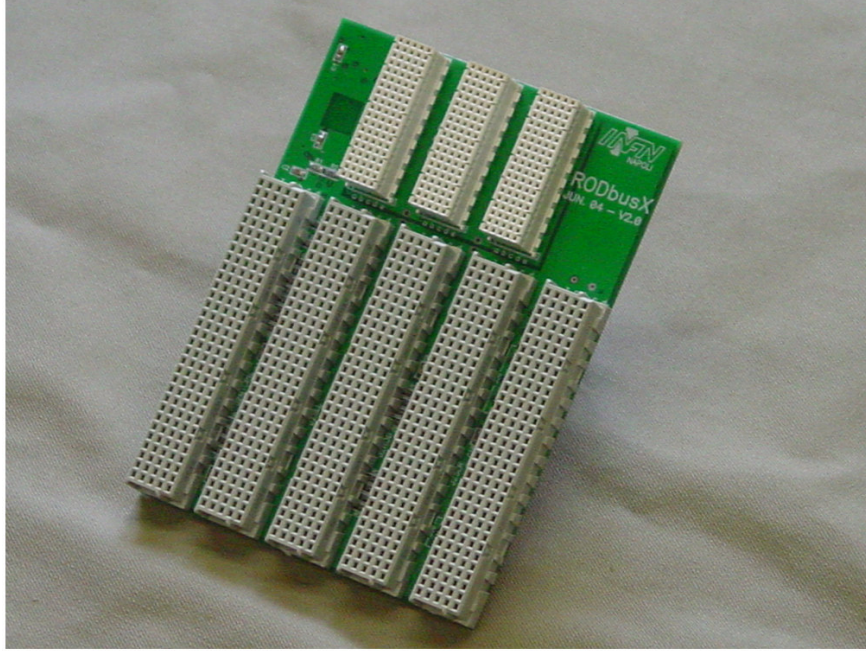


Fig. 2.28 The *RODbus* prototype

For the design of the *backplane* that contains the custom bus *RODbus*, some requirements must be kept in account. To have signals with a low *skew* and a low *jitter* a differential approach must be followed, and, in order to limit the power dissipation and to reach high performances in frequency, the LVDS standard has been chosen. Fig. 2.28 shows a *RODbus* prototype, designed and tested at the research laboratories of the “*Federico II*” University of Napoli and of the *INFN* Section of Napoli. A large description of the *RODbus* will be given in the next chapter of this thesis.

The optical link

Because of the requirements for the data transmission over optical fiber, for both trigger and *read-out* data, a unique transmission system, satisfying all the specifications, has been chosen, the optical link G^2 LINK.

The table 2.1 shows the bandwidth between the *DAQ* and the trigger structure that have been described in this chapter.

In the specific case of the ATLAS spectrometer, there is the need of transmitting both trigger and *read-out* data from the PAD to the RX-SL board. *Read-out* data

will be made of 16 bit words, trigger data will be synchronous 12 bit words at 40 MHz.

Connection	Link Bandwidth	Mean Bandwidth
from CM to PAD	10 Mbit/s	3.0 Mbit/s
from PAD to RX/SL board	640 Mbit/s	480 Mbit/s

Table. 2.1 Bandwidth of the connections between the different systems that constitute the acquisition structure

The optical link G^2 LINK is based on a classical serial approach and uses commercial devices.

The link uses the *chip-set HDMP-1032/1034* [36] that takes care of the serialization and the deserialization of the data. The opto-electronic section is based on the *HFBR-5912E* [37] devices. In the fig. 2.29 the TX and the RX boards can be seen, that host both the electronic and the optical components.

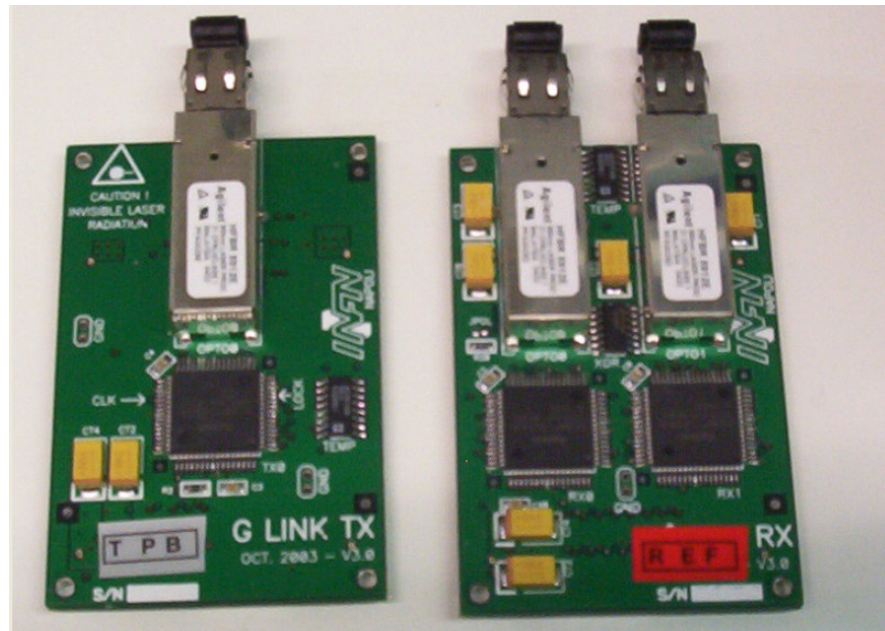


Fig 2.29 The data transmitter (TX) and the receiver (RX) boards

The opto-electronic transducers have a bandwidth of 1.25 Gbit/s, using VCSEL laser (*V*ertical *C*avity *S*urface *E*mitting *L*aser) with a wavelength of 850 nm. The working voltage is 3.3V and the logic standard for input and output is LVPECL (*L*ow *V*oltage *P*ositive *E*mitter *C*oupled *L*ogic).

The *chip-set* HDMP 1032/1034 is made of a transmitter and a receiver: their function is to serialize (TX) and deserialize (RX) the 16 bit (+ 1 *strobe* bit) that are managed by the *chip-set*. The transmitter's output is a differential serial signal, in PECL standard; the receiver input is the PECL couple of data and the output are the 16 bit data and the *strobe* bit in TTL standard.

The clock frequency of the devices can vary between 13 MHz and 70 MHz. The *chipset* does not transmit only the 16 bit data, that constitute the *data field* (*W-Field* or *Word Field*), but also 4 control bits, that constitute the so-called *C-Field* (*Control Field*): hence, the effective frequency over the optical fiber results to be 20 times greater than the clock one. In the specific case of the ATLAS apparatus, the working frequency of the link is the *bunch crossing* frequency (40 MHz), with a serial transmission frequency of 800 Mbit/s.

The serializers allow building either a synchronous architecture, in which the transmitter's and the receiver's clock have a well defined phase relationship, or an asynchronous architecture, with no relationship between the two phases.

Fig. 2.30 and fig. 2.31 show the two different configurations.

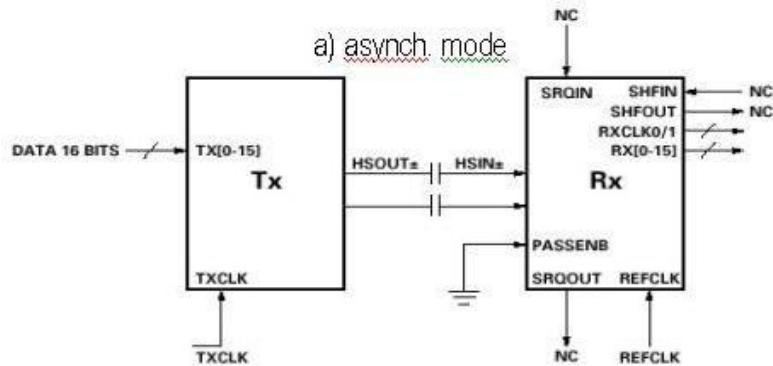


Fig. 2.30 The asynchronous configuration

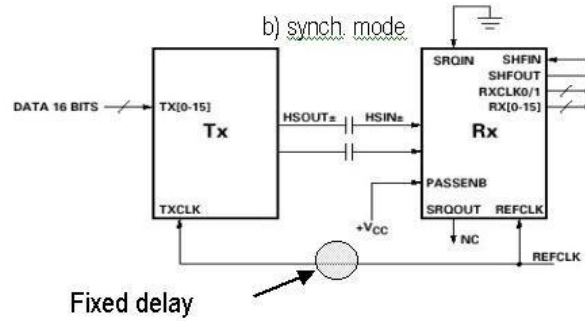
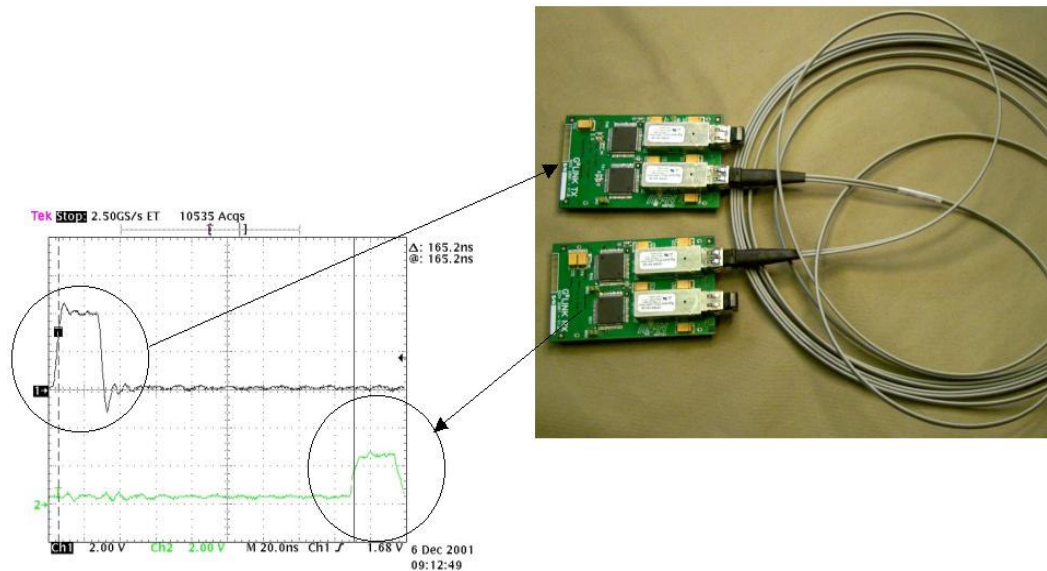


Fig. 2.31 The synchronous configuration

If there is no phase relationship between the transmitter's clock and the receiver's one, there is an asynchronous situation and the only way to read data correctly from the receiver is to use FIFO (*First In First Out*) memories. But if the two clock have only a phase difference, related to the different propagation delay in the transmission (as an example, the distance between the two devices) data can be read from the receiver using either the receiver's clock or the recovered clock. Data recovered by the receiver are driven at the output of the receiver on the rising edge of the recovered clock, that is synchronous with the transmitter clock.

Fig. 2.32 The latency of the G²LINK, in the synchronous configuration

In the synchronous working configuration, that will be used in ATLAS, the latency of the link, over 10 meters of optical fiber, results to be 165 ns, as shown in fig. 2.32.

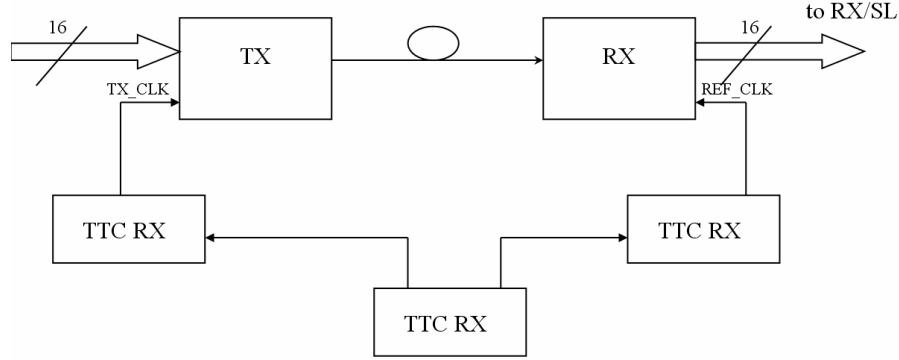


Fig 2.33 The configuration of G²LINK for the synchronous transmission in ATLAS

Some of the requirements for the ATLAS apparatus are the possibility to transmit 12 bit (for *trigger*) or 16 bit (for the *read-out*) data and the use of the same clock (distributed by the TTC). The configuration that satisfies these requirements for transmitting both *trigger* and *read-out* data is the one shown in fig.2.33. Each transmitter board is hosted by a PAD board and hosts an optical transmitter: it manages the transmission toward an RX-SL board.

The optical link G²LINK has been fully developed and tested in the research laboratories of the “*Federico II*” University in Napoli. The test work on the G²LINK optical link has produced the paper “*Do’s and Don’ts with the Agilent G²LINK chip-set*” that I presented at the “*IEEE Real Time 2005 conference*” [38]. In [39] some of the test results on G²LINK are presented.

The RX- Sector Logic board

Information from the *On-Detector* electronics of the spectrometer are transmitted, over the optical link G²LINK, to the counting room *USA15*, where are received by the RX-SL boards. Each RX-SL board hosts up to 8 G²LINK receivers and can be divided into two sections.

The trigger section (*Sector Logic*) elaborates the trigger information of one of the 64 sectors into which the spectrometer is divided, in the *barrel* region. From a “*Small Barrel*” sector, data arrive to each SL from 7 PAD Logic, as shown in fig.2.34; from a “*Large Barrel*” sector data arrive from 6 PAD Logic.

The *read-out* section is made by the RX section of the RX-SL board. It elaborates the *frames* transmitted by the PAD boards (selecting the frames depending on the same *bunch-crossing* parameters), performs a consistency check of the different *frames*, produces an output frame and transmits it to the ROD. In the next two paragraphs I will describe the two sections of the RX-SL board.

The Sector Logic board

Each Sector Logic [40] (the *trigger* section of an RX-SL board) is located in the *USA15* control room and covers a granularity region $\Delta\eta \times \Delta\phi \approx 0.2 \times 1.0$, receiving information from the Pad Logic boards and from the calorimetric trigger. The output of each Sector Logic is sent to the *Muon Central Trigger Processor Interface (MCTPI)*.



Fig. 2.34 The trigger information path from a “*Small*”sector to a SL

The SL implements five different logic functions. These functions are implemented in a “*pipelined*” architecture, with 5 steps at 40 MHz and with a step for every logic function. Fig. 2.35 shows the block scheme of the internal architecture of an SL, that has to do the following:

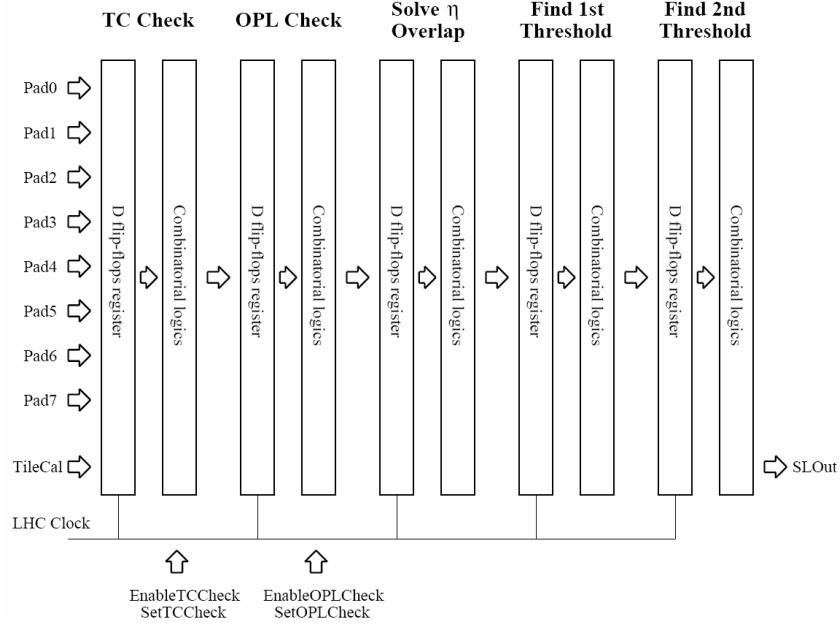


Fig. 2.35 The block scheme of the SL logic

- To confirm trigger information from PAD boards with data coming from the calorimetric trigger. A “*low p_T*” or a “*high p_T*” event is confirmed requiring a signal in the hadronic scintillating calorimeters of the apparatus, inside the interesting *bunch-crossing*. Some studies have shown a great efficiency of the system in separating muons from electrons and photons.
- To filter “*low p_T*” events, i.e. identify “*low p_T*” events starting from “*low p_T*” PAD data and from information coming from the third trigger station, that flags the absence of a “*high p_T*” event. So a “*low p_T*” event is confirmed if the status flag OPL is identified in at least one of the corresponding PAD board.
- To resolve the overlap in η direction and use a flag to identify the overlap in ϕ direction between the different sectors of the spectrometer.
- To select the muon track with the highest p_T , referring it to a unique *bunch-crossing*. Moreover, to define a momentum window around the highest detected p_T , and flag the presence of more than a muon, with a momentum inside that window, in the region identified by a PAD.
- To select the muon track with the second highest p_T , referring it to a unique *bunch-crossing*. To define a momentum window around the second highest

detected p_T , and flag the presence of more than a muon, with a momentum inside that window, in the region identified by a PAD.

0	> 2 candidates in a sector
1..5	1st candidate RoI
6..7	1st candidate reserved
8..9	1st candidate overlap
10..14	2nd candidate RoI
15..16	2nd candidate reserved
17..18	2nd candidate overlap
19..21	1st candidate p_T
22..24	2nd candidate p_T
25	> 1 candidate in RoI 1
26	> 1 candidate in RoI 2
27..29	BCID
30	candidate 1 sign
31	candidate 2 sign

Fig. 2.36 The format of the output data of the Sector Logic

The data format at the output of a Sector Logic, formatted in 32 bit words, is shown in fig. 2.36. Such data are sent to the MUCTPI on 32 couples of transmission lines, in LVDS standard. For control and diagnostic purposes, data produced by the Sector Logic are transmitted also to the ROD, over the RODbus.

The RX section

The *read-out* information produced by the PAD boards are transmitted at every L1A on optical fibre to the *USA15* counting room. Here *read-out* information are received by the RX (the *read-out* section of an RX-SL board) that elaborates data and transmits them to the ROD. The main task of the RX section is to receive frames of a spectrometer's *half-sector*, travelling on multiple optical connections based on G²LINK. Data are selected depending on their *bunch-crossing* parameters, some control bit are introduced and a new *RX Frame* is produced.

Fig. 2.37 shows the scheme of the RX logic, that hosts four optical fibre receiver modules. As each receiver module hosts two optoelectronics transducers, up to 8 optical fibres can arrive on every RX, and so data from a maximum of eight PAD

boards can be collected. Data transmission over optical fibre is done with an asynchronous protocol: so the use of *FIFO* (*F*irst *I*n *F*irst *O*ut) memories is essential for the acquisition of data from every PAD. The reason of the use of an asynchronous protocol can be easily understood if we think that data arrive on the RX/SL from PAD boards of different locations. An accurate phase relationship between the clock of every G²Link transmitter, located on a PAD board, and the clock of the corresponding receiver, located on the RX/SL board, can be calculated. But the phase relationship between the clocks of the different links is not predictable, because it depends on the length of the different optical fibres. Also, there is the need of finding the phase relationship between all the reconstructed clocks and the RX/SL board clock. For these reasons and to allow a correct synchronization of data with the RX/SL board clock, *FIFO* memories will be used.

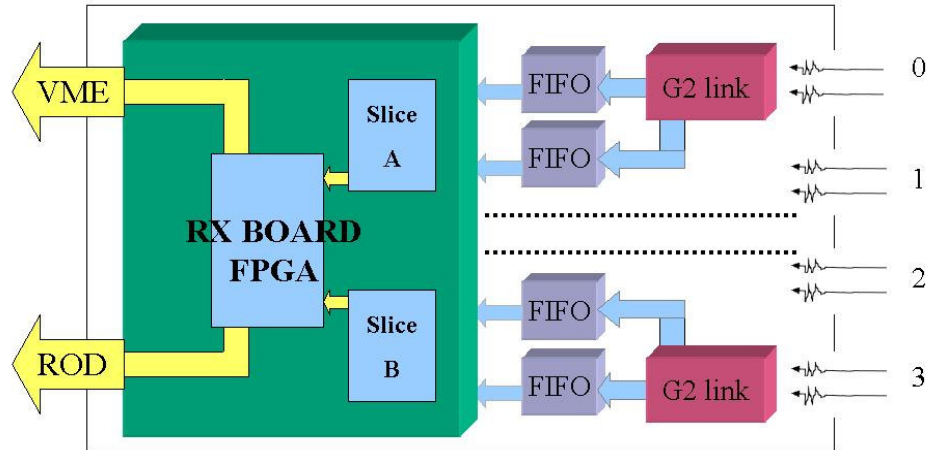


Fig. 2.37 The structure of the RX section

The *event builder logic* inside every RX section allows to execute the “*RX Frame building*” of PAD data, selecting and grouping data depending on their *bunch-crossing* parameters, or to simply transfer PAD data to the next levels of acquisition (i.e. to the ROD, for debug purposes). Data with the same timing parameters, even if arrived to the RX/SL board at different times, are grouped in the same *RX Frame*. The format of the *RX Frame* is shown in the fig. 2.38.

