

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Fisica

**A PCI Express board designed to interface
with the electronic phase-2 upgrades of the
ATLAS detectors at CERN**

Relatore:
Prof. Alessandro Gabrielli

Presentata da:
Fabrizio Alfonsi

Correlatore:
Dott. Davide Falchieri

Anno Accademico 2016/2017

Abstract

Nei prossimi 10 anni é in previsione un aggiornamento radicale dell'acceleratore LHC al CERN finalizzato al raggiungimento di piú alti valori di luminositá istantanea (oltre $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$) ed integrata (oltre un fattore 10 rispetto a quella attuale). Conseguentemente, anche i rilevatori degli esperimenti che lavorano al CERN, cosí come i loro sistemi di acquisizione dati, dovranno essere aggiornati per poter gestire un flusso notevolmente maggiore rispetto a quello utilizzato finora. Questa tesi tratta in particolare di una nuova scheda elettronica di lettura, progettata e testata nel laboratorio di elettronica del Dipartimento di Fisica ed Astronomia dell'Universitá di Bologna e nel laboratorio di elettronica della Sezione INFN (Istituto Nazionale di Fisica Nucleare) di Bologna. Le motivazioni che hanno indotto lo sviluppo della scheda prototipale sono molteplici. Un primo obiettivo da perseguire é stato quello di aggiornare la versione attuale delle schede elettroniche di acquisizione dati usate oggi nel Pixel Detector dell'esperimento ATLAS, visto che sono anch'esse sotto la responsabilitá della sezione INFN di Bologna. Secondariamente, la scheda (nominata Pixel-ROD) é orientata a gestire le esigenze elettroniche che seguiranno l'upgrade di LHC durante la fase 2. La complessitá del progetto e l'inertia intrinseca di una vasta collaborazione come quella di ATLAS, hanno poi indotto lo sviluppo di questo progetto elettronico in largo anticipo rispetto al vero upgrade di fase 2 di LHC, previsto per il 2024. In questo modo saranno anche piú facilmente eseguibili eventuali aggiornamenti tecnologici in corso d'opera, senza dover riprogettare da zero un sistema di acquisizione dati completo. Per le esigenze appena descritte, dal punto di vista hardware la scheda é stata dotata sia di moderni e veloci sistemi di interfacciamento, come la connessione PCI Express, sia di sistemi giá ampiamente collaudati, come la connessione Gigabit ethernet. Parallelamente a ció, la scheda gestisce sistemi di trasmissione dati ad alta velocitá sia su fibra ottica che tramite connessione elettrica su linee differenziali. La Pixel-ROD si basa su un'architettura gestita da due dispositivi programmabili di tipo FPGA. La prima, una Master FPGA (XILINX Zynq) ha il compito di controllo generale della scheda stessa mentre la seconda, una Slave FPGA (XILINX Kintex) ha il compito di gestire il flusso dei dati. La versatilitá di connessione e la capacitá di calcolo della scheda é potenziata da un processore fisico ARM, incluso nella Master FPGA, in grado di facilitare l'interfacciamento anche con altri progetti elettronici sviluppati all'interno della collaborazione ATLAS, che utilizzino linguaggi di programmazione standard non espressamente orientati all'elettronica. Questa tesi é composta da una parte introduttiva di presentazione dell'acceleratore al CERN, LHC, e dell'esperimento ATLAS. Poi segue una parte descrittiva del progetto e del funzionamento dell'attuale Pixel Detector di ATLAS. Successivamente la tesi descrive un possibile sistema di acquisizione dati aggiornato in visione della fase 2 di LHC, per esempio ancora relativamente al Pixel Detector, basato su schede come la Pixel-ROD. In questa parte sono descritti in dettaglio il progetto della scheda, il suo funzionamento e i test effettuati in laboratorio. I test in particolare riguardano il contributo personale che il sottoscritto ha fornito al progetto

ed alla collaborazione. Si descrivono anche i primi test congiunti, nel laboratorio del NIKHEF ad Amsterdam, con altri prototipi elettronici sviluppati dalla collaborazione internazionale ATLAS. I test sono stati eseguiti utilizzando un protocollo denominato Aurora 64b/66b, previsto nel panorama dei protocolli di comunicazione che verranno utilizzati nel futuro sistema di acquisizione dati del Pixel Detector, ed un protocollo denominato GBT (GigaBit Transmission), molto utilizzato anche adesso negli esperimenti al CERN. Infine la tesi propone una presentazione degli scenari di applicazione futura.

Contents

1	ATLAS Experiment	6
1.1	LHC Accelerator	6
1.2	LHC Experiments	8
1.3	ATLAS Detector	9
1.4	ATLAS Physics	9
1.5	ATLAS Coordinate System	9
1.6	ATLAS Composition	11
1.6.1	Inner Detector	11
1.6.2	Calorimeter	14
1.6.3	Muon Spectrometer	14
1.6.4	Magnetic System	15
1.7	ATLAS Trigger	17
2	Pixel Detector	19
2.1	Modules	20
2.2	Sensor	22
2.3	IBL	23
2.3.1	IBL Sensor and Modules	23
2.3.2	FE-I4	24
2.4	BOC-ROD System	28
2.5	BOC-ROD Communication	29
2.6	IBL-BOC	29
2.6.1	BCF	29
2.6.2	BMFs	30
2.7	IBL-ROD	32
2.7.1	ROD Controller	32
2.7.2	Spartan Slave	33
2.7.3	PRM	35
2.7.4	SBC	35
2.7.5	Lattice PLL	35
2.7.6	S-Link	36