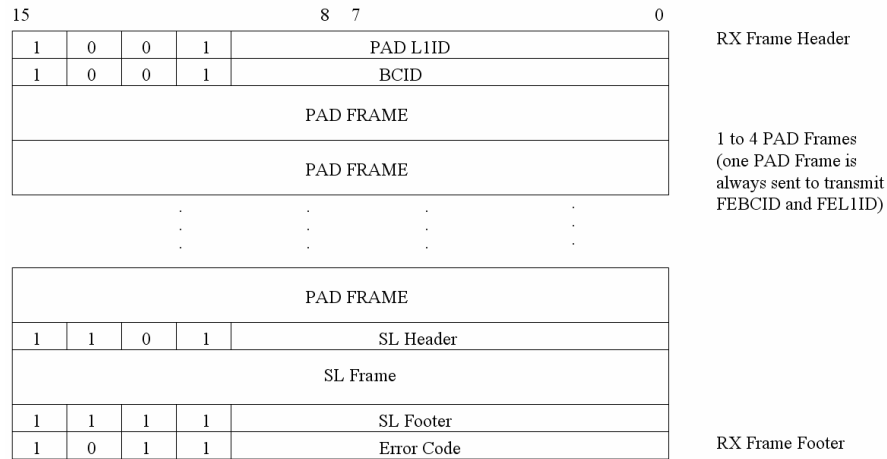


Fig. 2.38 The data format at the output of the RX section of the RX/SL board

The RX section logic also performs a control on the correct data transmission, analysing *header* and *footer* of the PAD frame in input. The RX/SL board has a *VME* interface that can be used both for the board configuration and for the test of the data transmission to the ROD: this task can be done by sending predefined patterns on the *RODbus* that can be received and checked by the ROD.

CHAPTER 3

The Read Out Driver of the Muon Spectrometer of ATLAS

In this chapter, that is part of my original contribution to the ATLAS Muon Spectrometer data acquisition system, the Read Out Driver (ROD) will be described in detail.

The ROD is put in the context of the ATLAS muon spectrometer DAQ. A brief description of the hardware design and simulation language (VHDL) used to design the ROD board logic is also given. The architecture of the board is described, together with the block schemes of the board sections. The different devices on the board are presented too.

The ROD in the acquisition structure

As already described in the previous chapter, every *Read Out Driver* (ROD) is located in one of the *off-detector* data acquisition structure shown in fig. 3.1. This structure is installed in the counting room *USA15*, located at about 80 meters from the beam interaction point in the LHC collider. In such structure the ROD manages *read-out* data and is responsible of one of the 32 sectors into which the spectrometer is divided (considering the $\eta > 0$ and $\eta < 0$ regions). The management of the trigger information is instead demanded to the trigger logic on the *RX/Sector Logic* boards.

Each of the sectors managed by the ROD corresponds to two logic sectors into which the trigger system is divided. Each ROD is in communication, via a custom bus (*RODbus*), with two *RX-SL* boards, that have the task to receive and elaborate trigger and *read-out* data. The *RODbus* is a custom designed bus that allows the boards to exchange data and control signals. In particular, data are transmitted in LVDS standard to have high working frequencies and signals with a low *skew* and a low *jitter*; control signals (as *busy*, *reset*, *offline*), that are not as critical as data, are transmitted in TTL standard. In the next paragraphs a detailed description of the *RODbus* will be given.

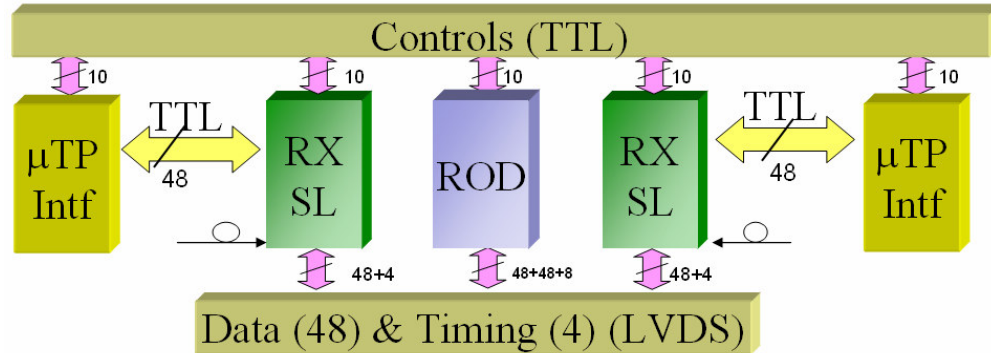


Fig. 3.1 Scheme of the crate that hosts the ROD

The *read-out* data in the RX-SL board

The RX section of the *RX/Sector Logic* board receives *read-out* data from the PAD boards, up to eight optical fibre. Such data are then elaborated and transmitted to the ROD, over the *RODbus* backplane.

The main *read-out* function of the *RX/Sector Logic* board is to format *read-out* data, coming from different PAD boards, in a single frame, as shown in fig. 3.2. Data are selected depending on the same *bunch-crossing* parameters and some control bits are added. Events with the same *bunch-crossing* identifiers (*FELIID* and *FEBCID*), even if arrived to the *RX/SL* board at different times, are grouped in the same *RX Frame*. The format of the *RX Frame* is shown in the fig. 3.3.

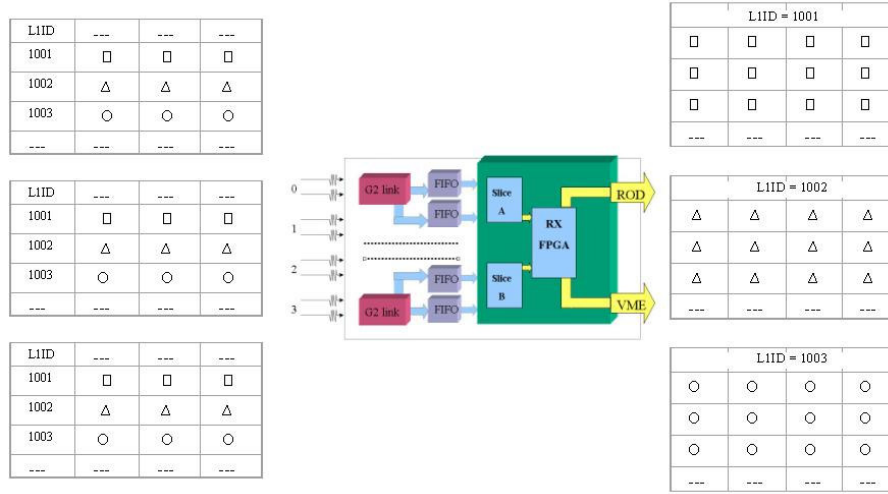


Fig. 3.2 The event building of the *RX/Sector Logic* board

The RX section logic also performs a control on the correct data frame, analysing *header* and *footer* in input from the PAD. The RX/SL board has a *VME* interface that can be used both for the board configuration and for the test of the data transmission to the ROD.

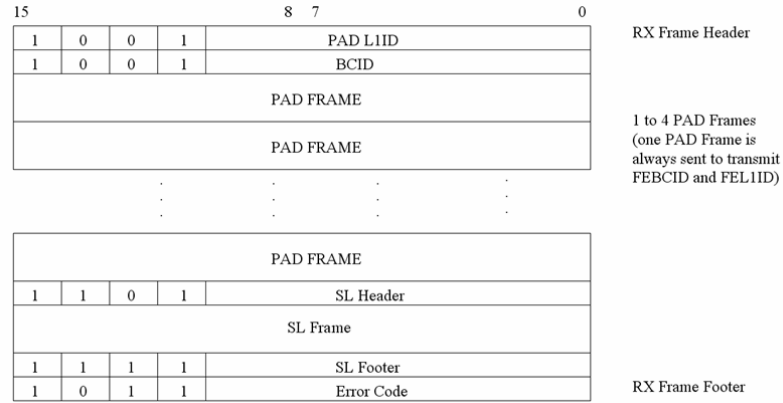


Fig. 3.3 The data format at the output of the *RX/Sector Logic* board

The RODbus backplane

The crate that hosts the ROD board is schematically displayed in fig. 3.1. In every crate, for each ROD board, there are two *RX/SL* boards and two Muon Central Trigger Processor Interfaces, whose task is to interface the trigger section of the nearest *RX/SL* board to the Muon Central Trigger Processor.

One of the fundamental elements of the crate is the custom backplane (*RODbus*) designed to handle all the functionalities needed for this section of the trigger and data acquisition system of the ATLAS spectrometer. Front and rear view of the *RODbus* are shown in fig. 3.4. It is a 10 layer PCB and it fits into the VME64x rear side: lower connectors are in TTL standard and are used for control signals (i.e. busy, reset, diagnostics); the upper connector (that is the central connector of the VME64X) hosts high frequency data, that are transmitted in LVDS standard.

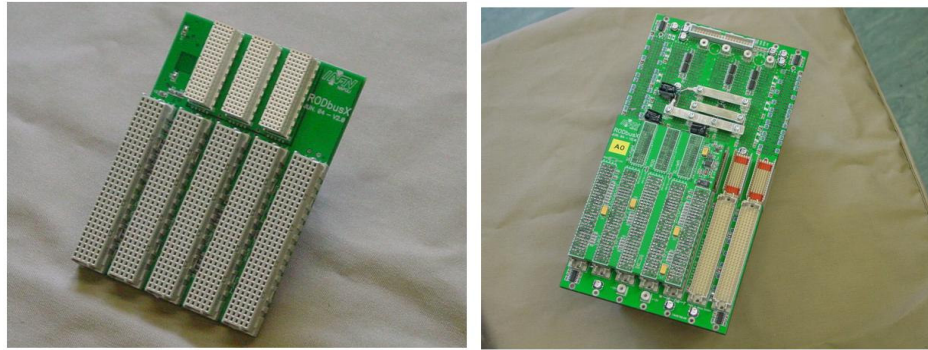


Fig. 3.4 Front (*left*) and back (*right*) view of the *RODbus*

As said before, trigger and read-out information arrive on the *RX/SL* boards, via optical fibre, and are sent to the ROD on the *RODbus*: each *RX/SL* sends 48 bit@40MHz. The signals received by the ROD from the LHC (via TTC) are sent on the *RODbus* too. Such signals are intended to the “daughter” boards, i.e. the *RX/SL*s and the MuTCPIs. For each *RX/SL* there is an 8 bit TTL private bus for timing signals; also there is a 10 bit TTL shared bus for control signals. The *RODbus* also allows the transmission of data from each *RX/SL* to the nearest MuTCPI over a 48bit TTL private bus. For the LVDS domain, the backplane is made of differential pairs, routed as edge-coupled microstrips. The noisy TTL lines are routed on separate planes and connectors and are terminated as VME

lines. In fig. 3.5 two more images of the *ROD*bus backplane are shown, and the two different domains, LVDS and TTL, are visible. It's worth underline that the ground plane between LVDS and TTL domains is split. Is also worth notice that the *ROD*bus is equipped with a temperature probe and a *Analog-to-Digital Converter* (ADC), in order to monitor the power supply.

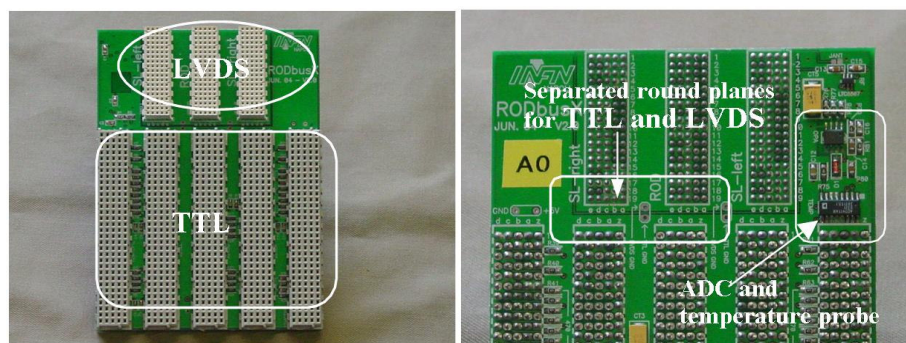


Fig. 3.5 Two images of the read section of the *ROD*bus

The design in VHDL

The design of the ROD is my original contribution to the development of the DAQ of the ATLAS experiment and it has been preceded by a proof of concept of the ROD subsystem. The integration of the ROD in the data acquisition system of ATLAS has been evaluated and the main problems that must be faced in the ROD design have been analysed. Particular care has been given, in the design phase of the ROD, to the acquisition of data from the custom backplane and to the communication with the other elements of the data acquisition system of the ATLAS experiment.

The ROD is a VME board that is based on the FPGA (*Field Programmable Gate Array*) technology. The design of the ROD has been performed by a CAD tools, that allow the simulation and the implementation of logic functions on *FPGA* programmable devices. *FPGAs* are programmable logic devices that are widely used in microelectronics because of their ability to be re-programmed in the field to fix bugs. The basic elements of an *FPGA* are programmable logic blocks and programmable interconnects. The logic block can perform basic

(AND, OR, NOT) or complex logic functions and can also include memory elements. The programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer. In the [41] a brief description of the FPGA devices can be found.

In the design phase, the programming language VHDL (*Very high speed integrated circuit Hardware Description Language*) [42, 43, 44] has been widely used. VHDL is a programming language that has been used in the last twenty years to perform the design, the models development and the simulation of digital devices. VHDL has been created at the beginnings of the '80s with the aim of producing a standard language for the description of electronics circuit, different from the traditional “*schematic*” tool with logic gates and flip-flops. The main usefulness of the VHDL are the fast times of development in code-writing and a great portability of the same project over different devices.

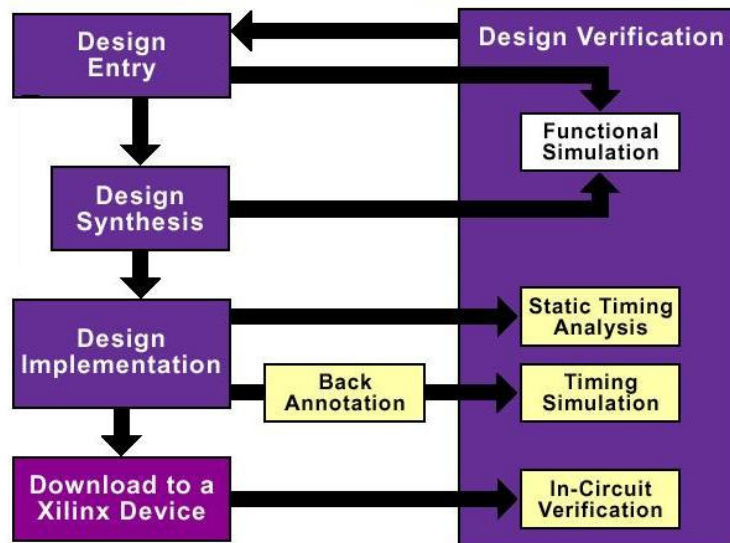


Fig. 3.6 The possible uses of a VHDL file

Fig. 3.6 shows the possible uses of a VHDL file. The file can be used for the simulation and for a proof of concept of a given project. It is also possible to proceed to the implementation of the project on a programmable device, an FPGA or an ASIC (*Application-Specific Integrated Circuit*). Indeed, a VHDL project consists of one or more files that can be elaborated by one of the commercial

compilation tools. This procedure is called synthesis. From the VHDL file, the synthesis tool produces a *netlist*, i.e. a punctual description of the hardware interconnections of logic gates and flip-flops. The fact that the *netlist* is generated by a synthesis tool is an enormous advantage offered by the VHDL design. This allows the designer to focus on the circuit description leaving the implementation to the tool. Moreover, the evolution of the production technologies of electronics devices requires project portability. A design approach that is as much as possible technology-independent is crucial to keep up with the technology evolution. The synthesis procedure allows to export a project on the up-to-date devices without changes and using only the necessary synthesis libraries.

The *netlist* undergoes a further elaboration that is the *implementation* of the project for the programmable FPGA device. During the *implementation* procedure, also the propagation delays inside the programmable device are calculated. This phase is called *Back Annotation*. By this procedure a new model, with the delays, is produced and it can be further simulated or downloaded in the FPGA.

The ROD

The main task of the ROD is to receive information from each of the two *RX/SL* boards connected to the ROD via *RODbus* and to perform a further framing, before transmitting data to the next acquisition levels, i.e. to the *Read Out Buffers* (ROB). The working principle scheme of the ROD's *event builder* is shown in fig. 3.7.



Fig. 3.7 The working principle scheme of the ROD "event builder"

The frames coming from each *RX/SL* (on the left of fig. 3.7) have an *RX Header*, a certain number of data words (*payload*) and a *footer*. The two 16-bit words of the *RX Header* contain information on L1A and BCID of the event and are checked by the ROD *event builder* engine; the payload of each frame is not inspected by ROD.

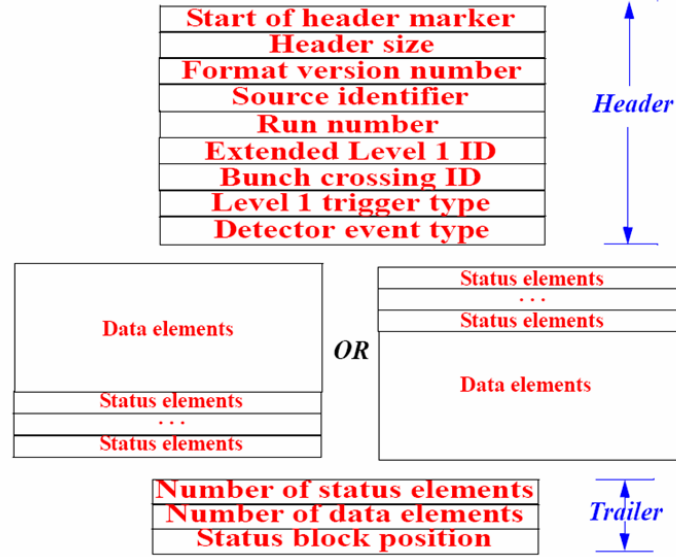


Fig. 3.8 The output data format of the ROD board

The ROD will produce a new frame, the *MUON ROD FRAME*, that will have a *ROD Header* (pertaining to a specific L1A value), a payload, made of the data coming from the selected *RX/SLs*, and a *ROD Footer*. In the *event builder* procedure the ROD performs also a control of syntactic and logical coherence on information coming from the *RX/SL* boards. In particular, the ROD detects errors in data transmission or mismatch between the *FELIID* and *FEBCID* codes (the signals generated by the *Front End* boards and contained in the *RX/SL* boards frames) and the corresponding codes transmitted by the TTC to the ROD, trying to restore the correct synchronization of data. Data coming from the *RX/SL* pertaining to the same *Bunch Crossing* parameters (*LIID* and *BCID*) are selected and written in the payload of the *Muon ROD frame*. If there is a discrepancy between some of the *LI* and/or the *BCID* identifiers, data are transmitted but flagged with one or more error bit.

The output data format of the ROD board [45] is shown in fig. 3.8: output data are 32 bit words and both ”Header” and ”Trailer” (or *footer*) are made of more words. The meaning of each word in the ”header” and ”trailer” fields will be explained in the next chapter.

About the operations of flow control and error management, some ROD features have been specifically designed. These features allow a user to:

- Obtain information about events and errors occurred.
- Obtain information about the internal working status of the board.

As already mentioned, the ROD also manages the control signals of the trigger and data acquisition system. For this purpose, the ROD hosts a TTC receiver module, the TTCrq, from which receives the control signals to be forwarded to the *RX/SL* boards. The control signals are the *LVLIAccept*, generated by the first level trigger processor to validate data related to a specific *bunch-crossing*, and the reset signals *BCR* and *ECR*. The ROD also provides the synchronous to LHC clock to the *RX/SL* boards.

The ROD has a VME interface that allows a user to access to every function of the ROD via a VME CPU, on which runs a specifically designed software. Another feature of the ROD is the option to control the power supply and the temperature on the *RODbus* backplane. This can be done by reading sensors with a microcontroller, via I²C protocol [46].

In fig. 3.9 a photo of the ROD board, completed in March 2005, is shown.

The board has the form factor of the VME 64X 6U and is equipped with two VIRTEX II [47] XILINX FPGAs, labelled in fig. 3.xx as *VME FPGA* and *ROD FPGA*. The board also hosts a microcontroller (labelled as *uP*) and the deserializers (*RX SerDes*) that receive data via *RODbus* backplane from the *RX/SL* boards. In fig. 3.9 also the slots for the TTC and S-Link modules are visible. These slots will host respectively the TTCrq receiver of the TTC optical system and the transmitter S-Link of the transmission system to the next acquisition levels. Each of these elements will be discussed later on.

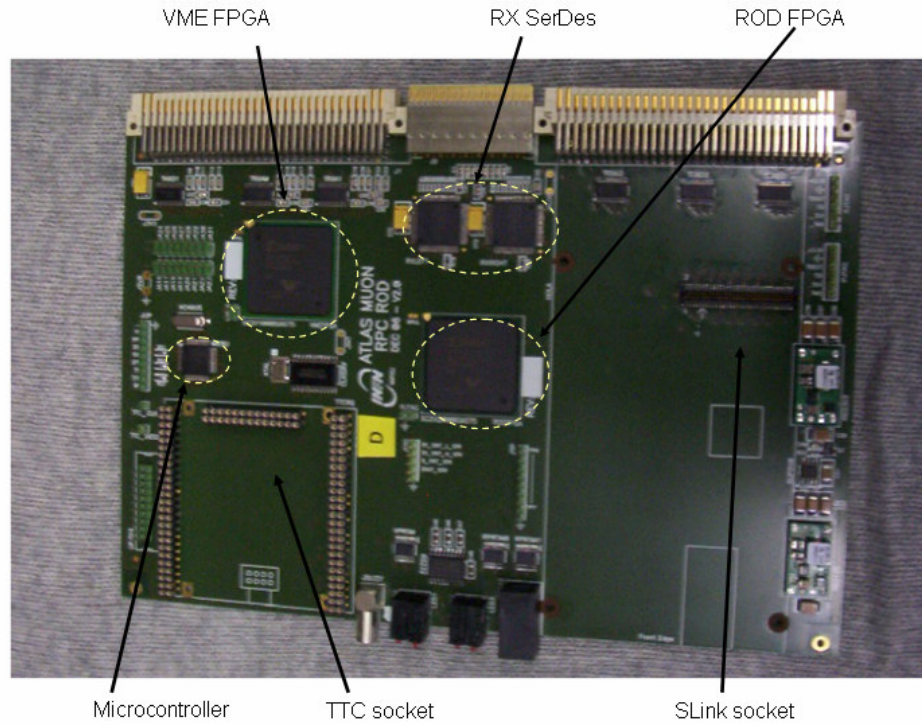


Fig. 3.9 The ROD board

Fig. 3.10 shows the block diagram of the ROD board. In this scheme, the *VMEbus*, connected to the *VME FPGA*, and the *RODbus*, connected with the two *SerDes*, are shown.

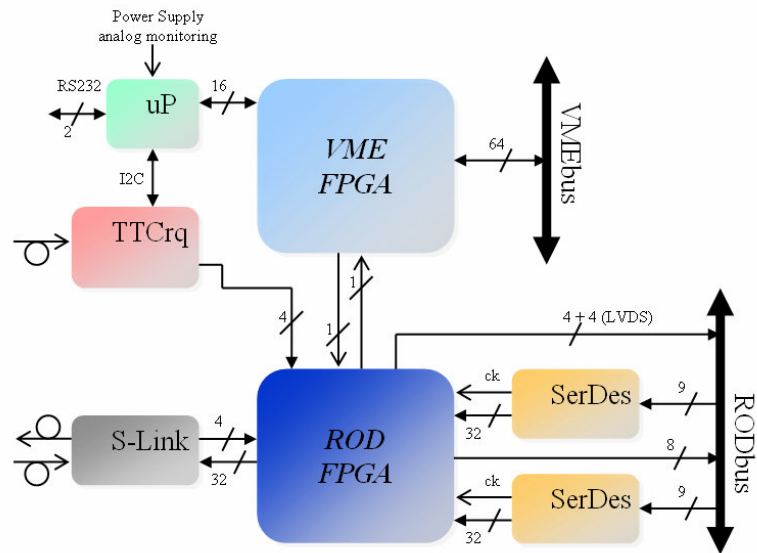


Fig. 3.10 The block diagram of the architecture of the ROD board

The VME FPGA

Fig. 3.11 shows the role played by the VME FPGA on the ROD board. The VME FPGA is a VirtexII 1000 bg575 FPGA, produced by XILINX. Its main task is to interface the whole board with the *VMEbus* and so, via the VME CPU, to the ATLAS Data Acquisition environment. In fig. 3.12 the occupation report of the FPGA, produced by the XILINX ISE development tool, is shown.

The connection with the *ROD FPGA*, that hosts the logic that performs the “*event building*”, is made by a serial custom protocol that will be described in the next paragraph. The *VME FPGA* is also connected, via a 16 bit bus, to the microcontroller *uP*, that allows the communication via I²C protocol, with the temperature and power supply sensors.

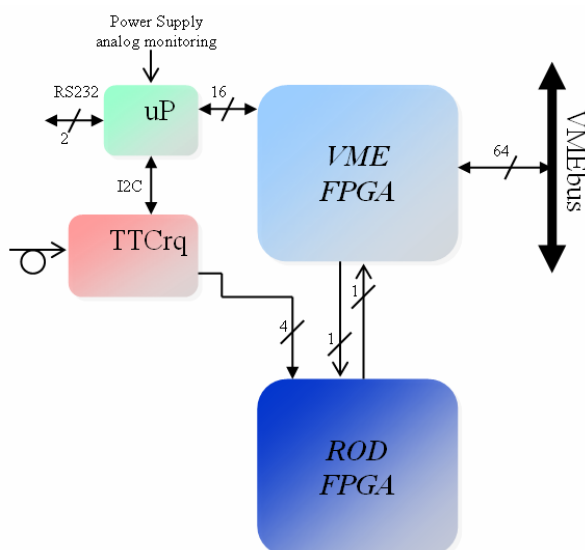


Fig. 3.11 The connections of the VME FPGA on the ROD board

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops:	1,029	10,240	10%	
Number of 4 input LUTs:	834	10,240	8%	
Logic Distribution:				
Number of occupied Slices:	907	5,120	17%	
Number of Slices containing only related logic:	907	907	100%	
Number of Slices containing unrelated logic:	0	907	0%	
Total Number 4 input LUTs:	884	10,240	8%	
Number used as logic:	834			
Number used as a route-thru:	50			
Number of bonded IOBs:	199	328	60%	
Number of GCLKs:	2	16	12%	
Number of DCMs:	1	8	12%	
Number of RPM macros:	1			

Fig. 3.12 The occupancy of the VME FPGA of the ROD board

Even if the *VME FPGA* is fed with the 40 MHz board clock, the *VME FPGA* working clock frequency has been upgraded to 80 MHz. This was made by using a clock multiplier, obtained by activating (*instantiating*) a special module of the XILINX Virtex FPGAs. Such module is the *Digital Clock Manager (DCM)* [48], that allows to implement e.g. delay locked loop, digital frequency synthesizer, or a digital phase shifter. In the case of the *VME FPGA*, the *DCM* is programmed to multiply by 2 the working frequency of the FPGA, that can have an internal work frequency of 80 MHz, i.e. the double of the board clock.

The *VME FPGA* hosts eight 32-bit registers, used to set and program the different functionalities of the FPGA. The meaning of each register is described in the Appendix A.

The serial custom protocol between the FPGAs

The *ROD FPGA* communicates with the *VME FPGA* via a serial synchronous custom protocol, carried out by two unidirectional connections. As shown in the fig. 3.13, data from the *VME FPGA* toward the *ROD FPGA* are transmitted on the *vme_to_rod* link; data from the *ROD FPGA* toward the *VME FPGA* are transmitted on the *rod_to_vme* link.

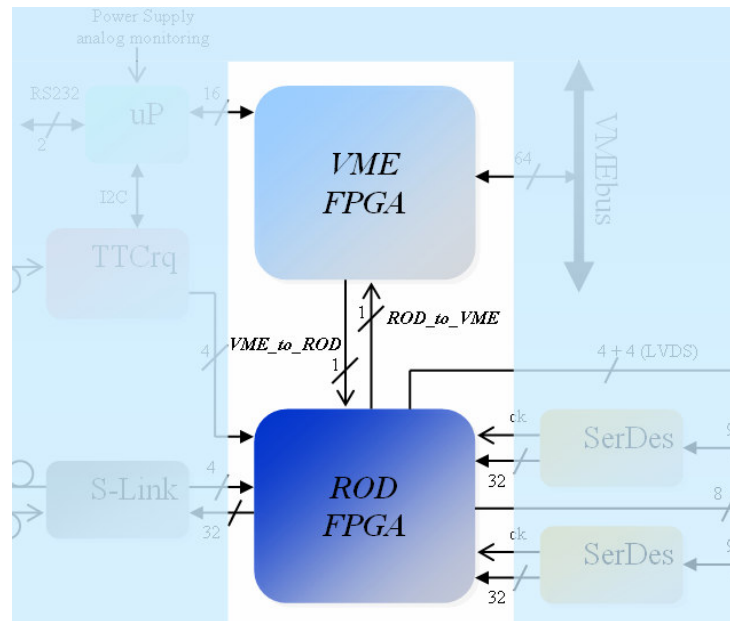


Fig. 3.13 The serial links that allow the communication between FPGAs

The main advantages of the use of a serial link are:

- A simple PCB layout.
- Limitations of *ground bounce* effects.
- Data and address bus functionalities integrated in the same protocol.
- The use of a small number of FPGA pins.

In the following, the different operations that can be performed via the serial protocol between the FPGAs are shown. The FPGA that behaves as the *Master* of the communication is always the *VME FPGA*, that manages both the write operation (for data and for address) and requires data in the read operation. As a consequence, the *ROD FPGA* can transmit data only if the *VME FPGA* had previously requested them.

The serial protocol allows to execute operation of address setting and to write data on the *ROD FPGA*; it also allows to read data from every memory location of the *ROD FPGA*. In order to access to one of the registers of the *ROD FPGA* two consecutive steps are required:

- 1) First, a write access will be necessary, in order to address the destination register;
- 2) Then a further access is required in order to:
 - transmit data that will be written in the previously addressed destination register;
 - receive data that have been read from the previously addressed destination register.

The *ROD FPGA* internal registers are identified by 8 bit involved in the transaction, and so the *ROD FPGA* can host up to 256 registers even if the used ones are only 16; each register is 32 bit wide and therefore data involved in the transaction will be 32 bit data.



The ROD FPGA

Fig. 3.16 shows the connections of the *ROD FPGA* on the ROD board. The *ROD FPGA* is a XILINX VirtexII 2000 bg575 FPGA. As previously said, its main task is to perform the *event builder* algorithm on data transmitted by the *RX/SL* boards and arriving on the ROD board via *RODbus* and *SerDes*. After being elaborated by the *event builder* engine, data are transferred to the next levels of data acquisition system via the S-Link system. The *ROD FPGA* is also responsible of correctly framing data and of the management of the controls to send information on the S-Link system. The *ROD FPGA* receives four signals (*LIAccept*, *Bunch Crossing Reset*, *Event Counter Reset* and the 40 MHz clock) from the TTC and distributes them, by LVDS connections over the *RODbus* backplane, to the two *RS/SL* boards. The *ROD FPGA* also hosts the receiver logic of the serial custom protocol that allows the *VME FPGA* (and so the *VME bus* and the *VME CPU*) to have access to *ROD FPGA* registers and data.

The ROD FPGA hosts 16 32-bit registers, used to configure the different functionalities of the FPGA. Such functionalities are, as example, the setting of the different options in the *event building* procedure, the activation of the various error handling procedures or the setup for debug procedure. The meaning of each register is described in the Appendix A.

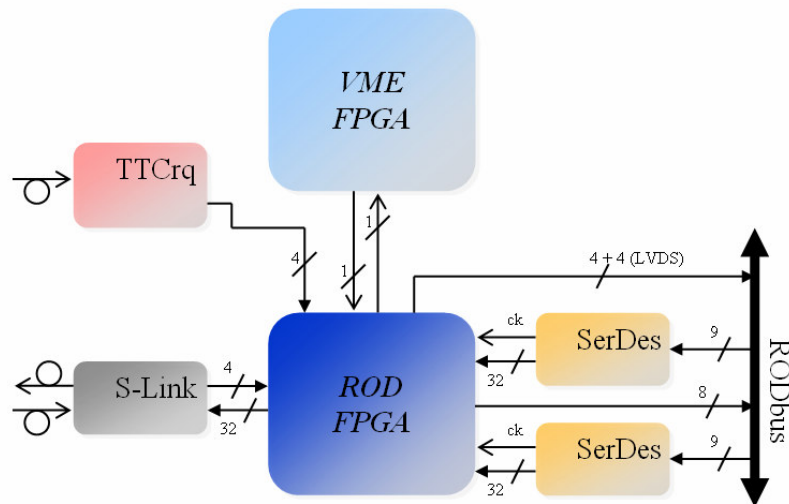


Fig. 3.16 The connections of the *ROD FPGA* on the ROD board

Fig. 3.17 shows the occupation report of the *ROD FPGA*, produced by the XILINX ISE FPGA development tool.

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slices:	2499	10752	23%	
Number of Slice Flip Flops:	3029	21504	14%	
Number of 4 input LUTs:	3804	21504	17%	
Number of bonded IOBs:	179	408	43%	
Number of BRAMs:	48	56	85%	
Number of GCLKs:	4	16	25%	
Number of DCM_ADVs:	1	8	12%	

Fig. 3.17 The occupancy of the *ROD FPGA* of the *ROD* board

Also the *ROD FPGA*, like as the *VME FPGA*, has a working clock frequency of 80 MHz. As in the case of the *VME FPGA*, this was made by using a clock multiplier, obtained by activating (*instantiating*) another *Digital Clock Manager* (*DCM*) of the XILINX Virtex FPGAs. As in the case of the *VME FPGA*, the *DCM* is programmed to multiply by 2 the working frequency of the *FPGA*, that can have an internal work frequency of 80 MHz, i.e. the double of the board clock. This is very important for the *ROD FPGA*, because the whole *event builder* engine can work two times faster than the clock frequency of the *ATLAS* apparatus and so can process data with an higher frequency.

The main Data and Control paths

Fig 3.18 shows the main Data and Control paths on the *ROD* board. Four paths are reported:

The most important path links the *ROD* board to the *RX/SL* boards via *RODbus*. This path is shown on the lower side of the fig. 3.21, with the continuous line. Data from the *RODbus* enter the *SerDes* and then enter the *ROD FPGA*. In the *ROD FPGA* data are buffered in a FIFO (*First In First Out*) memory that is very small, 32 bit with a depth of only 512 words. The task of this FIFO, as will be explained in the next paragraphs, is to decouple the different clocks arriving on the *ROD FPGA* and to synchronize all data with the board clock. Moreover, another task of the *ROD FPGA*'s internal FIFO is to allow the storage of data inside the *FPGA*. When data are read from the FIFO, they are processed by the

Event Builder engine. Then, data can be sent over S-Link to the next acquisition levels or can be sent on the *VMEbus* for monitoring.

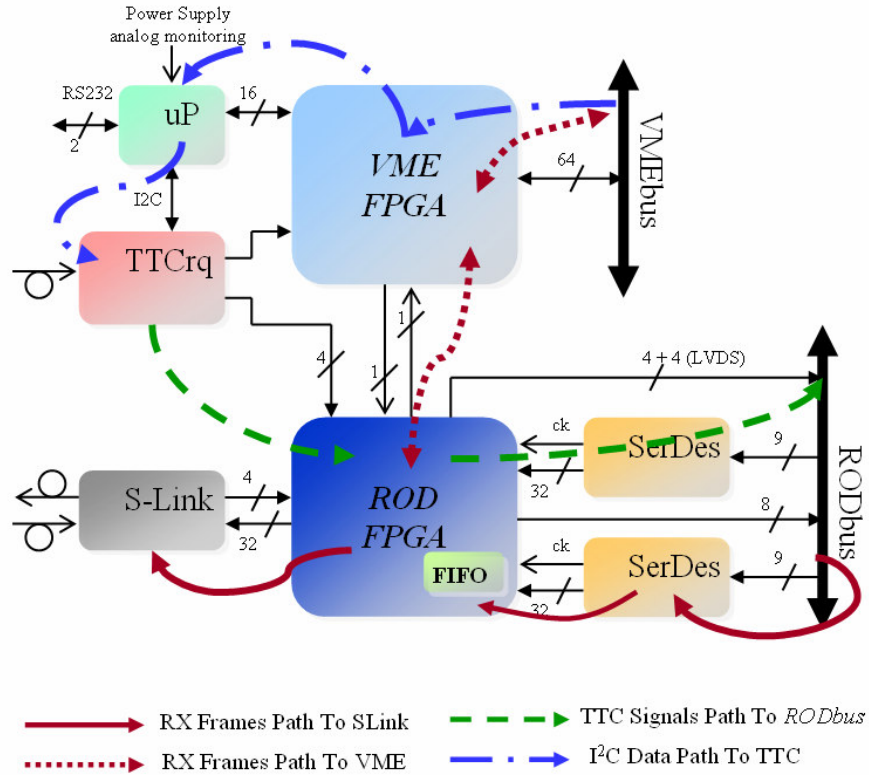


Fig. 3.18 The main paths on the ROD board

The path of data over *VMEbus* is shown with the dotted line in fig. 3.18. This path has a great importance for the debug of the *Event Builder* engine because allows to monitor the correctness of data at every step of the event building. In fact, data can be read when they are written by the *SerDes*, or when they are read from the external FIFO or when the *ROD Muon Frame* is complete and ready for the next acquisition levels. When ATLAS will be running, the DAQ system will be able to sample events (and so *ROD Muon Frames*) under software control, reading frames with a specific *LIID*, that can be set by user.

The path of the timing signals is shown with the hatched line in fig. 3.18. The four timing signals transmitted by the TTC (the *40MHz_Clock*, *L1_Accept*, *EventCounterReset*, *BuncCrossingReset*) are received by the *ROD FPGA* and distributed on the *RODbus* in *LVDS* standard.

The path of the signals for the control of the TTC is shown with the hatch-and-point line in fig. 3.18. TTC is controlled by the microcontroller via I²C interface, whereas the microcontroller receives information from the *VME FPGA*.

The serializers (SerDes)

The most evident problem in the data transfer between the *RX/SL* boards and the ROD is the great amount of data (48 bit at 40 MHz) involved in the transaction. In fact, the number of bits involved in every transmission cycle is much greater than the number of available connections on the VME backplane. That's why a custom backplane was designed to guarantee the required performances in the data acquisition system.

In the adopted solution, a commercial chipset serialize 48 bit data on 8 serial channels, with a transmission frequency of 280 MHz.

Such devices, produced by National semiconductors, are the *DS90CR483* and the *DS90CR484* [49], respectively the transmitter (or serializer) and the receiver (or deserializer) of the *chip-set*. The data transmission architecture is shown in fig. 3.19.

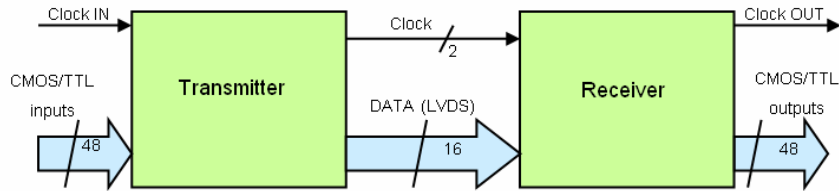


Fig. 3.19 The data path from serializer to deserializer

The transmitter accepts at inputs 48-bit data words, that are multiplexed on 8 couples of lines, following the scheme shown in fig. 3.20. Another couple of lines is used for the transmission of the synchronization signal (*clock*). These nine signals are transmitted using the LVDS standard over differential couples, with a difference between voltage levels of about 350 mV. A more accurate description of the LVDS standard is given in the next paragraph.

The receiver accepts at input the nine serial LVDS channels (data+clock) and recovers the original 48 bit words, synchronizing them with the transmitter's clock signal.

Let's give a look to the basic idea to transmit data over multiple serial connections. The *DS90CR483* and the *DS90CR484* devices acquire parallel data, transfer them with the described “pseudo-serial” protocol, over few lines, and return parallel data. So the total interconnections' number is reduced; also, the interference effects are limited by the differential transmission. In each period of the external clock, six bit of data word and an additional bit (*DC Balance*) are sent on every connection. The task of the *DC Balance* bit is to limit, through a particular inversion algorithm, the fluctuations of the mean value of the voltage on the transmission line [49]. The transmission clock frequency shall be seven times greater than the external clock: in particular, for this application in the ATLAS muon spectrometer data acquisition system, the transmission bandwidth over each line will be 280 Mbit/s.

Channel	Data [47:0]						
# 0	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	DCBAL
# 1	Data 8	Data 9	Data 10	Data 11	Data 12	Data 13	DCBAL
# 2	Data 16	Data 17	Data 18	Data 19	Data 20	Data 21	DCBAL
# 3	Data 6	Data 7	Data 14	Data 15	Data 22	Data 23	DCBAL
# 4	Data 24	Data 25	Data 26	Data 27	Data 28	Data 29	DCBAL
# 5	Data 32	Data 33	Data 34	Data 35	Data 36	Data 37	DCBAL
# 6	Data 40	Data 41	Data 42	Data 43	Data 44	Data 45	DCBAL
# 7	Data 30	Data 31	Data 38	Data 39	Data 46	Data 47	DCBAL

Fig. 3.20 - The 48 bit data traffic over the serial channels of the *DS90CR483-4* chip-set

The TTCrq

As said in the previous chapter, the TTC (*Timing, Trigger and Control*) system of the LHC delivers the level 1 trigger accept signal and all signals necessary to

synchronise the detectors, i.e. the 40 MHz clock, the *event counter reset* and the *bunch counter reset* signals.

A special timing receiver ASIC (TTCrx) has been developed by CERN for the TTC systems. The ROD board hosts the TTCrq module with the TTCrx ASIC, that produces at its output pins the TTC signals, as will be explained next. Four of these signals (the *Level 1 Accept*, the *Bunch Crossing Reset*, the *Event Counter Reset* and the 40 Mhz clock) are forwarded by the ROD to the RX/SL boards over the *RODbus LVDS* timing connections.