2.7.7	TIM	36
3	Pixel-ROD	38
3.1	Pixel-ROD	40
3.1.1	FPGAs	40
3.1.2	Memory	42
3.1.3	Internal Bus	43
3.1.4	I/O	43
3.1.5	Components	44
3.1.6	Switches, Bottoms and LEDs	44
3.1.7	Power Supply System	44
4	Preliminary Tests	46
4.1	Power On	46
4.2	System Clock and Internal Bus Test	48
4.3	FPGAs RAM Tests	48
4.4	Ethernet Ports Tests	52
4.5	I2C LVDS Oscillator Tests	53
4.6	Fast Links Tests	55
4.7	PCI Express Test	58
5	Implementation	59
5.1	Aurora 64b/66b Connection	59
5.2	GBT Protocol	61
6	Future Developments and Conclusion	63
7	Appendix	65
7.1	FPGA	65
7.2	JTAG	67
7.2.1	Boundary Scan	67
7.2.2	Registers	69
7.2.3	Test Access Port Controller	69
7.2.4	Boundary Scan Instructions	69
7.3	AXI4	71
7.4	IBERT LogiCore IP	73
7.5	Aurora Protocol	75
7.5.1	Data Transmission and Reception	75
7.5.2	Frame Transmission Procedure	76
7.5.3	Frame Reception Procedure	77
7.5.4	Flow Control	78

7.5.5	Initialization and Error Rate	78
7.5.6	PCS Layer and PMA Layer	79
7.6	Aurora Logicore IP	82
7.6.1	Clock and Reset Management	82
7.6.2	User Signals in Transmission and Reception	83
7.6.3	Flow Control	84
7.6.4	Error Signals	86
7.6.5	System Debug	86
7.6.6	Other Features	86
7.7	I2C Protocol	87
7.8	FM-S14	90
7.9	RD53a	90
7.9.1	Floorplan and organization	90
7.9.2	Analog Front End	92
7.9.3	Digital Matrix	92
7.9.4	I/O and Configuration	92

Chapter 1

ATLAS Experiment

1.1 LHC Accelerator

LHC (Large Hadron Collider) is the biggest ring collider in the world, with a circumference of 27 km; its purpose is to accelerate protons and heavy ions which will collide and consecutively allow the study of the high energy physics. It is situated below the city of Geneva, between the French and the Swiss border, 100 m under the road. Its collocation is the same of the old Large Electron-Proton collider (LEP), and a collaboration of 22 nations known as CERN (Conseil Européen pour la Recherche Nucleaire) works together at high energy physic experiment. In Figure 1.1 is shown the LHC underground complex. The collider is composed of a particle source and a series of accelerators which together should permit the proton beams collisions with a total center of mass collision energy of 14 TeV, even if now the maximum energy reached is 13 TeV, 6.5 TeV for each proton. The proton source is an Hydrogen container where the particles are split in their fundamental components, a proton and an electron; then the protons are collected and then are sent to the first stage of acceleration of LHC, LINAC2, a linear accelerator in which the protons reach the energy of 50 MeV. Then the protons are accelerated by 3 synchrotron accelerators before reaching the last accelerator (LHC): PSB (Proton Synchrotron Booster) where the protons reach 1.4 GeV, PS (Proton Synchrotron) where the protons reach 25 GeV, and SPS (Super Proton Synchrotron) which accelerates protons at 450 GeV. At the end, LHC, with radio frequency cavities working at 400 MHz, pushes the beams of protons at 6.5 TeV each, in its beam pipes. In Figure 1.2 the acceleration chain of LHC is shown. The proton beams in LHC are maintained in the 2 beam pipes and far from its walls by a complex system of magnets, precisely with 1232 superconducting dipole magnets, which maintain the beam in the beam pipes, and with 392 quadruple magnets for the beams focusing. After the acceleration, the beams collide in the 4 collision points where the 4 experiments now active at CERN are situated. The final beam consists of a 2808 bunches of protons, where the bunches are formed by the

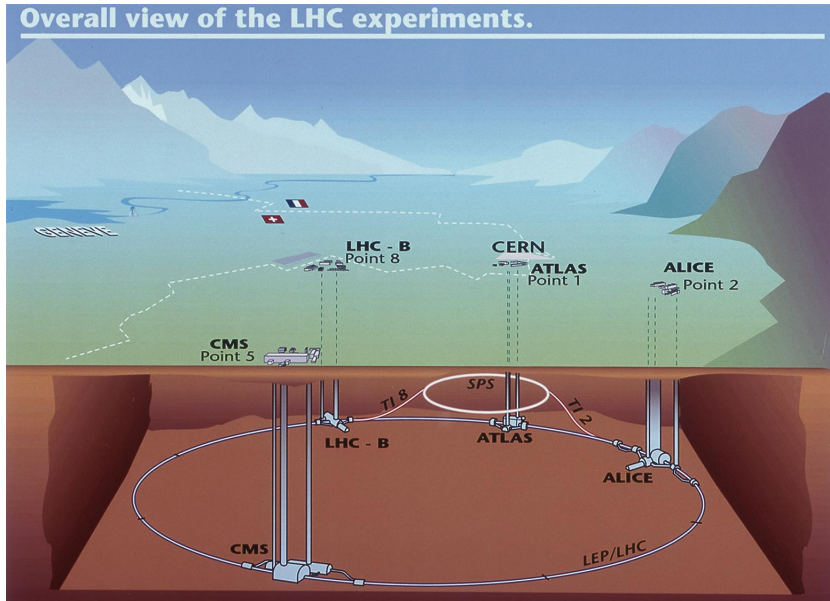


Figure 1.1: Scheme of the underground LHC complex.

CERN's Accelerator Complex

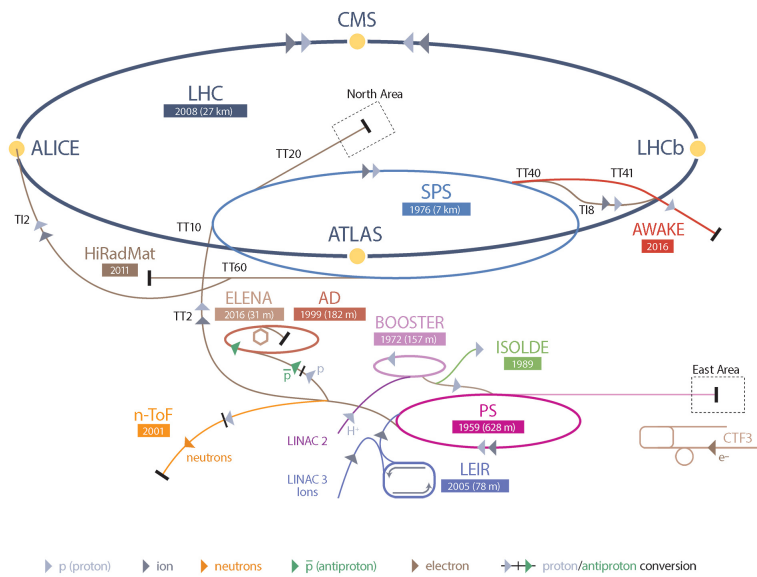


Figure 1.2: Scheme of the accelerators complex of LHC.

radio frequency cavities, with $\simeq 1.2 \times 10^{11}$ protons in each bunch. The collisions permit a luminosity with a peak of $10^{34} \text{ cm}^{-2}\text{s}^{-1}$.

1.2 LHC Experiments

As we said, the bunches of the beams collide in correspondence of the 4 experiments now active at LHC: ATLAS, CMS, ALICE, LHCb; each one of them is born for a specific motivation and task.

- ATLAS (A Toroidal LHC ApparatuS) is a multipurpose detector. It's composed of a series of detectors which surround the beam pipe: an Inner Detector (ID) for the particle tracking, a solenoid magnet to measure the particles momentum, an Electromagnetic Calorimeter to measure the energy of electromagnetic interactive particles, a Hadronic Calorimeter which measures the energy of particles which interact by Strong Interaction, a Muon Spectrometer to detect muons and their tracks and momentums;
- CMS (Compact Muon Solenoid) is another multipurpose detector like ATLAS which is built with different technologies but with the same layout and purpose;
- LHCb (Large Hadron Collider beauty) is a specific apparatus for proton-proton collisions where they are studied in a particular way. Its purpose is to investigate the physics of the quark b, in particular the CP-violation of the hadron B. The apparatus is composed of a tracker around the proton interaction region and, forward the tracker: a RICH detector, a series of other trackers, another RICH, an electromagnetic calorimeter, a hadronic calorimeter, and at the end a muon detector;
- ALICE (A Large Ion Collider Experiment) is an apparatus born to study Pb-Pb collision, where each couple of particles colliding reach an energy of 2.76 TeV. It studies the QCD physics, in particular the condition of high temperature and high energy density. It is composed of 18 detectors surrounding the collision point, including: a time projection chamber, a transition radiation chamber, a "time of flight" detector, electromagnetic and hadronic calorimeters, and a muon spectrometer.

These detectors have changed the way we see physics thanks to their results, such as the detection of the Higgs boson (ATLAS, CMS in 2012), the detection of charmless charged 2-body B decays (LHCb), the detection of the quark-gluon-plasma state of matter (ALICE).

1.3 ATLAS Detector

The multipurpose detector ATLAS is 46 m long and with a diameter of 26 m. The confirmation and improvement of the values of the Standard Model and the study of the theories that go beyond it are the objectives of this apparatus, which involves over 3000 physicists from over 175 institutes.

1.4 ATLAS Physics

The general purpose detector type of ATLAS and its detecting features have put it on forefront in the research for the study of the Standard Model in general. The detection of the Higgs boson ($m=124.98 \pm 0.98$ GeV) opened a new research area studying all its decays, in particular the decays that involve the quark b, and improved the knowledge of the characteristics of this boson. Another area under study at ATLAS is the improving of our knowledge of the quark top, which scope is to reach the Standard Model limits, finding new decay process at lower cross sections and hints of new physics. One last, but not the last, research area is the investigation of the most accepted theory for the idea of new physics, the Super Symmetric Model. The latter, at this point, has not been proved yet to be the natural extension of the Standard Model, together with other research like the extra-dimension and many other theories of new physics, that at the moment did not give any results.