As indicated in the pinout scheme in fig. 3.21, this VLSI chip [31] accepts a single input from the TTC photodetector/preamplifier and generates a full range of decoded and deskewed signals for the electronics controllers. The ASIC comprises an analogue part (including the postamplifier and clock recovery/fine deskew PLLs) and a digital part (including the decoding, demultiplexing, bunch counter, event counter and command processing sections).

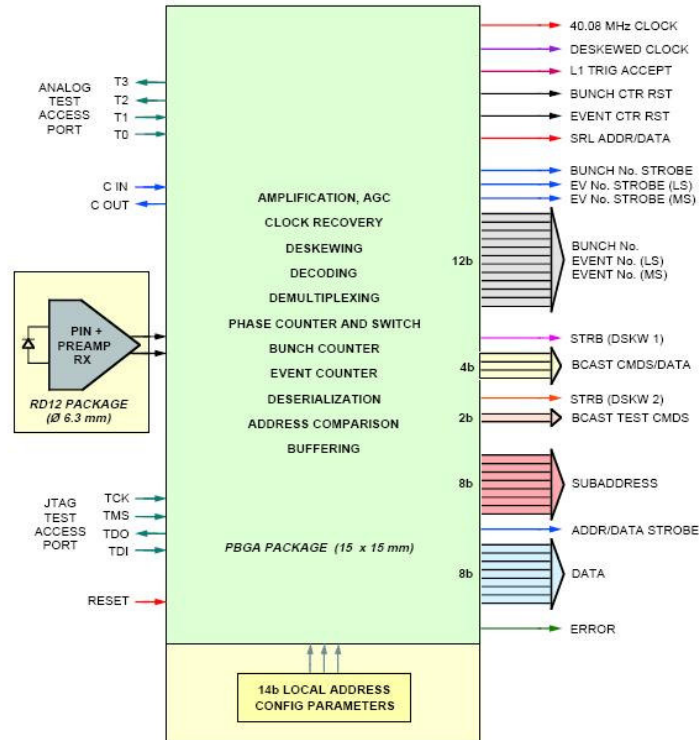


Fig. 3.21 The pinout scheme of the TTCrx receiver ASIC

In order to obtain a fine deskew resolution, the phase of the clock may be adjusted in steps of 104 ps, over the full 25 ns *Bunch-Crossing* interval. This allows to take into account the possible propagation variation due to differences in time-of-flight and optical fiber path length.

The *bunch counter number* is also provided for external use by a 12-bit counter, which delivers a unique *Bunch Crossing* number synchronously with the corresponding first level trigger decision and is integrated on-chip. During the 2 clock cycles following a trigger accept, for which the central trigger logic (global trigger) inhibits the generation of new triggers, the corresponding 24-bit *event number* is delivered on the same 12 output lines as the *bunch number*. Unlike the *bunch counter*, the *event counter* need not be reset periodically but rolls over after every 16M events (about every 3 minutes at the expected level-1 trigger rate of 100 kHz). All the TTCrx event counters are initialised by a broadcast command and may be reset by such a command during any gap in the LHC bunch structure.

The sequence of transmission of the level 1 accept signal, of the 12-bit bunch number and of the 24-bit event number is also called the “Trigger sequence”. The timing of such “Trigger sequence” is shown in fig. 3.22. The 12-bit *bunch number* generated by the TTCrx is required by the synchronization algorithms and permits the study of correlations between event data and the LHC orbit. The 24-bit *event number* suffices to detect possible problems of event ordering or loss in the data readout and event building. Additional information, rendering the event identification unique, will of course be added to the data at later stages of the DAQ chain.

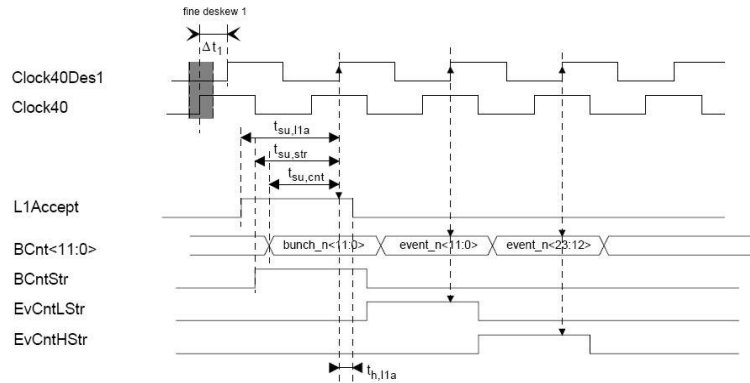


Fig. 3.22 The timing of the “Trigger sequence”

The TTC provides also for the transmission of synchronised broadcast commands and individually-addressed commands and data. The broadcast commands are decoded by all TTCrx's and can be up to 256 commands, encoded in 8 bit words. The individually-addressed commands are sent to specific chips with a certain identification number (a 14bit ID number).

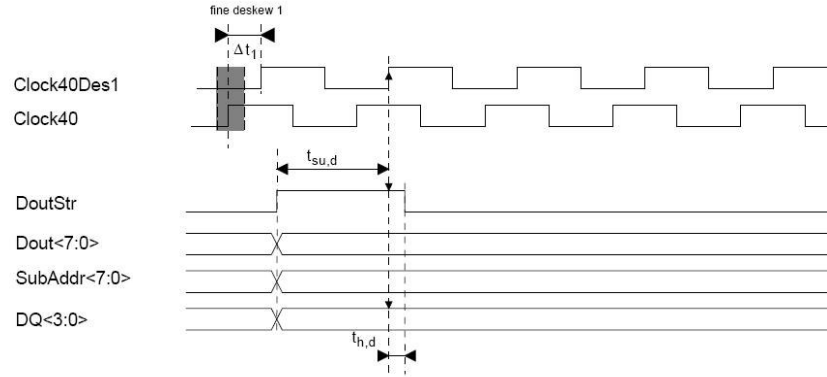


Fig. 3.23 The timing of the individually-addressed commands and data

Being the broadcast protocol common to the whole apparatus, the most interesting protocol for the ROD board logic is the individually-addressed commands protocol, reported in fig. 3.23. The net data contained in the TTC packet amounts to 16 bits. It is divided into an 8-bit *DATA* byte, and an 8-bit *SUBADDRESS* byte. At the output of the TTCrx, the net 16-bit data content appears on the **Dout<7:0>** and **SubAddr<7:0>** pins and **DoutStr** validates the signal.

To transmit the 32 bits of a generic individually-addressed word will take 4 individually-addressed command. When the TTCrx receives an individually-addressed command and decodes the 14-bit ID, the six bit *SUBADDRESS* <7:2> field will have to match with six bit written in one of the *ROD FPGA* registers, whereas the two bit *SUBADDRESS* <1:0> will span over the possible four values. The “00” value will validate the lower 8 bits of the word, i.e. *word_data* <7:0>, the “01” value will validate the *word_data* <15:8>, the “10” value will be for *word_data* <23:16> and the “11” value will be for *word_data* <31:24>.

One of the information transmitted by an individually-addressed command is the “*Trigger Type word*”. The level-1 (LVL1) trigger system generates an 8-bit

“*Trigger Type word*” for each accepted *bunch crossing*. The most-significant bit of the LVL1 trigger word (“*trigger-mode*” flag) is used to distinguish between physics triggers and calibration/test triggers. The meaning of the remaining 7 bits is different for these two cases and is described in [50]. As an example, in physics trigger mode some of the remaining bits can indicate that calorimeter and muon triggers might be treated differently, or that low p_T muon triggers might require a special treatment.

The TTC data used by the *ROD FPGA event builder* engine are the 24-bit *Bunch Crossing ID* word, the 12-bit *Level1Accept ID* word and the 8-bit *Trigger Type word*. These words are stored in two internal FIFOs, whose features are described in the next paragraphs. One 36-bit FIFO will host the 24-bit *Bunch Crossing ID* and the 12-bit *Level1Accept ID* and this FIFO will be labelled as *TTC L1 FIFO* from now on. An 8-bit FIFO will host the *Trigger Type* 8 bit data and will be labelled as *Trigger Type FIFO*. Both the *TTC L1 FIFO* and the *Trigger Type FIFO* can host 511 words. Each TTC word (*Bunch Crossing ID*, *Level1Accept ID* and *Trigger Type*) is received by the ROD and stored in the corresponding FIFO; then is used by the *Event Builder* engine in the ROD FPGA to build the ROD Muon Frame, with the format that is presented in the next chapter.

S-Link

S-LINK has been developed at CERN and can be thought of as a virtual ribbon cable, moving data or control words from one point to another. It can be used to connect any layer of front-end electronics to the next layer of read-out. S-LINKs can be constructed using different physical media and so such S-LINKs may have different timing characteristics and data transfer rates, but all they will have to comply to [51]. In fig 3.24 there is the conceptual scheme of S-Link.

In addition to simple data movement, S-LINK includes the following features;

- Error detection.
- Return channel for flow control and for return line signals (*duplex version only*).
- Self-test function.

The physical link can be conceived as the interface between the Source, on the Front End Electronics, and the Destination. The source is a Front-end Motherboard (FEMB) and the Link Source Card (LSC); the destination is the Link Destination Card (LDC) and the Read-out Motherboard (ROMB). In many applications a uni-directional data channel will be required. In other cases the user might require to send flow control and other information back to the FEMB. The S-LINK, therefore, can be constructed in either a *simplex* or a *duplex* version.

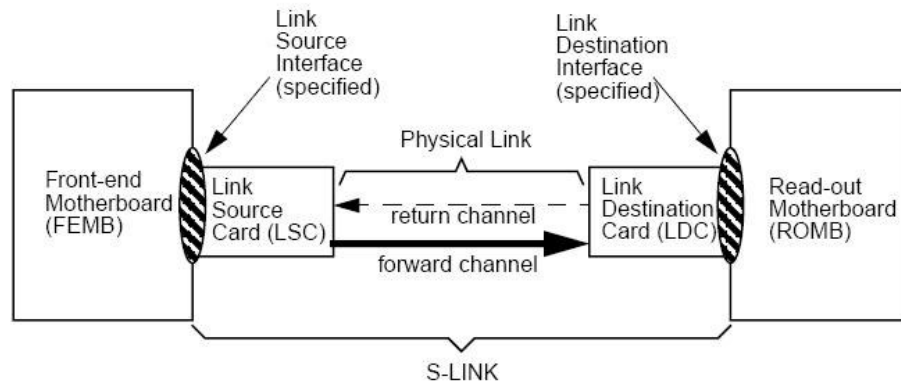


Fig. 3.24 The conceptual scheme of S-Link

In the *duplex* version, a return channel exists which allows the *LDC* to pass information back to the *LSC*. The main function of this return channel is to transmit flow control commands from the *ROMB* to the *FEMB*. Thus a duplex S-LINK transmits only when the *ROMB* is available to read the data. When the *ROMB* is unavailable, data transfers from the *FEMB* to the S-LINK are inhibited. In particular, the *ROMB* can assert a signal, *XOFF*, at the *LDC* which causes it to transmit an *XOFF* code to the *LSC*. This stops data transmission from the *LSC*. When the signal is de-asserted, an *XON* code is transmitted to the *LSC* which allows transmission to resume. This is exactly what happens on the ROD.

The fig. 3.25 shows the connections between the S-Link and the ROD board, that can be conceived as a FEMB; the mezzanine that is hosted by the ROD board represents the LSC.

The two pins that must be controlled by the ROD board are:

1) *Link Down (LDOWN)*: When low indicates that the S-LINK is not operational. It is asynchronous and can go low due to:

- S-LINK failure; then LDOWN is latched low until cleared by a reset cycle.
- S-LINK undergoing reset cycle, then LDOWN goes high when reset cycle is complete.
- S-LINK in test mode.

2) *Link Full Flag (LFF)*: Data *shall* only be written to the S-LINK when this line is high. After it goes low, up to two more words *may* be written.

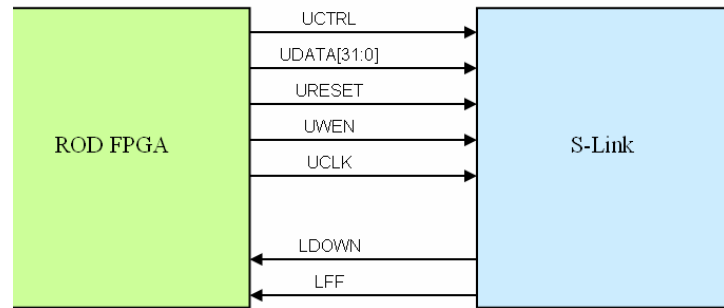


Fig. 3.25 The connections between ROD FPGA and S-Link

The pins that are generated by the ROD and are inputs to the Slink are

1) *User Clock (UCLK)*: Data is transferred to the S-LINK on the low-to-high transitions of *UCLK* when *UWEN* is low. This is a free running clock.

2) *User Write Enable (UWEN)*: When low enables data to be transferred to the S-LINK on the low-to-high transition of *UCLK*. Synchronous with *UCLK*.

3) *User Reset (URESET)*: When low initiates a reset cycle and is asynchronous.

4) *User Data (UDATA)*: Data on these lines are transferred to the *LSC* on a low-to-high transition of *UCLK* when *UWEN* is low. *UserData[3..0]* are ignored if *UCTRL* is low. *User Data* are synchronous with *UCLK*.

5) *User Control (UCTRL)*: When low indicates that the data to be transmitted is a control word. Causes *UserData[3..0]* to be ignored. Synchronous with *UCLK*.

Fig. 3.26 shows the timing of data transmission from the ROD board (i.e. the FEMB) to the S-Link mezzanine. Data transfer to the *LSC* is based on writing to a FIFO memory. The ROD provides a free-running clock, *UCLK*, which shall be present at all times. When data are to be transferred to the *LSC*, the write-enable

line, $UWEN$, is set low and the data words are transferred on each subsequent low-to-high transition of $UCLK$. Data can only be transferred to the LSC when *Link Full Flag* (LFF) line is high. If this line goes low, the S-LINK will be able to receive up to two more data words (this is to allow the ROD logic time to react to LFF). After transferring the extra words, the ROD should not try to transfer data to the S-LINK or data may be lost.

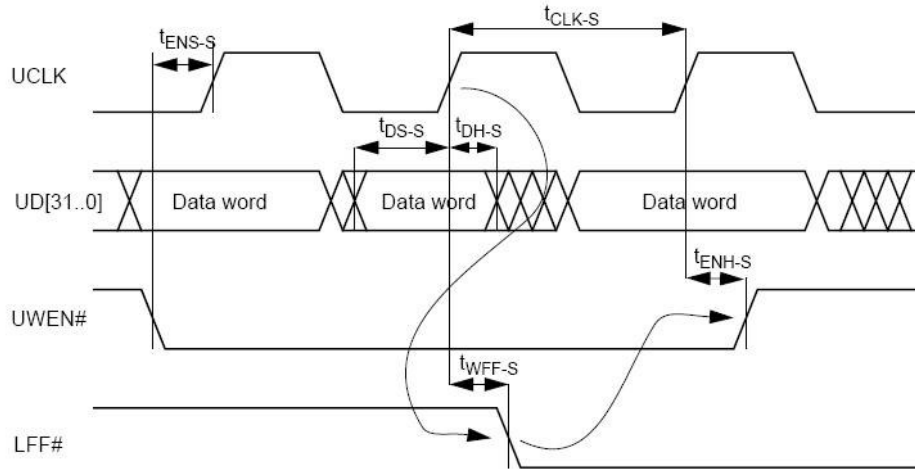


Fig. 3.26 The timing of data transmission on S-Link

The *FIFO* memories

FIFO memories are used inside the ROD FPGA. Such internal FIFOs are obtained using the internal resources of the FPGA: in particular, hardware RAM blocks of the FPGA are used. The development of the design techniques and the use of new technologies led in few years to the integration inside the FPGA devices of such RAM modules that allow to easily instantiate internal FIFOs. This was made because there was a great need of such devices, because in digital designs FIFOs are used for data manipulation tasks such as clock domain crossing, low latency memory buffering and bus width conversion.

In fig. 3.27 the pinout scheme of the Virtex II internal FIFO is displayed. Such *FIFOs* work with synchronous read and write protocols. The input is controlled by a write clock (Wr_CLK) and by the input pin *Write Enable* (Wr_En). Data (DIN) are written in the *FIFO* on every rising edge of the Wr_CLK , when Wr_En

is asserted. The output of the *FIFO* is controlled by the read clock *Rd_CLK* and by the input pin *Read Enable* (*Rd_En*). Data are put on the *FIFO*'s output pins (*DOUT*) on every rising edge of the *Rd_CLK*, when *Rd_En* is asserted.

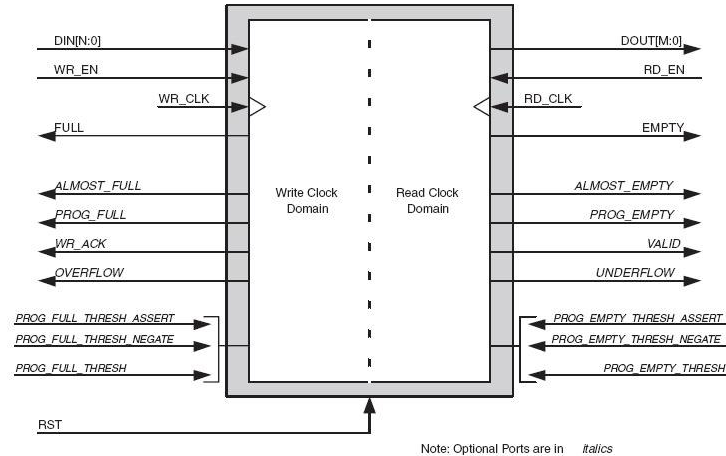


Fig. 3.27 The pinout scheme of the VIRTEX II internal FIFOs

Also the FIFO occupancy flags can be seen: full, empty, almost full and almost empty. It's worth to point out the inputs *Programmable Empty Threshold* and *Programmable Full Threshold* and the corresponding outputs *Programmable Empty* and *Programmable Full*. The *Programmable Empty Threshold* and *Programmable Full Threshold* inputs allow to program the threshold value for the assertion and de-assertion of the *Programmable Empty* and *Programmable Full* flag. In particular, the *Programmable Full* plays a key role for the *ROD FPGA* logic to have control of the flux of data: when the number of data inside the FIFO exceeds the *Programmable Full Threshold*, the *Programmable Full* flag is asserted. In an analogous way, when the number of data inside the FIFO is below the *Programmable Empty Threshold*, the *Programmable Empty* flag is asserted, even if the *Programmable Empty* flag is not presently used by the *ROD FPGA* logic.

The values of the *Programmable Full Threshold* are user programmable by software and can be set by accessing to internal registers of the *ROD FPGA*, as can be seen in Appendix A.

The *ROD FPGA* internal FIFOs play a crucial role on the *ROD board*. Their function is to allow to store data and also to act as a buffer between different clock

domains. This feature is very important because the board clock is a 40 MHz local clock, not the LHC's one.

Data arrive to the *ROD FPGA* from two different *RX/SL* boards, via *RODbus* and elaborated by two different couples of serializer-deserializer (*SerDes*). Such devices work with the 40 MHz clock recovered by the transmission clock (i.e. the clock of every *RX/SL* board) and can promote data to output with a delay of tenth of picoseconds due to the propagation delay on the *RODbus* and to the latency of the *SerDes*. Also, the data elaboration engine of the *RX/SL* boards can introduce few nanoseconds of delay: the reasons of this delay is that some operations can have a great execution latency. Hence data can arrive at different moments to the *ROD* board, via *RODbus*, even if they are always clocked by a 40 MHz clock. This problem can be solved by using FIFO memories, in particular using the clock generated by *SerDes* as write clock for the corresponding FIFO and using the board clock as the read clock for the FIFOs. With this architecture, all output data of the FIFOs are synchronous to the *ROD* board clock.

Clock domains and clock muxes

One critical problem to be faced in the design of the *ROD* board is the matching of the four concurring clock domains on the board. Such clock domains are:

- the 40 MHz clock recovered by the *TTCrx* receiver of the *TTC* signals;
- the 40 MHz clock generated by the quartz oscillator on the board;
- the 40 MHz clock recovered by the *SerDes* receiving data from the left *RX/SL*;
- the 40 MHz clock recovered by the *SerDes* receiving data from the right *RX/SL*;

Some design strategies have been developed to match all the clock domains and are described next.

- 1) The instantiation of internal FIFOs. All clock domains are 40 Mhz clocks, but as previously written, it's not possibile to predict the relationships between the different phases of the various clocks. For this reasons, the use of FIFO memories to match the different clock domains is essential. A particular care must be taken

for the *ROD FPGA*, the ROD board section where all the clock domains meet each other. In the *ROD FPGA* all the clock domains must be matched opportunely, by using internal FIFO memories.

Two FIFOs have the task to receive data from the *SerDes* and to decouple the *SerDes* clocks from the board clock: from now on, the FIFOs will be labelled as *SerDes FIFOs*. The *SerDes FIFOs* are 32bit wide and can host 4095 words.

Two FIFOs have the task to receive data from the TTC and to decouple the TTC clock domain from the board clock. These FIFOs are the *TTC LI FIFO* and the *Trigger Type FIFO* described previously.

Two FIFOs have the task to buffer the ROD Frames produced by the event builder engine, that are to be sent out of the board. One FIFO will host the 32-bit data for the S-Link and from now on this FIFO will be labelled as *S-Link FIFO*. The other FIFO will also host data produced by the *event builder* engine, but only with the purpose of sending such data on demand on the VME bus; from now on this FIFO will be labelled as *VME FIFO*. Both the *S-Link FIFO* and the *VME FIFO* are 32 bit wide and can host 4095 words.

All the six FIFOs have been instantiated and tested on the ROD board and are fully working.

2) The definition of the new 80 MHz clock domain, obtained by using 2 DLLs and spread over the *ROD FPGA* and the *VME FPGA*. The 80 MHz domain has been tested on the ROD board and is fully working.

3) The S-Link transmitter will be fed with a 40 MHz clock obtained by the division of the 80 MHz *ROD FPGA* clock. This is made in order to tune the phases of the S-Link clock with the phase of data sent to S-Link: indeed, in this way all signals are synchronous with the same clock. The firmware that controls the clock generation for S-Link has been already designed, but not yet tested on the ROD board.

Fig. 3.28 shows the spread of the different clock domains over the architecture of the ROD board.

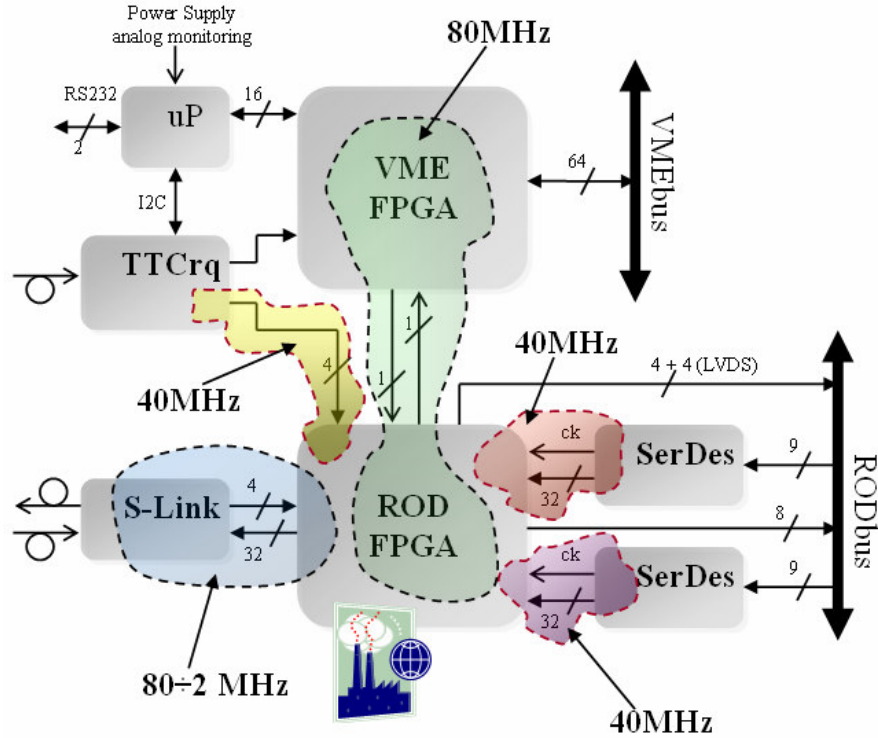


Fig. 3.28 The spread of the clock domains on the ROD board

In dealing with more than one clock domain, it's crucial to develop a test strategy. This test strategy must allow to perform the test of the ROD board, or of a part of the board, also when one or more clock sources are missing. In order to reach such task, a XILINX Virtex II internal device has been used. Such device is the clock multiplexer BUFGMUX, a multiplexed global clock buffer that can select between two input clocks and use the selected one over a certain FPGA clock domain. As shown in fig. 3.29, the source for any of the clock domains (i.e. the clock for the TTC domain and the clocks for the two *SerDes* domains) can be separately selected to be either the corresponding external clock or the 80 MHz internal *ROD FPGA* clock, generated by the *DCM*. The use of the BUFGMUX allows to perform clock switching as a safe and glitch-less operation and, as already said, allows to test the board “*stand alone*”, without TTCrq module and/or *RX/SL* boards. The selection of the source for a certain clock domain is made under software control via ROD FPGA control registers.

Another important benefit derived from the use of the BUFGMUX clock multiplexer is that the TTC and *SerDes* domains have been tested and now are

certified to fully work at full speed (80 MHz), even if their working frequency in the ATLAS experiment will be only the LHC 40 MHz frequency.

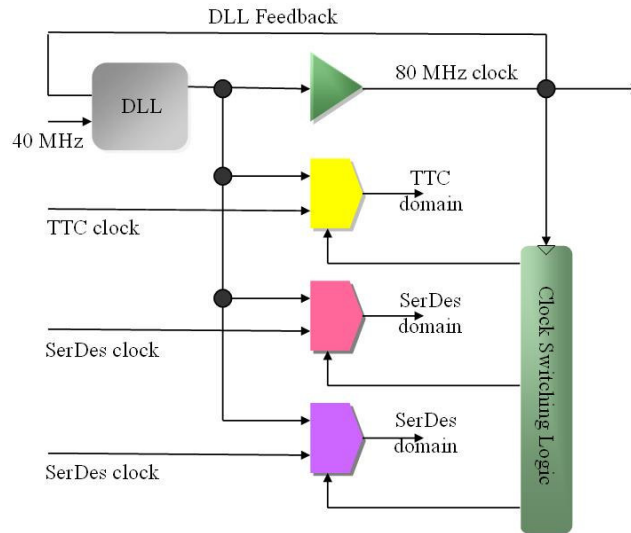


Fig. 3.29 The architecture that allows the clock switching on the ROD board

The ARM7 microcontroller

The microcontroller installed on the new layout of the ROD board is an LPC2138 [52], based on a 32/16 bit ARM7TDMI-S CPU, that combines the microcontroller with embedded high speed Flash memory. Due to their tiny size and low power consumption, these microcontrollers are ideal for applications where miniaturization is a key requirement. Moreover, with a wide range of serial communications interfaces and on-chip SRAM, they are very well suited for communication gateways and protocol converters, soft modems and voice recognition, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit 8 channel ADC(s), 10-bit DAC, PWM channels and 47 GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers particularly suitable for industrial control and medical systems.

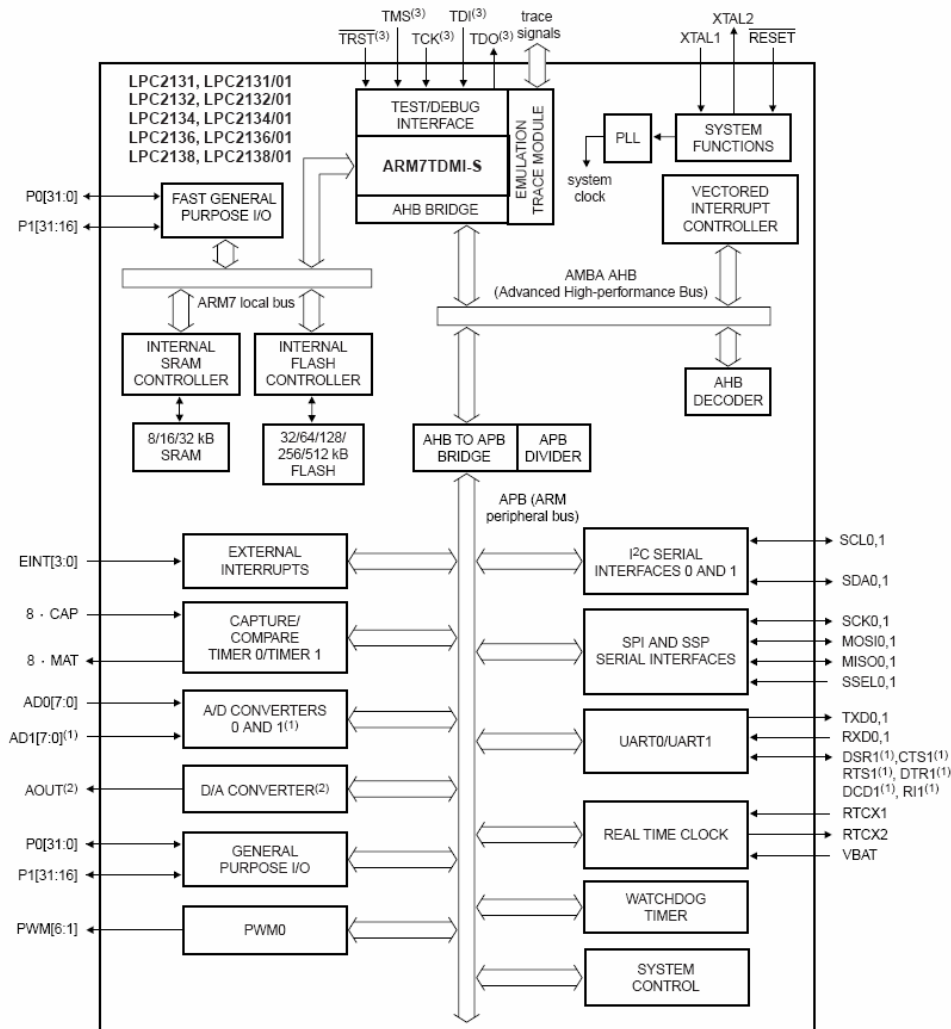


Fig. 3.30 The block scheme of the ARM7 microcontroller

In fig. 3.30 the block scheme of the LPC2138 is shown. As can be seen, the ARM7TDMI core is surrounded by many interfaces (or *bridges*) that allow the communication with the internal connections network. Among the connections, we can see:

the peripheral bus, that connects the ARM7 core to the I²C interfaces, to the serial interfaces, to the UARTs, to the A/D and the D/A converters, etc.;

the local bus, that connects to the internal memory (SRAM or Flash) controllers;

the Advanced High-performance Bus, that connects the core to the Vectored Interrupt controller, whose main task is to assign priorities of interrupts from the various peripherals.

The foremost feature of the ARM 7 microcontroller is that it is a register based load-and-store architecture. The programmer's model of the ARM 7 consists of 15 user registers, with one being used as the Program Counter (PC). Since the ARM 7 is a load-and-store architecture, an user program must load data from memory into the CPU registers, process this data and then store the result back into memory. Unlike other processors no memory to memory instructions are available.

To increase the performance of instructions' execution, the ARM 7 has a three-stage pipeline, with the FETCH, DECODE and EXECUTE operations. The hardware of each stage is designed to be independent, so up to three instructions can be processed simultaneously. The pipeline is most effective in speeding up sequential code. However a branch instruction will cause the pipeline to be flushed marring its performance.

For the use on the ROD board, the task of the microcontroller is to allow the communication, via I²C protocol, of the VME CPU with the TTCrq module. In this way, the internal configuration registers of the TTCrq module can be accessed and the TTC module can be programmed. The microcontroller also allows reading from the *RODbus* the values of the temperature and of the voltage on the bus itself.

Chapter 4

The internal logic of the Read Out Driver

This chapter continues the description of my original contribution to the ATLAS Muon Spectrometer data acquisition system, describing in detail the internal logic of the ROD.

A presentation of the Event Builder engine is given, together with an accurate description of the ROD output frame. Also, the Event Builder checks on the incoming frames are presented. The various error handling procedures are described too.

The *Event Builder* functionality of the ROD

The main task of the ROD is to receive information from each of the two *RX/SL* boards connected to the ROD via *RODbus*, to perform a logic and syntactic analysis of the received data and to execute a further framing, before transmitting data to the next acquisition levels, i.e. to the *Read Out Buffers* (ROB).

The frames coming from each *RX/SL* have an *RX Header*, a certain number of data words (*payload*) and a *footer*. The two 16-bit words of the *RX Header* contain information on L1A and BCID of the event and are checked by the ROD *event builder* engine; the payload of each frame is not inspected by ROD. The format of the *RX Frame* is shown in the fig. 4.1.

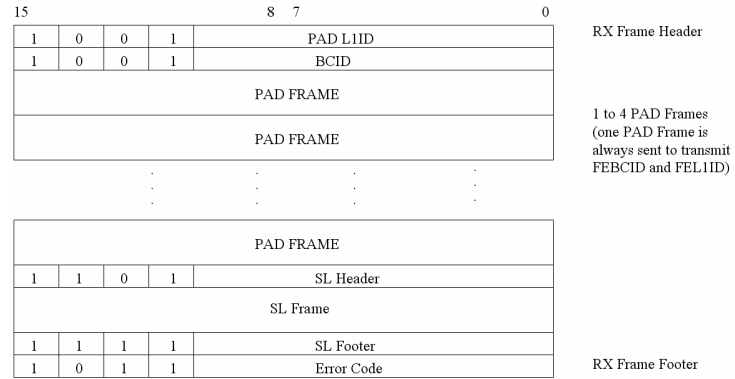


Fig. 4.1 The data format at the output of the *RX/Sector Logic* board

The ROD will produce a new frame, the *MUON ROD FRAME*, that will have a *ROD Header* (pertaining to a specific L1A value), a *payload*, made of the data coming from the selected *RX/SLs*, and a *ROD Footer*. In the *event builder* procedure the ROD performs also a control of syntactic and logical coherence on information coming from the *RX/SL* boards. In particular, the ROD can detect errors in data transmission or mismatch between the *FEL1ID* and *FEBCID* codes - the signals generated by the Front End boards and contained in the *RX/SL* boards frames - and the corresponding codes transmitted by the TTC to the ROD, restoring the correct synchronization of data. Data coming from the *RX/SL* referring to the same *Bunch Crossing* parameters (*L1ID* and *BCID*) are selected and written in the payload of the Muon ROD frame. If there is a discrepancy

between some of the L1 and/or the BCID identifiers, data are transmitted but flagged with one or more error bit.

The Output Frame of the ROD

The output data format of the ROD board [45] is shown in fig. 4.2: output data are 32 bit words and both "Header" and "Trailer" (or footer) are made of more words.

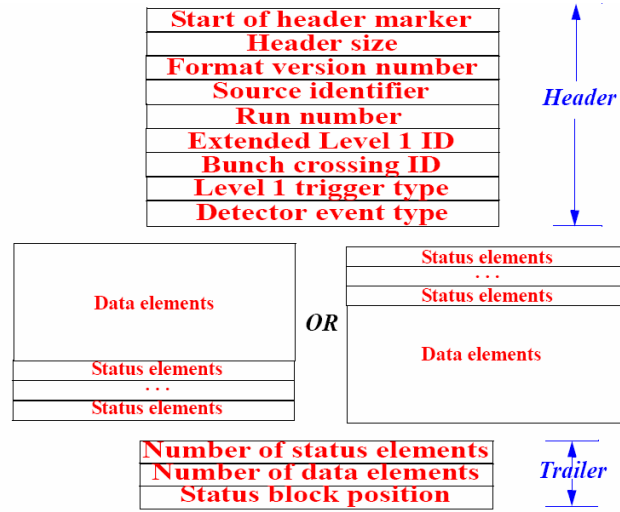


Fig. 4.2 The output data format of the ROD board

The header of the ROD Muon Frame is made of nine words, whose meanings are explained next:

1. *Start of header marker*. This marker indicates the start of a fragment header and is itself part of the header. Hence, it is the first word of a fragment. For all the elements of the DAQ system, the structure shall be identical, but the value of this element will be unique for each type of fragment. For the ROD the hexadecimal value of the "Start of header marker" is 0xee1234ee.

2. *Header size*. The element indicates the total size of the Header. Hence for the ROD the hexadecimal value of the "Header size" is 0x9.

3. *Format version number*. This element gives the format version of the fragment. It allows the format of a ROD fragment to change independently of, for example, a sub-detector fragment.

This element consists of two 16-bit fields, as shown below in fig. 4.3. The combined value of these fields identifies the fragment format version. The Major version number shall be the same for all fragments in the event, i.e. it refers to the format of the ROD Header and the ROD trailer. The Minor version number has a value dependent on the fragment type and will be used to identify the format of the specific part of the fragment header and in a ROD fragment the format of the sub-detector Data.

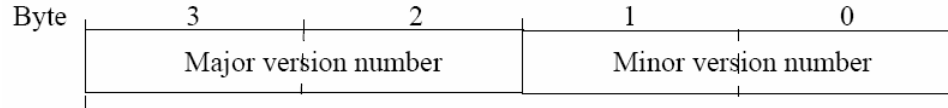


Fig. 4.3 The format version number structure of the ROD frame

4. *Source identifier*. This element identifies the origin of the fragment. It consists of a subdetector ID, Module Type and Module ID. The Module type and ID refer to the module which builds and adds the header to the event fragment.

The structure of the Source ID, as shown in fig 4.4, consists of four byte fields.

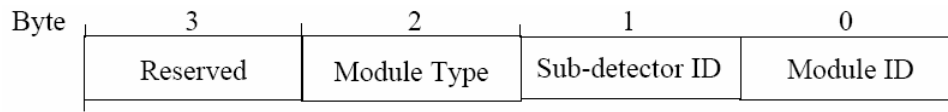


Fig. 4.4 The structure of the *Source ID* in the ROD frame

The combination of these four fields allows the Source ID to be unique across all sub-detectors. The possible values of the Sub-detector ID and Module Type for the ROD are shown in table 4.1 and 4.2. The values that may be assigned to the Module ID are free to be defined by the system or sub-system implementers concerned. The third byte is reserved and has been initialised to a value of zero.

Detector	Sub-Detector ID
<i>RPC Barrel A side</i> ($\eta > 0$)	<i>0x65</i>
<i>RPC Barrel C side</i> ($\eta < 0$)	<i>0x66</i>

Table 4.1 The possible values of the *Sub-Detector ID* in the *Source ID*

Module type	Value
<i>ROD</i>	<i>0x00</i>
<i>ROB</i>	<i>0x01</i>
<i>VMEbus module emulating ROD</i>	<i>0x0a</i>

Table 4.2 The possible values of the *Module Type ID* in the *Source ID*

5. *Run Number*: An element whose value is unique during the lifetime of the experiment. The run number is a 32-bit integer. The 8 highest bits are defined by the run control and identify the type of run. Examples of run types are: sub-detector calibration; physics running; combined detectors run, e.g. Level 1 calorimeter and a Liquid Argon sub-detector. The value of lower 24-bits represent the ordered sequence of runs within a type. The structure of the run number is shown in fig. 4.5.

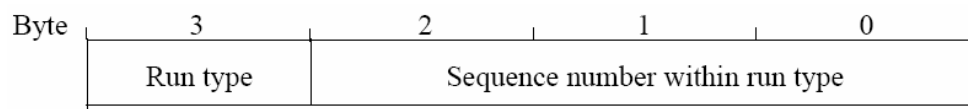


Fig. 4.5 The structure of the run number

An enumeration of possible Run Types is shown in table 4.3.

Run Type	Value
<i>Physics</i>	<i>0x00</i>
<i>Calibration</i>	<i>0x01</i>
<i>Cosmics</i>	<i>0x02</i>

Table 4.3 The possible values of the *Run Type* in the *Run Number*

6. *Extended Level 1 ID*: The Extended L1ID [53] formed by the 24-bit L1ID generated in the TTCrx and the 8-bit ECRID implemented in the ROD.

7. *Bunch Crossing ID*: The 12-bit bunch-crossing identifier generated in the TTCrx.

8. *Level 1 Trigger Type*: An 8-bit word as generated by the Central Trigger Processor and transmitted by the TTC system [50]. The remaining 24-bits are unused.

9. *Detector event type*: This element allows additional information to be supplied on the type of event, particularly in the case of calibration events. It allows the detectors to specify the exact type of calibration event that they have generated. It can be defined and modified in an internal register of the ROD.

The Trailer of the ROD Muon Frame contains the Number of data elements, Number of Status elements and the status block position. The Data and Status elements are 32-bit integers.

1. *Number of status elements*. The value of this element is the number of status elements in the *Header*. As specified in the ATLAS official documentation, for the initial implementation there should be at least one status element. Therefore this element must have a value greater than or equal to one. In the ROD implementation, there are three status elements.

2. *Status element*. This element contains information about the status of the data within the fragment. The first status word must indicate the global status of the fragment, i.e. there shall be at least one status element in a fragment.. The structure of this element is specific to the module which builds the header. Each status element is a 32-bit integer. A non-zero value of the first status element indicates that the event fragment is corrupted, *e.g.* truncated, missing data and or bit errors. The first Status element, as shown in fig. 4.6, is divided into two 2-byte fields labeled Generic and Specific. The values and error conditions indicated by the Generic field are the same for all fragments, while the values and error conditions indicated by the Specific field have meanings specific to the fragment.

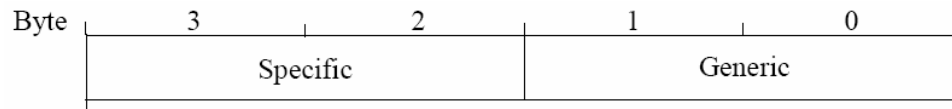


Fig. 4.6 The format of first status element in the *ROD Muon Frame*

The currently defined values and meanings of the Generic field are given in Table 4.4.

Generic field value	Description
1	<i>An internal check of the BCID has failed</i>
2	<i>An internal check of the ELIID has failed</i>
4	<i>A time out in one of the modules has occurred. The fragment may be incomplete</i>
8	<i>Data may be incorrect. Further explanation in Specific field</i>
16	<i>An overflow in one of the internal buffers has occurred. The fragment may be incomplete</i>

Table 4.4 The possible values of the Generic field in the status element

The structure and values of subsequent Status elements are free to be defined by the designers of the modules which build the specific Event fragment header.