1.5 ATLAS Coordinate System

The ATLAS coordinate system isn't a complex or a particular system. The interaction point is considered to be in the coordinate "0:0:0" in a 3-D Cartesian axes system, with coordinates x:y:z, where the coordinate z is along the beam line, and the x-y plane is perpendicular to the beam line with the positive x-axis points to the center of LHC accelerator and the positive y-axis points upward to the sky (see Figure 1.3). In this transverse plane the position coordinates of a particle are given by the distance R from the center of the plane and by the azimuthal angle ϕ in the transverse plane, from the x-axis around the z-axis (see Figure 1.4). The particle momenta measured in the transverse plane is called transverse momenta p_T . The polar angle θ is in the z-y plane from the positive z-axis. An important coordinate is the pseudorapidity, defined as $\eta = -\ln tg(\theta/2)$, which allows to measure the distance ΔR from two particle by the formulation $\Delta R = \sqrt{(\Delta\eta^2 + \Delta\phi^2)}$. The pseudorapidity range goes from 0, alongside the y-axis, and infinity, alongside the z-axis. The high energies involved in the proton collisions makes the partons of the protons, where each parton carry a fraction of the proton momentum, collide. For the analyses of the collisions are used the transverse

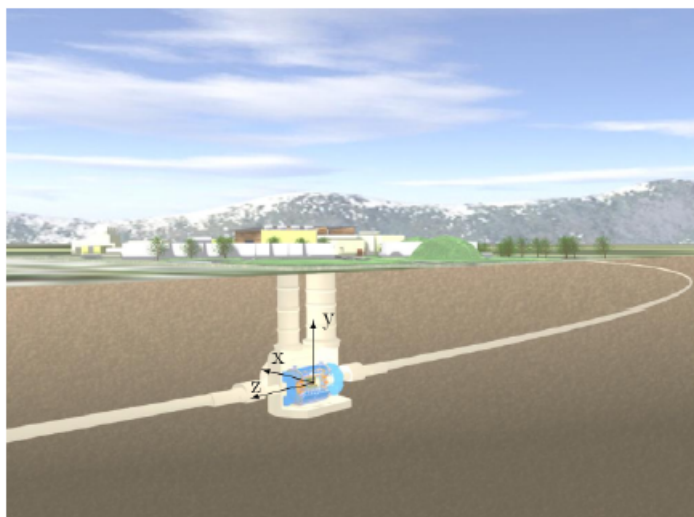


Figure 1.3: *Scheme of the ATLAS coordinates.*

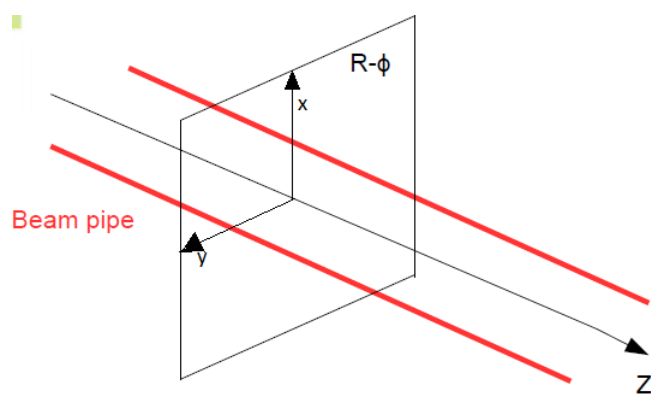


Figure 1.4: *Scheme of the used ATLAS coordinates.*

momentum p_T , the transverse energy E_T , and the transverse missing energy E_T^{miss} .

1.6 ATLAS Composition

As we said, ATLAS is an apparatus composed of different detectors, where each one cover a η range and has a particular and important purpose. Immediately after the proton-proton collision, the first detector with which the product particles of the collision can or can't interact is the Inner Detector (ID), a tracking apparatus formed by 3 detectors surrounding the beam line and covering the region $|\eta| < 2.5$. Here the precision of the particle track is very high, with an intrinsic accuracy varying approximately from 10 to 100 μm , accuracy that allows to obtain the first and the second vertex of the reaction. Immediately outside ID, the first stage of the Magnet System encountered by the particles is a central solenoid which provides a 2 T magnetic field and performs a momentum resolution for the p_T measurements of $\sigma_{p_T}/p_T = (4.83 \pm 0.16) \times 10^{-4} \text{GeV}^{-1} \times p_T$. To this point, only charged particles can be detected. After this stage, the next detector is the Electromagnetic Calorimeter, in which electrons, positrons and photons are detected by the Electromagnetic Interaction that permits to form electromagnetic shower for each particle, shower where the particles produced in it are detected for the measurements. We measure the energy of these particles and the track of the shower. After that, the more energetic particles that survived encounter the Hadronic Calorimeter where, thanks to the Strong Interaction, the hadrons born by the proton-proton collisions form hadronic showers detected by the detector with an energy resolution from 0.13 to 0.06 when jet p_T increases. The calorimeter stage covers an angle up to $|\eta| = 4.9$. After this stage, only the particles with a very low cross section survived, mainly muons and neutrinos. The latter can't be detected by ATLAS, indeed they are studied with the Energy Missing technique. The muons instead can be detected and studied, and this task is performed by the Muon Spectrometer, a very large detector. It is composed of 2 tracking chambers and 2 trigger chambers and has a coverage $|\eta| < 2.7$. In Figure 1.5 is shown the ATLAS layout.

1.6.1 Inner Detector

The Inner Detector is a 6.2 m long apparatus with a diameter of 2.1 m, placed around the beam line, with a coverage of $|\eta| < 2.5$. Built for the first tracking stage of ATLAS, it is composed of the Pixel Detector, the Semiconductor Tracking and the Transition Radiation Tracker, which technologies and components must be the most radiation hard possible. These are placed in the barrel region and in the end-cap region of ID. The Figure 1.6 shows their configuration, dimension and coverage, while in the Table 1.1 important characteristics are described.

- The Pixel Detector (PD) is the first detector encountered by a particle produced in

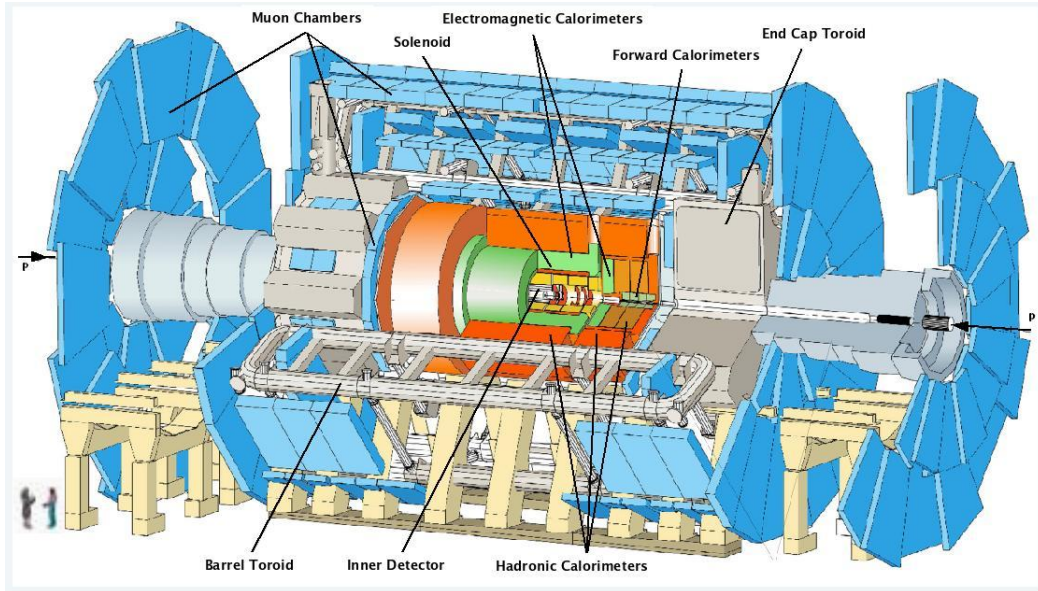


Figure 1.5: *Scheme of the ATLAS layout.*

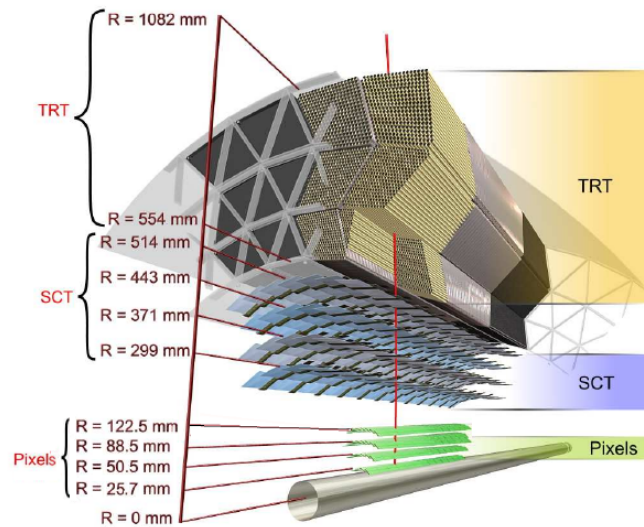


Figure 1.6: *Scheme of the ID*

Detector	Hits Tracks	Element Size	Hits Resolution[μm]
PD, $ \eta < 2.5$			
4 barrel layers	3	50 x 400 μm^2	10(R- ϕ).115(z)
3 x 2 lateral disks	3	50 x 400 μm^2	10(R- ϕ).115(R)
SCT, $ \eta < 2.5$			
4 barrel layers	8	50 μm	17(R- ϕ).580(z)
9 x 2 end-cap disks	8	50 μm	17(R- ϕ).580(R)
TRT, $ \eta < 2.0$			
73 barrel tubes	30	d=4 mm, l=144 mm	130/straw
160 end-cap tubes	30	d=4 mm, l=37 cm	130/straw

Table 1.1: *Characteristics of the detectors of ID.*

the p-p interaction. It is a Silicon based detector which uses the pixel technologies; it has the highest granularity in all ATLAS and it consists of 4 barrels (Insertable B-Layer, B-Layer, Layer1, Layer 2) and 3 disks for each side. It will be described in more details, especially in its off-detector read-out part, in the next chapter;