The value of the *Status block position* defines the relative order of the Data and Status elements. A value of zero indicates that the status block precedes the data block and a value of one indicates that the status block follows the data block.

Beginning and End of fragment control words. Each ROD fragment is preceded and terminated by a single S-LINK control word. These control words are referred to as the *Beginning of Fragment* and the *End of Fragment*. The words are 32-bits integers and take the values shown in Table 4.5.

Control word	Value
<i>Beginning of Fragment</i>	<i>0xb0f0rrrr</i>
<i>End of Fragment</i>	<i>0xe0f0rrrr</i>

Table 4.5 The definition of the S-Link control words in the *ROD Muon Frame*

The lower sixteen bits of the *Beginning of Fragment* and *End of Fragment* control words are used by S-LINK to report transmission errors and are therefore subsequently reserved. All bits should be set to zero prior to transmission of the control words. Only bits [1..0] are currently used. The description and meaning of these bits is shown in Table 4.6.

	Meaning when	
	Value is 0	Value is 1
Bit 0	Previous data fragment ok	Transmission error in previous data fragment
Bit 1	Control word ok	Transmission error in control word

Table 4.6 The meanings of the control bits in the S-Link control words

The Architecture of the *Event Builder*

The block scheme of the *ROD FPGA* is shown fig. 4.8. As already said, the *ROD FPGA* is responsible of the management of ATLAS muon spectrometer RPCs' data and of the production of the *ROD Muon Frame*.

In fig. 4.7, the data flow is from right to the left. Data from *RX/SL* arrive to the FPGA as RX Data, are received by the *SerDes Interface* module and are stored in FIFOs. Data from TTC arriving to the FPGA (*L1A*, *BCID*, *Trigger Type* and controls) are received by the *TTC Interface* module and are stored in FIFOs. It's possible to select, for debug purposes, which *SerDes* channel (and so which source FIFO) to use. The selection of the source FIFO is made accessing to one of the registers in the *Configuration Register File*

From FIFOs, data are read and elaborated by the *Event Builder Engine*, working with the 80 MHz clock generated by the *2x DLL*. All the operations are managed by the *Control Logic* module, that is interfaced to the *Configuration Register File*, to the *RODbus Interface*, to the *Synchronous Serial Link* and to the Output section of the FPGA.

Event builder output data can be stored in the S-Link FIFO, that allows to transmit data to S-Link, or in the VME FIFO, that can be read via VME for debug or event spy functionality, or in both of the destination FIFOs. The selection of the destination FIFO is made accessing to one of the registers in the *Configuration Register File*. Because of the low bandwidth of the VME, the risk it to fill the VME FIFO with less than ten *ROD Muon Frames*. So data can be written in the VME FIFO only in two ways:

1) *Trap Next Data*: only the next produced *ROD Muon Frame* is written in the VME FIFO;

2) *Trap a specific L1A Data*: only the *ROD Muon Frame* pertaining to a *L1Accept* value, specified in one of the *ROD Internal register*, is written in the *VME FIFO*.

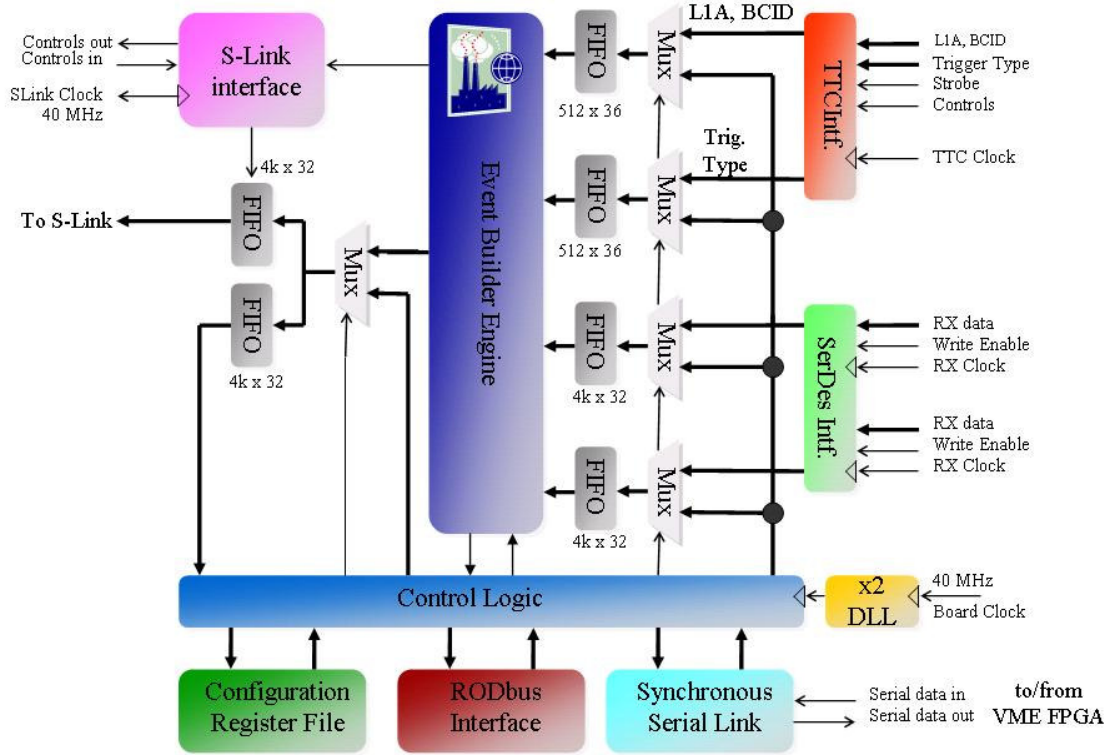


Fig. 4.7 The Block Scheme of the *ROD FPGA*

It's worth noting the presence of some *MUXes*, used for debug purposes to select data to be written in the FIFOs. Indeed, each *MUX* allows to load input FIFOs with predefined data and to run the *Event Builder* algorithm. In this way, it's easy to debug the *Event Builder* and the *Control Logic* module of the *ROD FPGA*, even without the *RX/SL* boards and/or the *TTC* receiver. Moreover, the *MUX* before the *S-Link* FIFO can be used to send known data over the *S-Link* and therefore to test the physical link and the output logic of the board.

The *Event Builder* engine

The block scheme of the *Event Builder* engine is shown fig. 4.8. The data flux is from the right of the fig. 4.8 to the left. The main element of the engine is the *Event Builder Master Machine*: it is the Finite State Machine that controls the

whole *Event Building*, both for the data flux and for the data check operations. The *Event Builder Master Machine* also receives configuration data from the *Configuration Register File* and manages the controls for the output too.

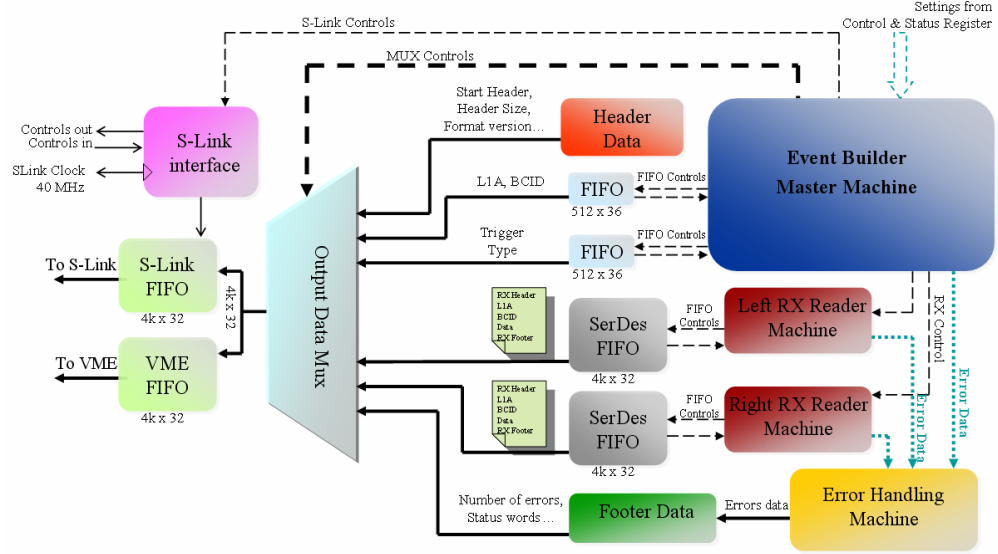


Fig. 4.8 The block scheme of the *Event Builder* engine

As described previously, the construction of the ROD muon frame can be divided in three different steps.

The first step is to write in the destination FIFO the nine words (labelled in fig. 4.8 as *Header Data*) of the *ROD Muon Frame Header*: this is made by the *Event Builder Master Machine*.

The second step is to retrieve *RX/SL* data from the two SerDes FIFOs, to parse their correctness and to package them in the destination FIFO. This operation is performed by the *Left RX Reader Machine* and by the *Right RX Reader Machine*. Hence, these two state machines have the key role of the management and the processing of the ATLAS RPC data. Moreover, the two machines also perform all the defined checks (described later in this chapter) on *RX/SL* data and manage the control signals for the *Error Handling Machine*.

The third step is to write in the destination FIFO the words (labeled in fig. 4.8 as *Footer Data*) of the *ROD Muon Frame Footer*: this is made by the *Event Builder Master Machine* too.

The basic algorithm of the *Event Builder Engine* is displayed in fig. 4.9.

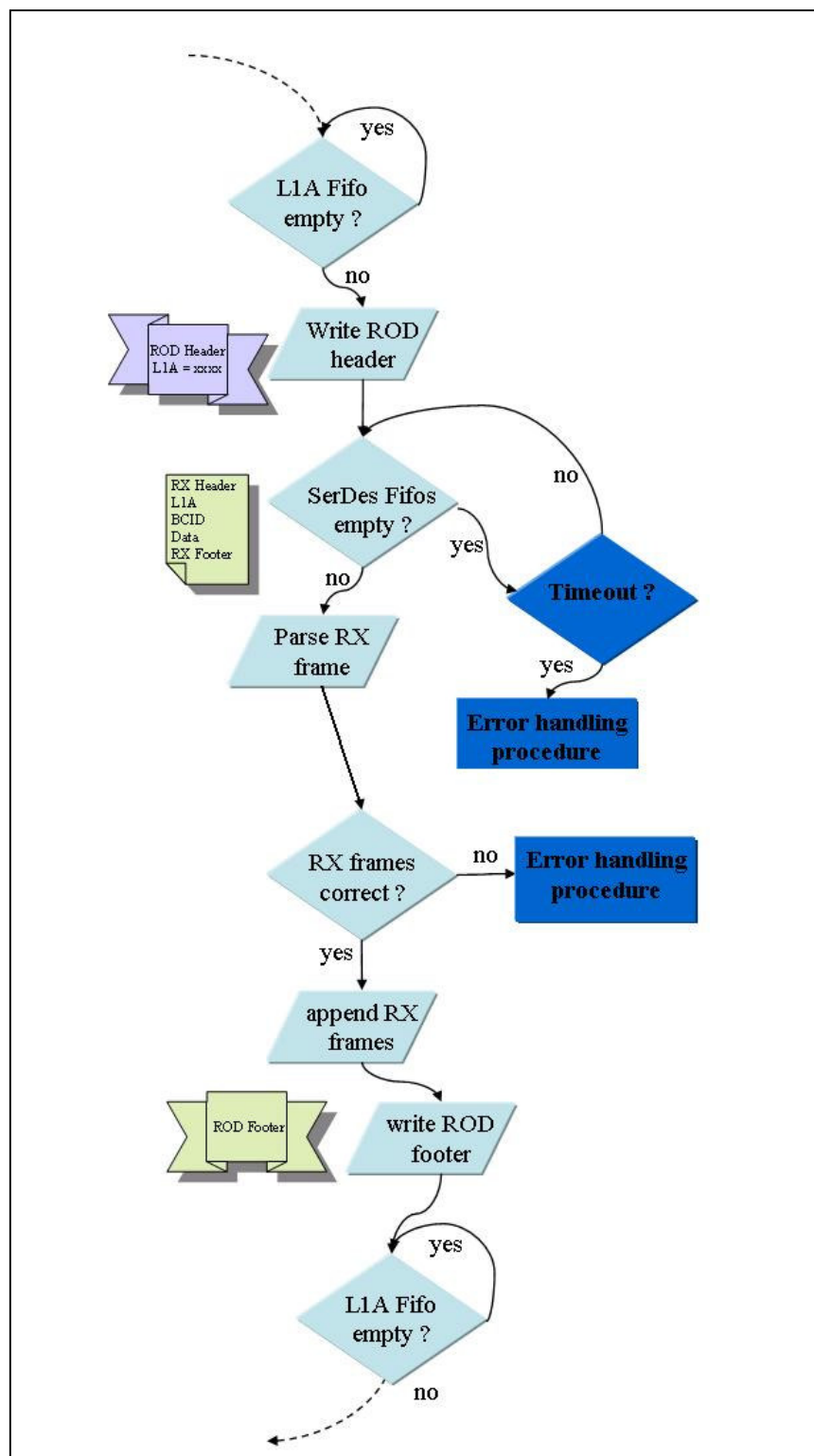


Fig. 4.9 The algorithm of the *Event Builder*

Every *ROD Muon Frame* will pertain to a specific *Level1Accept* value, so the *Event Builder Engine* will start on every *Level1Accept* data stored in the L1AFIFO. Hence, the *Event Builder Engine* checks the empty flag of the L1AFIFO and waits for a L1A to process. When the L1AFIFO is not empty and so a L1A value is available, the *Event Builder engine* starts writing a valid header into the output fifo (S-Link, VME or both).

Then, the *Event Builder Engine* searches for data in the SerDes FIFOs. As in the previous step, the *Event Builder Engine* checks the empty flag of each SerDes FIFO: if data is not available from RX boards within a programmable time window, an error handling procedure (*Timeout*) is started; otherwise RX frames are retrieved from SerDes FIFOs and are parsed to find header and L1A. If the frame is correctly formatted and the embedded L1A matches the current one, it is appended to the ROD frame. An error procedure is started elsewhere.

The ROD frame is closed by the specific footer previously described. Then the *Event Builder Engine* restarts and wait for a new *L1Accept* value.

The *Event Builder* Error Handling

While reading an RX frame, different errors can occur: syntax errors, timeout error and L1Accept errors. All error handling procedures of the *Event Builder Engine*, performed by the *Error Handling Machine*, are now described.

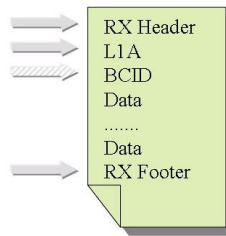


Fig. 4.10 The simplified structure of the RX board output frame

In fig. 4.10, a simplified structure of the RX frame is shown. Every RX frame is formatted with an *Header* - that contains a L1Accept value (*L1A*) and a *Bunch Crossing Identifier (BCID)* value - a payload and a *Footer*. Some of these RX frame's control fields are parsed to check consistency. In particular:

- *Header*, *L1A* and *Footer* are always parsed.
- The payload is not inspected.
- Also the total length of the RX Frame is checked.
- The *BCID* can be optionally parsed, depending on the value of a specific bit in a register of the *Configuration Register File*. This is because the *Bunch Crossing Identifier* can be not synchronized in the commissioning phase of the ATLAS Data and acquisition system and so the useless *BCID* check can be skipped.

The RX frame syntax error

Three different RX frame syntax error procedures have been foreseen. Fig. 4.11 shows examples of occurrence of such errors.

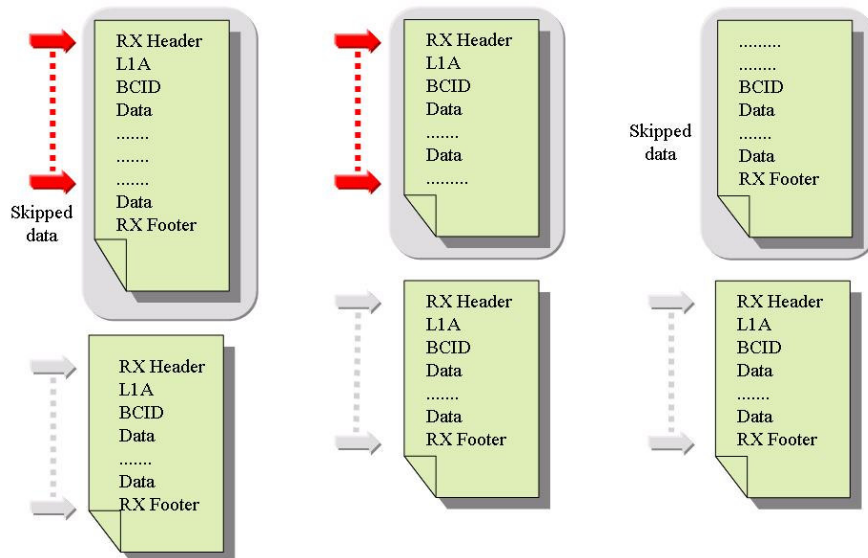


Fig. 4.11 The possible RX frame syntax errors: Jumbo frame (left column), Incomplete frame (central column) and Corrupted frame (right column)

In the first column, the so-called *Jumbo frame error* is shown. A *Jumbo frame error* is identified when the maximum allowed length of an RX frame is exceeded: this value is called *maxlength* and is a programmable value in a specific register of the *Configuration Register File*. In the case of a *Jumbo frame error*, all RX frame's data exceeding *maxlength* are skipped. There will be a realignment on next header: this means that an error flag is set in a status word of the *ROD Muon Frame* and the frame is closed; then, all *SerDes* FIFO's data are read until the next

RX Header, and such data are lost. In this way, data for the next RX frame will be correctly available.

In the second column, the so-called *Incomplete Frame error* is shown. An *Incomplete Frame* error is identified when the RX Frame is missing of the *Footer*. There will be a realignment on next header: this means that when the next *RX Header* is read from the *SerDes* FIFO, an error flag is set in a status word of the *ROD Muon Frame* and the frame is closed; data for the next RX frame will be correctly available.

In the third column, the so-called *Corrupted Frame error* is shown. A *Corrupted Frame* error is identified when the RX Frame is missing of the *Header*. There will be a realignment on next header: this means that all *SerDes* FIFO's data are read until the next RX Header, then the frame is checked and assembled. In this case, a de-synchronization between frames can occur, but there is no workaround.

The Timeout error

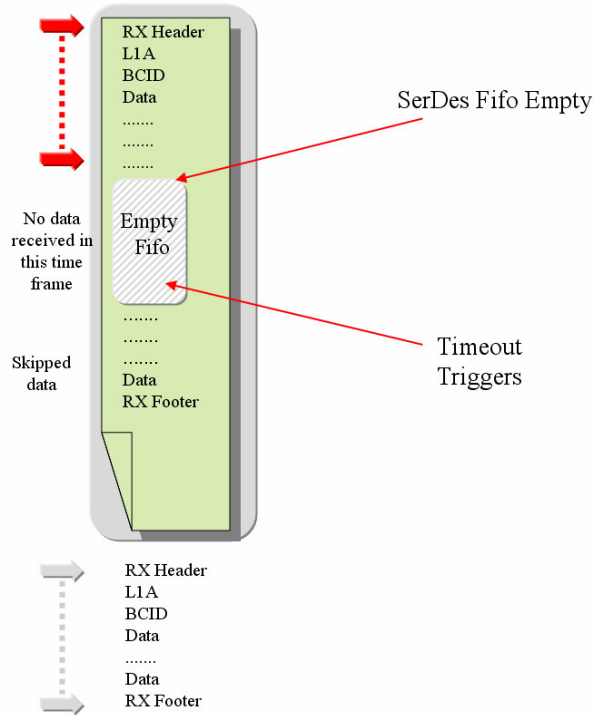


Fig. 4.12 The scheme of the occurrence of a Timeout error

Fig. 4.12 shows a *Timeout* error occurrence. *Timeout* occurs when a source *SerDes* FIFO goes empty during the event building process and no data is present in the FIFO within a programmable window. After a *Timeout*, the RX frame processing is halted, and a ROD frame is casted in the destination target FIFO with error flag set. The builder then realigns on the next available RX Header.

This means that all *SerDes* FIFO's data are read (and not written in the destination FIFO) until the next *RX Header*.

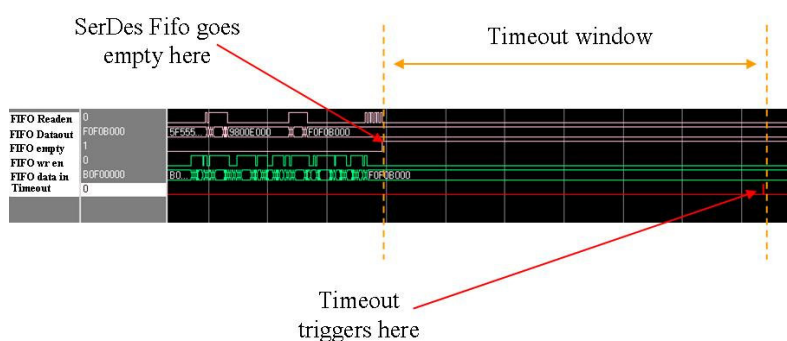


Fig. 4.13 The simulation of a *Timeout* error occurrence

Fig. 4.13 shows the simulation of the occurrence of a *Timeout* error. In the simulation the following signals are listed:

- the signal that allows data to go at the output of the *SerDes FIFO* (*read enable*);
- the data at the output of the *SerDes FIFO* (*FIFO Dataout*);
- the *SerDes FIFO empty* signal;
- the *write enable* in the destination FIFO;
- the data at the input of the destination FIFO (*FIFO datain*);
- the *Timeout* signal.