- The SemiConductor Tracker (SCT) is 4 layers a Silicon microstrip detector. Each layer is formed by modules composed by two microstrip detector bounded together and glued with a 40 mrad angle of their planes, layout used to obtain a better z-measurement. The two microstrip detectors of a single module are glued with the angle between the two microstrips shifted of 90° . In the barrel region the plane of the microstrip detector is parallel to the beam line, while in the end-cap region is perpendicular;
- The Transition Radiation Detector (TRD) is the largest track detector of ID and surrounds the other 2. It consists of a large number, about 5×10^4 , of straw tubes, that are cylindrical tubes with one positive wire in their inside and the internal wall at negative voltage. The straws all together contribute to the measurement of the particle momentum thanks to the high number of hits, and each one is filled with a mixture of Xenon (70 %), CO_2 (27 %), and O_2 (3 %). In the barrel region the tubes are parallel to the beam line, while in the end-cap region are perpendicular.

There are two algorithms used to reconstruct the particle track:

- the inside-out algorithm, in which primary tracks of charged particles born from primary reactions are reconstructed using 3 seeds in the Silicon detectors (SCT and PD) and then the successive hits are added by a combinatorial Kalman-fitter algorithm;

- the outside-in algorithm, where the hits in TRT of secondary charged particles, formed by decays from primary particles or other secondary particles, are used to reconstruct the tracks adding the Silicon hits always with the combinatorial Kalman-fitter algorithm, if there are. Efficiency of track reconstruction is measured by simulated events, and it varies as a function of p_T and η .

1.6.2 Calorimeter

The ATLAS calorimeter system is composed by sampling calorimeters, with the absorber made of Lead, Copper or Iron and the active medium composed by plastic scintillator and liquid Argon. The inner calorimeter is the electromagnetic one, after which there is the hadronic calorimeter. Each calorimeter consists of 4 parts: a barrel part, an extended barrel part, an end-cap part and a forward part. This system includes a pseudorapidity up to $|\eta|=4.9$ and a complete ϕ coverage. The resolution of a sampling calorimeter can be written as

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

where:

- the first term is called the stochastic term and comes from intrinsic fluctuations of the secondary particles of the shower due to their statistical behavior;
- the second term is due to the electronic noise of the read-out channel;
- the third term is a constant that considers temperature, age of detector, radiation damage, and other factors.

The energy resolutions are different based on the calorimeter: $\frac{\sigma(E)}{E} = \frac{10\%}{E} + (1.2 \pm 0.1_{-0.6}^{+0.5})\%$ for the electromagnetic calorimeter in the barrel region, from 0.13 to 0.06 for the hadronic one in the barrel and end-cap region when p_T increase, $\frac{\sigma(E)}{E} = \frac{100\%}{E} + (2.5 \pm 0.4_{-1.5}^{+1.0})\%$ for the forward electromagnetic calorimeter. A scheme of the calorimeter system is shown in Figure 1.7 .

1.6.3 Muon Spectrometer

High p_T muons provide signatures for many physics processes studied in ATLAS. This makes the Muon Spectrometer very important. It is designed for high precision and resolution in the measurement of high p_T muons and it also provides an independent muon trigger. It is divided in barrel and end-cap region. It has barrel and end-cap toroid magnets, and it's composed by 4 different detector technologies, divided in 4 detectors which can be divided in 2 precision tracking detectors and in 2 trigger chambers. We have the Monitored Drift Tube (MDT) chambers which provides the measurement of

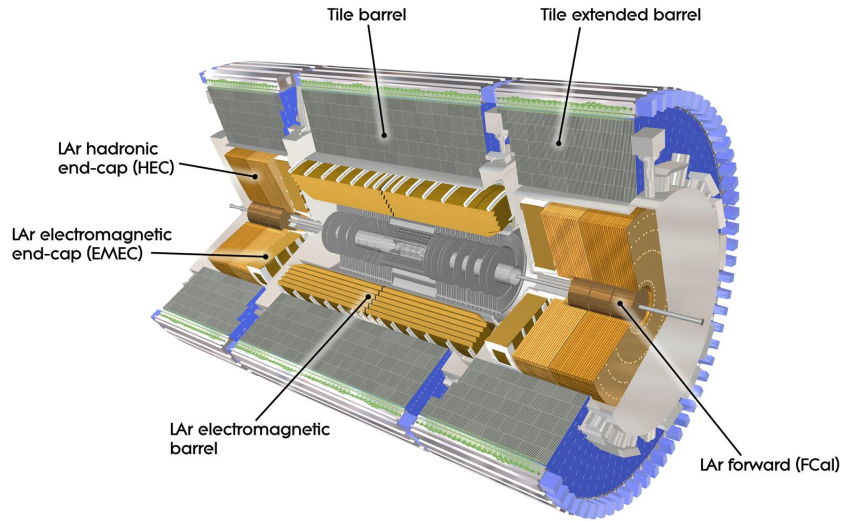


Figure 1.7: *Scheme of the calorimeter system of ATLAS*

momentum and track in the barrel and end-cap regions, the Cathode Strip Chambers that have the same purpose of MDT but are positioned close to the beam pipe in the innermost layer of the end-cap. Then there are 2 trigger detectors: the Resistive Plate Chambers in the barrel region and the Thin Gap Chambers in the end-cap region. This whole system recognizes muons for which $|\eta| < 2.7$, with a threshold of $p_T > 3 \text{ GeV}$, since muons with lower energy lose it completely before the Muon Spectrometer, and with a resolution of p_T measurement of about 20 % at 1 TeV. The muons energy limit for momentum measurement is 3 TeV. In Figure 1.8 is shown the scheme of the Muon Spectrometer.

1.6.4 Magnetic System

ATLAS use a system of superconducting magnets (shown in Figure 1.9) for the measurement of the charged particles momentum. The system is composed by a Central Solenoid (CS) surrounding the Inner Detector, and by a system of 3 large air-core toroids (1 barrel and 2 end-cap) generating the magnet field of the Muon Spectrometer, with a dimension of 26 m in length and 20 m in diameter. The CS, for the momentum measurement of ID, has a magnetic field of 2 T and it points in the positive z-axis direction, while the toroids magnet emits a magnetic field of 3.9 T (barrel) and 4.1 T (end-cap). The entire system work at 4.7 K of temperature. The most important parameters for momentum

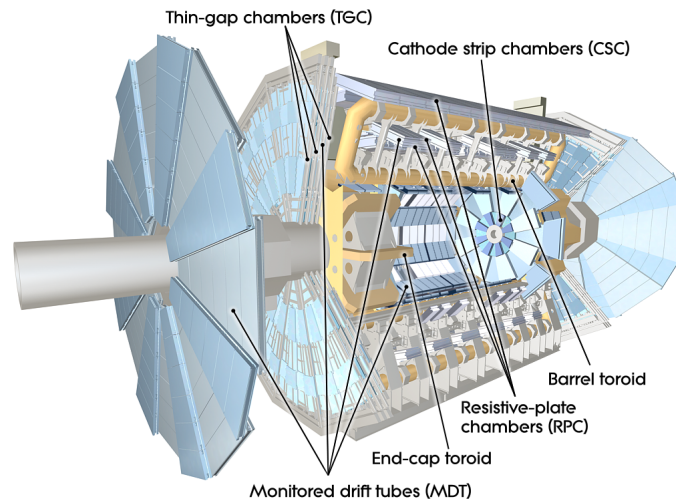


Figure 1.8: *Scheme of the Muon Spectrometer of ATLAS*

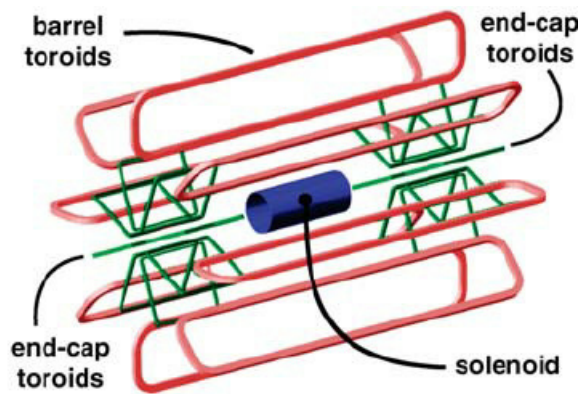


Figure 1.9: *Scheme of the Magnet system of ATLAS*

measurements are the field integrals over the track length inside the tracking volume:

$$I1 = \frac{0.3}{p_T} \int_0^l B \sin(\theta)_{(d \vec{l}, \vec{B})} dl$$

and

$$I2 = \frac{0.3}{p_T} \int_0^{l \sin(\theta)} \int_0^{r/\sin(\theta)} B \sin(\theta)_{(d \vec{l}, \vec{B})} dl dr$$

where I1 is the measurement of bending power field ($p_T=q \times \text{bending power}=q \times (B \times L)$) and I2 represents the total transverse deflection of the particle from its initial path. θ represents the longitudinal component of the angle between the track and the magnet field and the integrals are calculated on the azimuthal direction of the particle ($l=r/\sin \theta$) and on the radial trajectory of it.

1.7 ATLAS Trigger

Thanks to the high luminosity ($L=10^{34} \text{ cm}^{-2}\text{s}^{-1}$), LHC produces in 1 second an order of magnitude of 10^8 proton-proton processes, but the electronics for the read-out can reach only a recording speed of 300 Hz. For this reason ATLAS has a system of multi-level trigger composed by a series of 3 triggers: Level-1, Level-2 and Event-Filter trigger. In Figure 1.10 a scheme of the TDAQ of ATLAS is shown. Level-1 trigger is a hardware