The L1Accept error

About the L1Accept values, the correct situation is that in which the L1A value contained in every RX frame matches the corresponding value transmitted by the TTC. On the left column of the fig. 4.14 there is the typical situation, i.e. a succession of correctly formatted RX frames, each pertaining to a specific TTC

L1A value. Hence, there is the first RX frame, containing the L1A=999 value, that matches with the TTC L1A =999 value arrived to the ROD. Next, the RX frame with the L1A=1000 value, that matches with the TTC one, and so on.

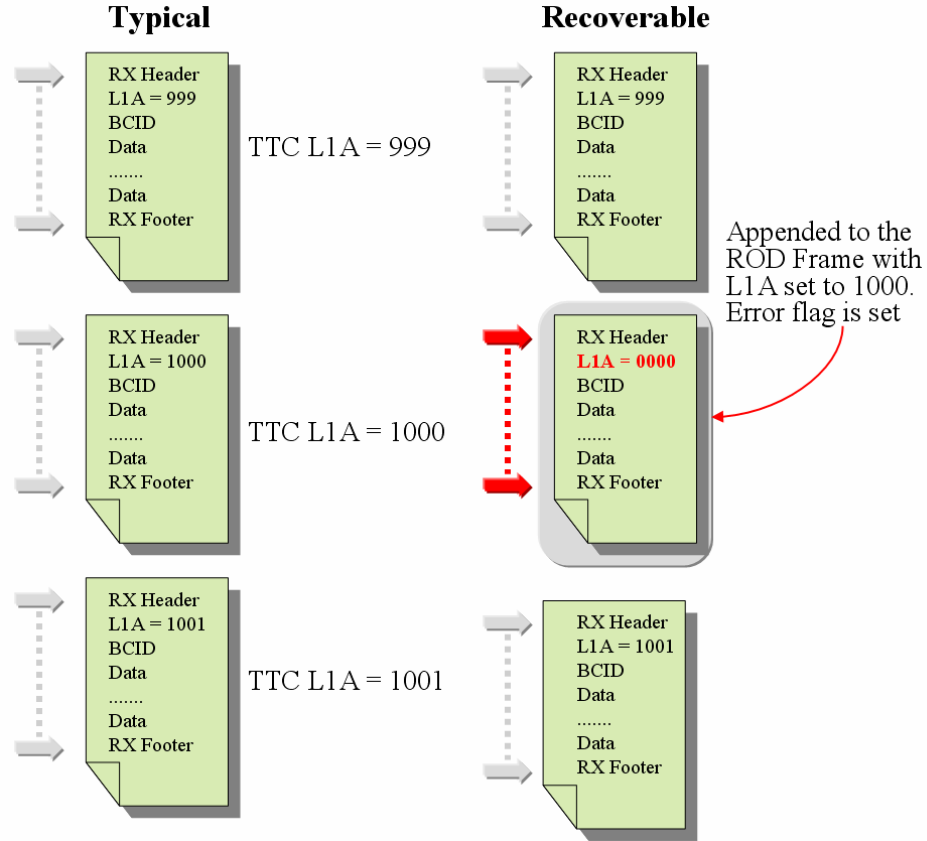


Fig. 4.14 The occurrence of a recoverable L1A error

There are two kinds of L1A errors. The first kind of L1A error is a recoverable error and it is detected when a situation like the one sketched in the right side of fig. 4.14 happens.

While the first RX frame is correctly related to the TTC L1A value, the second RX frame has a corrupted L1A value. In this example, the error is clearly due to a bit flip. The corrupted RX frame can be appended to the *ROD Muon Frame* with the L1A value set to 1000 (i.e. the TTC value), with an error flag set in a *ROD Muon Frame* status word. Then, if the next RX frame has the correct L1A value (i.e. 1001), there is no loss of synchronization and so the L1A=1000 error has been recovered.

The second kind of L1A error is an unrecoverable error and it is detected when a situation like the one sketched in the right side of fig. 4.15 happens. Also in this case, while the first RX frame is correctly related to the TTC L1A value, the second RX frame has a corrupted L1A value. In this example, the error is due to a missing frame. The corrupted RX frame can be appended to the *ROD Muon Frame* with the L1A value set to 1000, with an error flag set in a *ROD Muon Frame* status word. But the next RX Frame also has a mismatch between its own L1A value and the TTC L1A. Hence, the second corrupted RX frame can be appended to the *ROD Muon Frame* with the L1A value set to 1001, with an error flag set in a *ROD Muon Frame* status word. All the following frames will have a discrepancy in L1A values with the corresponding TTC L1A values. So, after N errors (where N is a programmable value in a specific register of the *Configuration Register File*) the *SerDes* FIFO with the synchronization problem will be put offline.

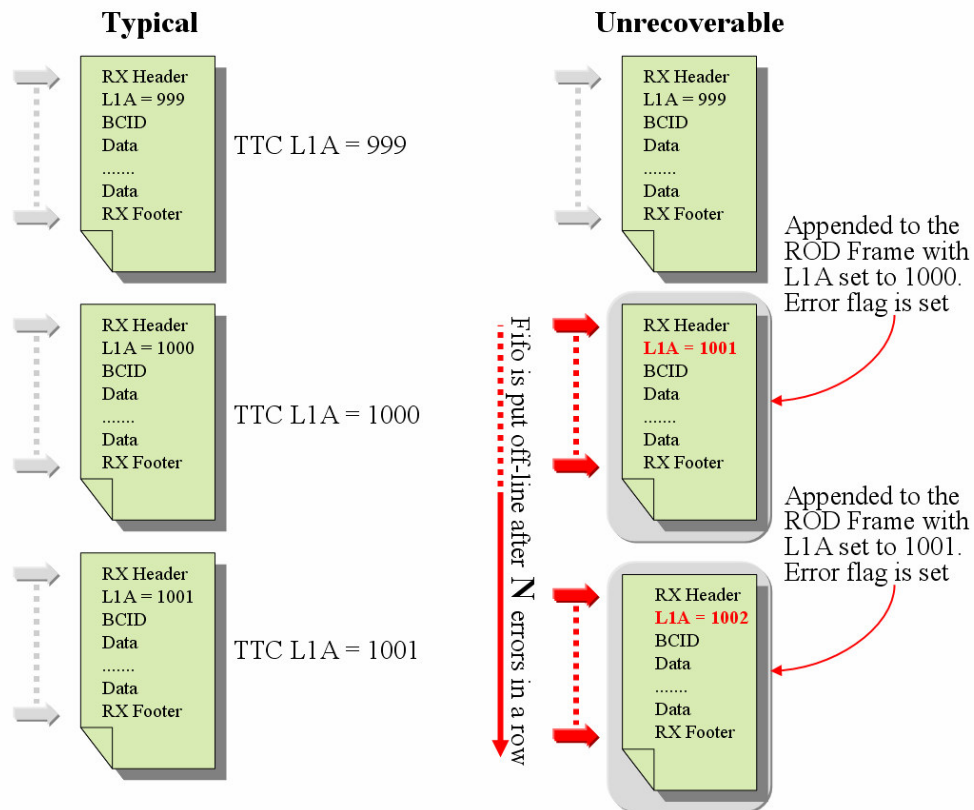


Fig. 4.15 The occurrence of an unrecoverable L1A error

The test of the ROD

Since March 2005, the ROD board has been intensively tested “*Stand alone*” in the Microelectronics Research Laboratories of the Dept. of Physics of the “Università di Napoli – Federico II”.

For more than a year, the ROD board has undergone a severe testing, useful to fix all the known bugs, to improve performances and to implement new specifications. It is now faster, easier to debug and to program. The tests have also allowed to plan new features for the future improvements of the logic functions of the board. Moreover, many of the most critical changes have been patched and successfully tested.

The 80 MHz clock domain has been tested and works fine. All internal FIFOs have been implemented in the ROD board FPGAs and all work fine. As already mentioned in the previous chapter, also the TTC domain and the two SerDes domains in the *ROD FPGA* have been successfully tested at the 80 MHz frequency, even if they will work at the LHC bunch crossing frequency, i.e. 40 MHz. The *Event Building* logic has been simulated and successfully tested pre-loading the FIFOs and reading the frame via VME.

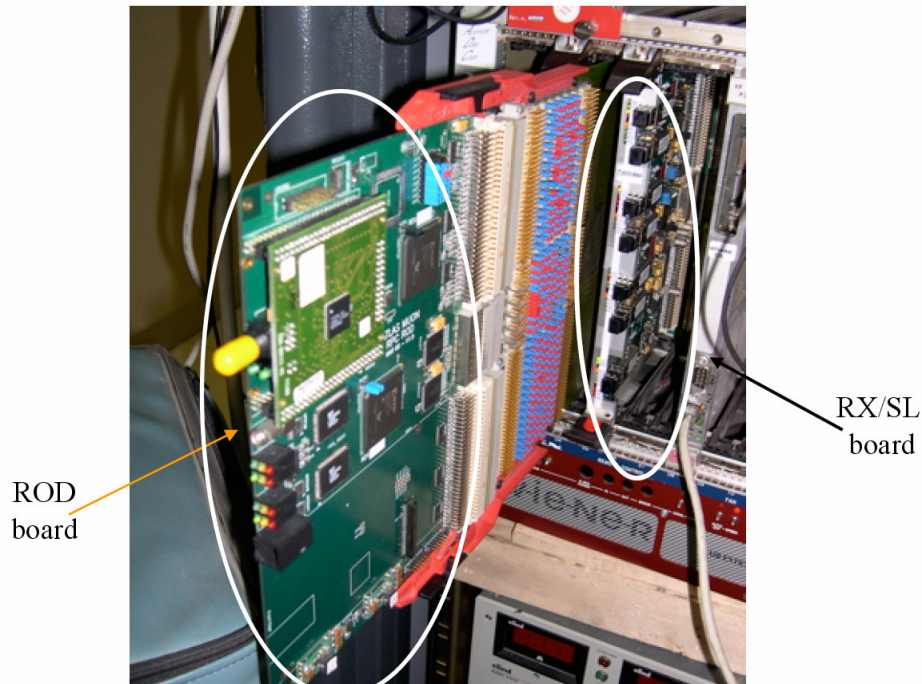


Fig. 4.16 The photo of the RX/SL and ROD combined test

In July 2006, the patched version of the ROD board has been used in an integration test in the Research Laboratories of the Dept. of Physics of the “Università di Roma – La Sapienza”. Fig. 4.16 shows a photo of the system used to perform the test.

The system was made of a ROD board and an *RX/SL* board, both inserted in a VME 64x crate and connected via a *RODbus* backplane. The VME CPU was used to access to the two boards and to execute the testing software. RX frames have been correctly exchanged between RXs and ROD. By this test, we can infer that *RODbus*, *SerDes*, internal FIFOs and 80 MHz clock domain all work fine.

Other successful test has been carried out in September 2006, in the ATLAS pit at the CERN. The system used in the test at CERN was similar to the one used in Rome. An *RX/SL* board sent data to the ROD, via a *RODbus*, in a crate controlled by a VME CPU. The test has allowed to analyze the *Event Builder* logic so far implemented in the ROD. New configuration options and new features have been tested too. The *Event Building* logic works fine, even if the error management logic and the error recovery procedures are not yet completely defined and must be definitely implemented.

Tests on the S-Link output channel are planned to be performed in the first months of the 2007, after the qualification of the *Event Builder* Logic.

The synchronization protocol between the *RX/SL* boards and the ROD board the over the *RODbus* still have to be completely defined and are planned to be tested in the first quarter of the 2007.

Tests on the TTCrq receiver programming procedure, performed via ARM7 microcontroller, are planned to be performed in the first months of the 2007.

Conclusions

This Ph.D. thesis has concerned the feasibility study and the design of the *Read Out Driver* (ROD), a VME board for the acquisition and the elaboration of data of the RPC detectors of the muon spectrometer of the ATLAS spectrometer. The *Read Out Driver* (ROD) of the spectrometer is a VME board intended to receive the information generated by the on detector electronics, to verify the logical and syntactical coherence of such data and to build a whole structure with the data and all the information related to a given detected event. Such structure (or *frame*) is then sent on optical fiber to the next trigger and acquisition systems.

The main results obtained in this Ph.D. thesis work are the design and the tests of the ROD. In particular, the main task has been the development of the ROD *event building* functionality. Such results have been achieved taking care of the integration of the ROD in the ATLAS acquisition system, both by obeying the specifications of the experiment and by interfacing the ROD with other acquisition system boards. All the ROD board functionalities have been successfully tested, respecting the ATLAS requirements. A particular care has been paid to the interfacing with the other devices used in the experiment: the physical communication channel is managed by electronics devices that transfer data in a pseudoserial protocol, with a bandwidth of 280 Mbit/s.

The design of the ROD has been performed by using a *CAD* software, that allows the simulation and the implementation of the logic functionalities of the ROD on FPGA programmable devices. In the design of the ROD, the most recent FPGA devices have been used, that allow to perform complex operations on data at a working frequency greater than hundreds of MHz. In particular, the event building logic has been optimized to guarantee a low latency and *real time* performances.

Moreover, in the design phase, the *VHDL* (*Very high speed integrated circuit Hardware Description Language*) design language has been widely used. The ROD logic, developed in *VHDL* language, can be easily exported in the future over devices that can offer performances greater than the ones of the actual FPGA devices. Starting from these considerations, the future objectives of the ROD development are to carry out further functionalities of the board and to perform integration tests in the ATLAS experiment acquisition system.

Appendix A

In this appendix, the meanings of all the 32-bit internal registers of the two FPGAs used on the ROD boards are presented.

The VME FPGA registers

In the VME FPGA, eight 32-bit configuration registers have been implemented. The meanings of each register are now explained.

The RESETS register concerns to the reset pins - of the devices connected to the VME FPGA - that can be driven by the VME FPGA. Fig. A.1 shows the data of the RESETS register. All bits are active low and are Read/Write.

BIT 1	BIT 0
MICRO Reset	ROD Reset

Fig. A.1 The VME FPGA RESETS register

Bit 0 is the reset of the ROD FPGA. Bit 1 is the reset of the Microcontroller hosted by the board. The remaining bit are reserved.

Fig. A.2 shows the content of the *Geographical address* register. It's made of six bit that allow to identify the position of the board inside the VME crate. It's a *Read Only* register. The remaining bit are reserved.

BIT [5:1]	BIT 0
GAdd	GAP

Fig. A.2 The content of the *Geographical address* register

Fig. A.3 shows the data of the *TTC_Config* register. It's the register for the configuration of the TTC receiver. The three least significant bits (*TTC Ready*, *TTC Locked* and *TTC Error*) are *Read Only* and allow to retrieve information about the working status of the TTC receiver. The *QPLL mode* (bit 3) allows to choose the frequency multiplication mode (120 MHz or 160 MHz) of the QPLL (*Quartz based Phase-Locked Loop*) of the TTC receiver.

BIT [9:6]	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
QPLL F sel	QPLL Ext Control	Auto Restart	QPLL Mode	TTC Error	TTC Locked	TTC Ready

Fig. A.3 The content of the *TTC_Config* register

The *Auto restart* (bit 4) and the *QPLL Ext control* (bit 5) allow to select the operations to be performed by the QPLL of the TTC receiver each time the lock is lost [see *qpll_manual* for more details]. The 4-bit field *QPLL F sel* (bit [9:6]) allow to control the *VCXO* (*Voltage Controlled Crystal Oscillator*) free running oscillation frequency. The remaining bit are reserved.

Fig. A.4 shows the data of the *Microcontroller_data* register. It's the register that allows the data exchange between the VME FPGA and the microcontroller. It's made of two bytes that host data that are involved in the transactions with the Microcontroller. The remaining bit are reserved.

BIT [15:8]	BIT [7:0]
Micro Data Byte 1	Micro Data Byte 0

Fig. A.4 The content of the *Microcontroller_data* register

Fig. A.5 shows the data of the *Microcontroller_controls* register. It's the register that contains information about the data exchange between the VME FPGA and the microcontroller: it contains seven bits that all are *Read Only*. RD/WR (Bit0) shows the kind (read or write) of operation performed in the transaction. All the other bits - DONE (bit 1), DAV (bit 2), LOP (bit 3), ACK (bit 4), ATTN (bit 5) and ERR (bit 6) - are used by the communication protocol between the VME CPU and the microcontroller itself. The remaining bit are reserved.

BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ERR	ATTN	ACK	LOP	DAV	DONE	RD/WR

Fig. A.5 The content of the *Microcontroller_control* register

The remaining three registers are general purpose registers and are currently reserved.

The ROD FPGA internal registers

In the ROD FPGA, sixteen 32-bit configuration registers have been implemented. The meanings of each register are now explained.

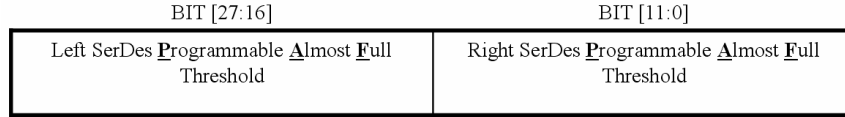
The FIFO_RESET register concerns to the reset pins of the ROD FPGA's six internal FIFOs. Fig. A.6 shows the data of the FIFO_RESET register. All bit are active low and are Read/Write.

BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S-Link	VME	Trigger type	L1A	SerDes Right	SerDes Left

Fig. A.6 The FIFO_RESET register

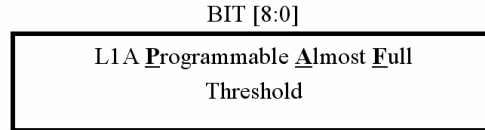
The two least significant bit (i.e. bit0 and bit1) are referred to the reset pins of the two FIFOs that receive data from the two *SerDes* (*SerDes* FIFOs); the bit2 and bit3 are referred to the reset pins of the two FIFOs that receive *Level1* data and *TriggerType* data from the TTC receiver (*L1A* FIFO and *Trig_Type* FIFO); the bit4 and bit5 are referred to the reset pins of the two output “destination FIFOs” written by the *Event Builder* logic: such FIFOs host data to be transferred to the S-Link or to the VME bus (*S-Link* FIFO and *VME* FIFO). The remaining bit are reserved. This register is also useful because the programmable flags' thresholds of a FIFO can be set only when the FIFO reset pin is low.

Fig. A.7 shows the data of the *FIFO_SerDes_PAF_Threshold* register. All bits are Read/Write.

Fig. A.7 The *FIFO_SerDes_PAF_Threshold* register

This register is used to set the *Programmable Almost Full Threshold* value for the two FIFOs that receive data from the two *SerDes*. It's made of two 12-bit fields, each for every SerDes FIFO, that allow to choose any value between 0 and 4095. The remaining bit are reserved.

Fig. A.8 shows the data of the *FIFO_TTC_PAF_Threshold* register. All bits are Read/Write.

Fig. A.8 The *FIFO_TTC_PAF_Threshold* register

This register is used to set the *Programmable Almost Full Threshold* value for the two FIFOs that receive *Level1* data and *TriggerType* data from the TTC receiver. It's made of a 9-bit fields, used for both TTC FIFO (*L1A* FIFO and *Trig_Type* FIFO), that allow to choose any value between 0 and 511. The remaining bit are reserved.

Fig. A.9 shows the data of the *FIFO_VME/SLink_PAF_Threshold* register. All bits are Read/Write.

Fig. A.9 The *FIFO_VME/SLink_PAF_Threshold* register

This register is used to set the *Programmable Almost Full Threshold* value for the two destination FIFOs, written by the *Event Builder Logic*. It's made of two

12-bit fields, each for every FIFO, that allow to choose any value between 0 and 4095. The remaining bit are reserved.

Fig. A.10 shows the data of the *FIFO_flags* register. All bit are *Read Only*.

BIT [29:25]	BIT [24:20]	BIT [19:15]	BIT [14:10]	BIT [9:5]	BIT [4:0]
Full	Full	Full	Full	Full	Full
Almost Full	Almost Full	Almost Full	Almost Full	Almost Full	Almost Full
Progr. A. Full	Progr. A. Full	Progr. A. Full	Progr. A. Full	Progr. A. Full	Progr. A. Full
Almost Empty	Almost Empty	Almost Empty	Almost Empty	Almost Empty	Almost Empty
Empty	Empty	Empty	Empty	Empty	Empty
<i>S-Link</i>	<i>VME</i>	<i>Trigger type</i>	<i>LIA</i>	<i>SerDes Left</i>	<i>SerDes Right</i>

Fig. A.10 The *FIFO_flags* register

This register is used to gather the flags of all the ROD FPGA internal FIFOs. Five bit for every FIFO are displayed, for a total of 30 bit. The remaining two bit are reserved.

Fig. A.11 shows the content of the *Clock_config* register.

BIT [19:16]	BIT [15:12]	BIT [11:8]		BIT3	BIT 2	BIT 1	BIT 0
TTC Clock Activity	SerDes Right Clock Activity	SerDes Left Clock Activity		DCM Clock locked	TTC BUFGMux selection	SerDes Right BUFGMux selection	SerDes Left BUFGMux selection

Fig. A.11 The *Clock_config* register

The three least significant bit of the register allow the selection of the clock sources for the different clock domains in the ROD FPGA. The bit3 is a *Read Only* bit that allows to monitor the *lock* pin of the *Digital Clock Manager* of the FPGA. Moreover, there are three 4-bit fields *bit[11:8]*, *bit[15:12]* and *bit[19:16]*: each field is *Read Only* and is useful to monitor the activity of the selected clock source in every clock domain of the ROD FPGA. The remaining bit are reserved.

Fig. A.12 shows the content of the *Event_builder_config* register.

BIT [31:24]	BIT [23:18]	BIT [17:8]	BIT 6	BIT 5	BIT [4:3]	BIT [2:1]	BIT 0
Timeout	Error Number	Max Length	ARM/ STOP	VME FIFO rules	Data Destinations	Data Sources	Event Builder ON

Fig. A.12 The *Event_builder_config* register

The bit 0 is the *Event_Builder_ON* bit: writing zero in this location starts the *Event_Builder_engine*. When the *Event_Builder_ON* bit is active, some operations (such as the setting of the source *SerDes* FIFO or the destination FIFO) are not possible. The 2-bit field *Data Source* (bit [2:1]) allows the user to set the source *SerDes* FIFO: it cannot be modified if *Event_Builder_ON* bit is 0. The 2-bit field *Data Destination* (bit [4:3]) allows the user to set the source *destination* FIFO and it cannot be modified if *Event_Builder_ON* bit is 0. The possible choice is between the S-Link FIFO only, the VME FIFO only, both FIFOs or no FIFO selected. Table A.1 shows the different configurations of these two fields.

BIT values	DATA Destinations	BIT values	DATA Sources
00	Both VME and SLink FIFO Selected	00	Both Sources Selected
01	SLink FIFO Only Selected	01	SerDes Left Source Selected
10	VME FIFO Only Selected	10	SerDes Right Source Selected
11	No Destination FIFO Selected	11	No Source Selected

Table A.1 The different configurations for the *Data Source* and *Data Destination* fields

The 1-bit field *VME FIFO rules* (bit [5]) allows the user to select between the two modes of trapping event builder's data in the VME FIFO. This field is read by the trap engine only when the *Data Destination* value is 00. The two possible values of this field refer to the following functionalities: trap the next available L1A value (*VME FIFO rules*=0); trap a specific L1A value, written into the *Event_Trap_config* register (*VME FIFO rules*=1).

The 1-bit field *ARM/STOP* (bit [6]) allows the user to arm or stop the functionality of trapping event builder's data in the VME FIFO.

The 10-bit field *Max_length* (bit [17:8]) allows the user to specify a value to define the *Jumbo frame* length. The 6-bit field *N* (bit [23:18]) allows the user to specify the value of consecutive errors before excluding a *SerDes* FIFO. The 8-bit

field *Timeout* (bit [31:24]) allows the user to specify the Timeout value, described in the fourth chapter.

Fig. A.13 shows the content of the *Event_builder_status* register. All the bit are *Read Only*.

BIT [13:11]	BIT [10:8]	BIT [4:0]
Left FIFO reader Machine Status	Right FIFO reader Machine Status	Event Builder Machine Status

Fig. A.13 The *Event_builder_status* register

The 5-bit field *Event Builder machine status* (bit [4:0]) allows the user to monitor the internal status of the Event Builder machine, described in the fourth chapter. The 3-bit field *Right fifo reader Machine status* (bit [10:8]) and the 3-bit field *Left fifo reader Machine status* (bit [13:11]) allow the user to monitor the internal status of the two *SerDes* FIFO reader machines, described in the fourth chapter. The remaining bit are reserved.

Fig. A.14 shows the content of the *Trigger_Type_config* register.

BIT 16	BIT [15:8]	BIT [5:0]
Trigger Type ON/OFF	Sweeper Event	TTC Subaddress

Fig. A.14 The *Trigger_Type_config* register

The 6-bit field *TTC subaddress* (bit [5:0]) allows the user to set the value of the correct board's subaddress, for the reception of an individually broadcast command from the TTC. The 8-bit field *Sweeper event* (bit [15:8]) allows the user to set the value for the so-called *Sweeper Event*: the default value is 0x07. Every time a *Sweeper event* occurs - and so every time this code is broadcast from the TTC – some of the ATLAS run parameters have been changed. The 1-bit field *Trigger Type ON/OFF* (bit [16]) allows the user to exclude (or activate) to read data from the *Trigger_Type* FIFO, during the *Event_Building* procedure. The remaining bit are reserved.

Fig. A.15 shows the content of the *Run_Number_config* register.



Fig. A.15 The *Run_Number_config* register

The 24-bit field *Run_Number* (bit [23:0]) allows the user to set the value of the *Run Number*, used to build the ROD Muon Frame. The *Run Number* field is incremented when the *Sweeper Event* is received. The 8-bit field *Run_Type* (bit [31:24]) allows the user to set the value of the *Run Type*, used to build the ROD Muon Frame.

Fig. A.16 shows the content of the *Slink_Control_Word_config* register.

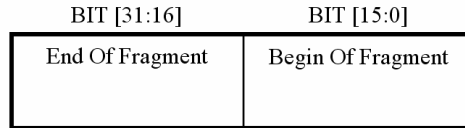


Fig. A.16 The *Slink_Control_Word_config* register

The 16-bit field *Begin_Of_Fragment* (bit [15:0]) allows the user to set the value of the *Begin_Of_Fragment* field, used to build the ROD Muon Frame. The 16-bit field *End_Of_Fragment* (bit [31:16]) allows the user to set the value of the *End_Of_Fragment* field, used to build the ROD Muon Frame The default values of these fields are respectively: 0xb0f0 and 0xe0f0.

Fig. A.17 shows the content of the *Busy_config* register. All bit are set to zero at power-up.

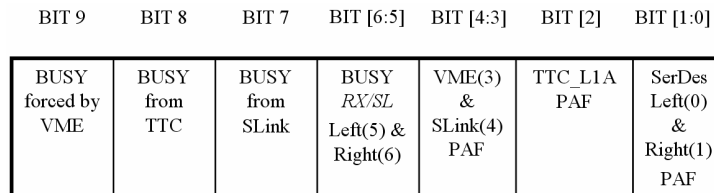


Fig. A.17 The *Busy_config* register

The 5-bit field *Busy_masks* (bit [4:0]) allows the user to exclude (or not) the internal FIFOs' busy signals from the ROD's global busy. The various bit are

referred to *SerDes Left* FIFO (bit[0]), *SerDes Right* FIFO (bit[1]), *TTC_LIA* FIFO (bit[2]), *VME* FIFO (bit[3]), *S-Link* FIFO (bit[4]). The bit from bit[5] to bit[8] allow the user to exclude (or not) the external busy sources from the ROD's global busy. The various bit are referred to *RX/SL Left* busy (bit[5]), *RX/SL Right* busy (bit[6]), *S-Link* busy (bit[7]), *TTC* busy (bit[8]). The bit *Busy_forced_from_VME* (bit[9]) allow the user to force busy via VME. The remaining bit are reserved.

Fig. A.18 shows the content of the *Detector_event_type* register.

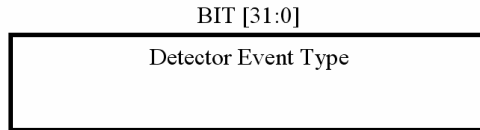


Fig. A.18 The *Detector_event_type* register

The 32-bit field *Detector_event_type* (bit [31:0]) host user-programmable information for the *Event Builder Machine*, i.e. the user can set the value of the *Detector_event_type*, used to build the ROD Muon Frame.

Fig. A.19 shows the content of the *Event_counter_reset_ID* register.

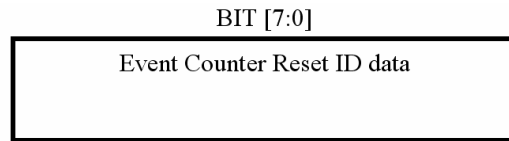


Fig. A.19 The *Event_counter_reset_ID* register

The 8-bit field *Event_counter_reset_ID* (bit [7:0]) host user-programmable information for the *Event Builder Machine*; in particular, the user can set the value of the *Event_counter_reset_ID*, that is used to build the *Extended_LIID* field in the ROD Muon Frame. The remaining bit (bit [31:8]) are reserved.

Fig. A.20 shows the content of the *Event_Trap* register.

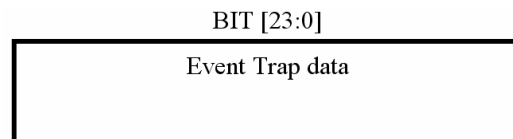


Fig. A.20 The *Event_Trap* register

The 24-bit field *Event_Trap_data* (bit [23:0]) allows the user to set the value of the *Event_Trap_data*, that is used to identify the event to be trapped into the *VME* FIFO during the *Event_Building* procedure. The remaining bit (bit [31:24]) are reserved.

Fig. A.21 shows the content of the *Actual_L1A* register. The register is *Read_Only*.

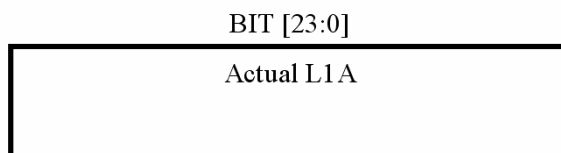


Fig. A.21 The *Actual_L1A* register

The 24-bit field *Actual_L1A_data* (bit [23:0]) allows the user to monitor the current L1A data, read from the L1A_FIFO by the *Event_Builder Machine*. The remaining bit (bit [31:24]) are reserved.

Appendix B

The LVDS standard

In the LVDS (*Low Voltage Differential Signaling*) standard, the logic signal is transmitted in a differential way over two lines, with a voltage difference of few hundreds of mV. The main advantage of this approach is that the switching times and power consumption values are lower than the other logic standards. The main disadvantage is that the limited swinging has a low noise immunity.

The input stage of an LVDS receiver is a differential amplifier, whose function is to amplify the difference between two signals.

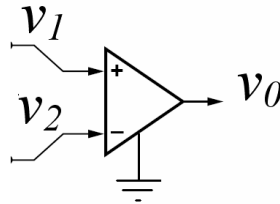


Fig. App B-1 A generic differential receiver

In the fig. App B-1 a generic differential receiver device is displayed, with two input voltages, v_1 and v_2 , and the output voltage v_0 , each referred to ground. In an ideal differential amplifier, the output voltage v_0 can be stated as:

$$v_0 = A_d (v_1 - v_2)$$

where A_d is the amplification of the differential amplifier. In this way, all the signals common to both inputs have no effects on the output voltage. But in a real differential amplifier, the output voltage v_0 is not only a function of v_d , the difference of the two input voltages, but also depends on their mean value v_c (called *Common Mode Signal*), where

$$v_d = v_1 - v_2$$

$$v_c = \frac{1}{2} (v_1 + v_2)$$

Let's express the output voltage as a linear combination of the input voltages, i.e. $v_0 = A_1 v_1 + A_2 v_2$, where A_1 (A_2) is the voltage amplification of the input 1(2) in the hypothesis that the input 2(1) is connected to ground. As $v_1 = v_c + \frac{1}{2}v_d$ and $v_2 = v_c - \frac{1}{2}v_d$, we obtain

$$v_0 = A_d v_d + A_c v_c, \quad (\text{eq. App B-1})$$

with $A_d \equiv \frac{1}{2}(A_1 - A_2)$ and $A_c \equiv A_1 + A_2$.

We can see that an high A_d is needed, while A_c should be low, and ideally it should be equal to zero. A good factor for the differential amplifier is the *Common Mode Rejection Ratio*, defined as

$$\rho \equiv \left| \frac{A_d}{A_c} \right|$$

From the App B-1 equation derives $v_0 = A_d v_d (1 + v_c / \rho \cdot v_d)$: so, the greater ρ is, with respect to v_c / v_d , the better the differential amplifier is.

The LVDS standard has the considerable advantage of being not very sensitive to a common mode "noise". Indeed, because of the differential data transmission, the presence of a disturb can be identified and removed by the receiver stage, as shown in fig. App B-2. The common mode noise immunity for the LVDS standard results to be about 3 V peak-to-peak on the newest devices.

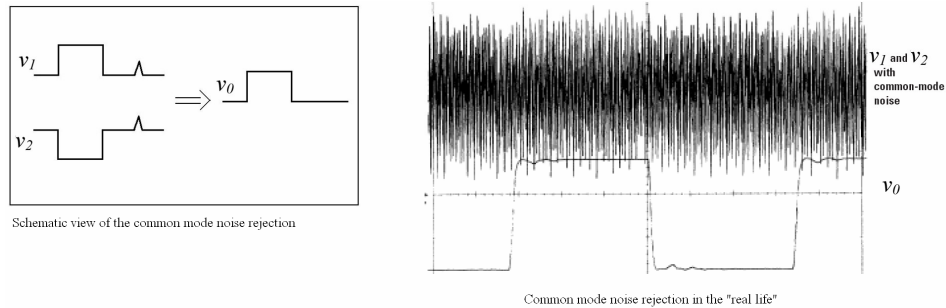


Fig. App B-2 - The elimination of a disturb over two lines of the LVDS standard

Bibliography

- [1] *The Large Hadron Collider Accelerator Project*, LHC CERN-AC-93-03
- [2] *The Large Hadron Collider Conceptual Design*, LHC CERN-AC-95-05
- [3] ATLAS Collaboration, *ATLAS Technical Proposal for a General Purpose pp Experiment at the Large Hadron Collider at CERN*, CERN/LHCC/94-43, LHCC/P2, December 1994
- [4] ATLAS Collaboration, *ATLAS: Detector and Physics Performance Technical Design Report*, CERN/LHCC/99-14, ATLAS TDR 14 (May 1999)
- [5] ATLAS Collaboration, *ATLAS: Detector and Physics Performance Technical Design Report*, CERN/LHCC/99-15, ATLAS TDR 15 (May 1999)
- [6] F. Halzen, D.A .Martin, *Quarks and leptons*, J. Wiley & Sons
- [7] LEP Electroweak working group,
<http://lepewwg.web.cern.ch/LEPEWWG/Welcome.html>
- [8] LHC Experiment Accelerator Data Exchange Working Group (LEADE),
updated Nominal LHC Beam Parameters,
<http://lhc-data-exchange.web.cern.ch/lhc-data-exchange/ruggiero.pdf>
- [9] ATLAS Collaboration, *ATLAS Inner Detector Technical Design Report Vol.1*, CERN/LHC/97-16, ATLAS TDR 4 (1997)
- [10] ATLAS Collaboration, *ATLAS Inner Detector Technical Design Report Vol.2*, CERN/LHC/97-17, ATLAS TDR 5 (1997)
- [11] ATLAS Collaboration, *ATLAS Pixel Detector Technical Design Report*, CERN/LHC/98-13, ATLAS TDR 4 (1998)
- [12] ATLAS Collaboration, *ATLAS Magnet System Technical Design Report*, CERN/LHCC/97-18, ATLAS TDR 6 (1997)
- [13] ATLAS Collaboration, *ATLAS Liquid Argon Calorimeter Technical Design Report*, CERN/LHCC/96-41, ATLAS TDR 2 (1996)

- [14] ATLAS Collaboration, *ATLAS Tile Calorimeter Technical Design Report*, CERN/LHCC/96-42, ATLAS TDR 3 (1996)
- [15] ATLAS Collaboration, *ATLAS Muon Spectrometer Technical Design Report*, CERN/LHCC/97-22, ATLAS TDR 10 (1997)
- [16] G. Viehhauser, *Detector Physics of ATLAS Precision Muon Chamber*, PhD Thesis, CERN
- [17] H. Groenstege, *The RASNIK/CCD 3D Alignment System*, ATLAS Note MUON-NO-155, 1997
- [18] V. Polichronakos, *A Proposal to Use Cathode Strips Chambers (CSC) for the ATLAS Forward Muon System*, ATLAS-MUON-94-038
- [19] R. Santonico, *Topics in Resistive Plate Chambers*, Scientifica ACTA: Resistive Plate Chambers and related Detectors; Vol. XI, n.1 pagg.1-10, Università degli Studi di Pavia, 1996
- [20] R. Santonico and R. Cardarelli, *Development of Resistive Plate Counters*, Nucl. Inst. Meth. A 187 (1981), pag. 377
- [21] R. Santonico and R. Cardarelli, *Progress in Resistive Plate Chambers*, Nucl. Inst. Meth. A 263 (1988), pag. 200
- [22] M. Alviggi et al., *Resistive Plate Chambers in ATLAS*, ATLAS-MUON-NO-96-131, CERN (1996)
- [23] R. Santonico, *RPC: Status and Perspectives*, Proceedings of “RPC93, II International Workshop on Resistive Plate Chambers and related detectors” Scientifica ACTA, Vol. VIII, n.3, pagg. 1-11, 1993
- [24] Y. Ari et al., *Thin Gap Chamber: Performance as a Time and Position Measuring Detector*, Scientifica ACTA: Resistive Plate Chambers and related Detectors; Vol. XI, n.1 pagg. 349-358, Università degli Studi di Pavia, 1996
- [25] E. Duchovni et al., *Possible Utilization of Thin Gap chambers in the ATLAS Muon System*, ATL-MUON-93-023
- [26] ATLAS Collaboration, *ATLAS First Level Trigger Technical Design Report*, CERN/LHCC/98-14 (1998)

- [27] ATLAS Collaboration, *ATLAS High-level Triggers, DAQ and DCS Technical Design Report*, CERN/LHCC/2000-17 (2000)
- [28] ATLAS Collaboration, *The ATLAS Data Acquisition and High-Level Trigger: Concept, Design and Status*, ATL-DAQ-CONF-2006-016
- [29] B.G. Taylor, *Timing Distribution at the LHC, Presented at the 8th Workshop on Electronics for LHC Experiments, Colmar, September 2002*,
<http://ttc.web.cern.ch/TTC/intro.html>
- [30] J. Varela, *Timing and Synchronization in the LHC Experiments*
lebwshop.home.cern.ch/lebwshop/LEB00_Book/plenary/varela.pdf
- [31] B.G. Taylor, *TTC distribution for LHC detectors*, IEEE Trans. Nuclear Science, Vol.45, 1998, pp. 821-828
- [32] M. Menouni, *L0(μ) Trigger High Speed Optical Links*
http://lhcb-elec.web.cern.ch/lhcb-elec/meetings/electronics_workshop_jan01/link_muon_trigger.pdf
- [33] V. Bocci et al., *The Coincidence Matrix ASIC of the Level-1 Muon Barrel trigger of the ATLAS Experiment*, IEEE Trans. Nuclear Science, Vol.50, n.4, 2003
<http://sunset.roma1.infn.it/muonl1/docs/publications/>
- [34] Final Data Format Description of the RPC Detector of the Muon System, ATL-COM-DAQ-2006-020
- [35]
http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/V1REV1/L1TDR_MuBar.pdf
- [36] Agilent HDMP-1032/1034 Transmitter/Receiver Chip Set Data Sheet,
http://mhansen.web.cern.ch/mhansen/Cms/Cms_ecal/Documentation/lp_glink.pdf
- [37] HFBR-5912E/HFCT-5912E Small Form Factor MT-RJ Fiber Optic Transceivers for Gigabit Ethernet Technical Data, <http://www.avagotech.com/>
- [38] A. Aloisio et al., "Do's and Don'ts with the Agilent G-Link chipset" Proceedings of the "14th IEEE NPSS Real Time Conference", Albanova University Center, Stockholm, 4-10 June 2005, pagg. 139-142

- [39] A. Aloisio et al., "Do's and Don'ts with the Agilent G-Link chipset", IEEE Trans. On Nucl. Sci., Vol. 53, no.3, part 1, pagg. 795-800, 2006
- [40] V. Bocci et al., *The ATLAS LVL1 Muon Barrel Sector Logic demonstrator simulation and implementation*, ATL-COM-DAQ-2000-51
<http://sunset.roma1.infn.it/muonl1/docs/notes/com-daq-2000-051.pdf>
- [41] <http://direct.xilinx.com/bvdocs/publications/ds001.pdf>
- [42] S. Sjöholm e L. Lindh *VHDL for designer*, 1996
- [43] Douglas L. Perry *VHDL*, 1991
- [44] K. Skahill *VHDL for Programmable Logic*, 1996
- [45] C. Bee et al., *The raw event format in the ATLAS Trigger & DAQ*, ATL-DAQ-98-129,
<http://atlas.web.cern.ch/Atlas/GROUPS/DATABASE/project/event/TDAQ-event-format-0019v24.pdf>
- [46] Philips Semiconductors, The I²C-bus specification,
http://www.esacademy.com/faq/i2c/I2C_Bus_specification.pdf
- [47] <http://direct.xilinx.com/bvdocs/publications/ds031.pdf>
- [48] http://www.xilinx.com/bvdocs/ipcenter/data_sheet/dcm_module.pdf
- [49] DS90CR483/ DS90CR484 48-bit LVDS Channel Link SER/DES,
<http://cache.national.com/ds/DS/DS90CR483.pdf>
- [50] Definition of the trigger-type word, ATL-DA-ES-0022
- [51] The S-Link Interface Specification, <http://www.cern.ch/HSI/s-link/>
- [52] LPC2131/32/34/36/38 Product data sheet,
http://www.nxp.com/acrobat_download/datasheets/LPC2131_32_34_36_38_3.pdf
- [53] R. Spiwoks, Presentation given to the Front-end Electronics Co-ordination - 27/02/02