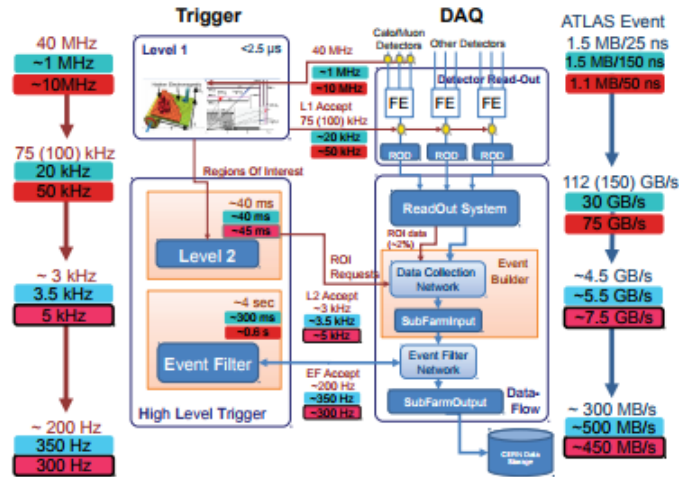


Figure 1.10: *Scheme of the Trigger and Data Acquisition of ATLAS*

based trigger that divides the data read by high p_T leptons, photons, jets and large missing and total transverse energy. It reduced the rate of data down to 50 kHz approximately with a decision time for each collision of $2 \mu\text{s}$ from the collision (where $1 \mu\text{s}$ is

the time due to cables). The data for this trigger arrives from the calorimeters and from the Muon Spectrometer, in particular from the RPC and the TGC chambers. Level-1 trigger defines the regions in η and ϕ coordinates where the other subsequent triggers will start their work, regions called Regions of Interest (ROI); furthermore Lv1 muon trigger searches for coincidences of hits in different trigger stations within a road pointing to the interaction point, because the width of this road is correlated with the transverse momentum. There are 6 muon p_T thresholds governed by the hardware-programmable coincidence logic for this trigger, 3 for the range 6-9 GeV (low p_T) and 3 for the range 9-35 GeV (high p_T). Level-2 trigger is a software based trigger which starts from ROIs defined in Lv1 and uses all the detector informations in these regions in its trigger algorithms. It permits to reach less then 5 kHz in less then 50 ms. Event Filter is the final stage of the trigger chain and it reaches the rate of approximately 300 Hz in less then 4 s. This time isn't due by algorithms, instead it's the time of the standard off-line event reconstruction of ATLAS. There is a trigger menus, where there's a list of characteristics of an event like $E_{t\ miss}$, etc., with a certain threshold (given by the luminosity) for each one. Those events that passed the selection criteria of this menus are tagged and sorted into data streams. In addition with these data streams there are also the streams due to the calibration data of the detector. Lv2 with Event Filter trigger formed High Level Trigger.

Chapter 2

Pixel Detector

Now we will talk more in depth of the Pixel Detector. Its layout is based on 4 layers with a high radiation hardness electronic and high resolution, and it has the important task of tracking the first particles produced by the proton-proton collisions, and measuring the interaction vertexes. The layers are, from the closest to the farthest from the beam pipe: Insertable B-Layer, B-Layer, Layer-1, Layer-2, with a total of 112 staves, each one formed by 13 modules; these modules are tilted respect the z-axis of 1.1° , while the staves are tilted of 20° on the x-y plane. In Figure 2.1 and 2.2 are shown the positioning

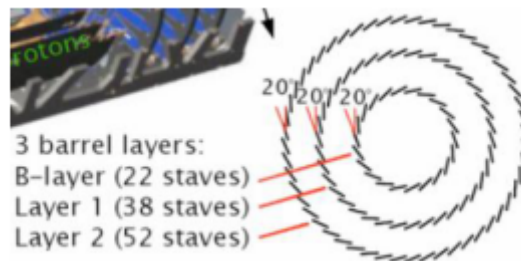


Figure 2.1: *Image of the layout of the Pixel Detector staves*

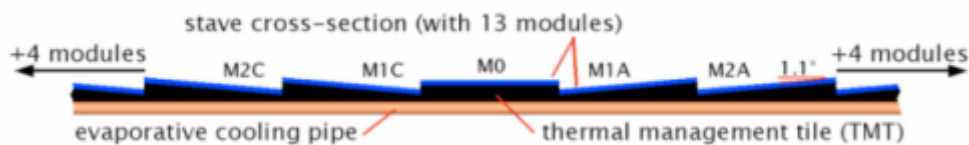


Figure 2.2: *Image of the layout of the Pixel Detector modules*

of the staves and the modules respect the coordinate system, while in Table 2.1 are

Layer	Mean Radius (mm)	Number of Staves	Number of Modules	Number of Channels	Active Area (m ²)
B	50.5	22	286	13178880	0.28
L-1	88.5	38	494	22763520	0.49
L-2	122.5	52	676	31150080	0.67
Total		112	1456	67092480	1.45

Table 2.1: *Characteristics of the Pixel detector.*

written some characteristics of the Pixel Detector’s inner detectors. Here and in the next section we will talk about the 3 layers more outside. About IBL, because of its different technology, we will talk later.

2.1 Modules

The modules that composed the staves are formed by: 16 Front end (FE-I3) sensors chips which are responsible for reading signal from the sensors, a flex-hybrid, a Module Controller Chip (MCC) and a pigtail. In Figure 2.3 a scheme of a module is shown, while in Figure 2.4 is shown a scheme of the connection between the pixel sensors and the read-out chips (FE-I3). An FE-I3 is 195 μm thick and 1.09 by 0.74 cm^2 large and counts 3.5×10^6 transistors in 250 nm CMOS technology. Each one is bump bounded over the sensors and each analog amplifier can discriminate signals of 5000 electrons with a noise threshold of 200 electrons. These signals are then digitalized and buffered inside the End Of Columns (EOC) electronic waiting for the trigger signal. EOC signals are sent to MCC which distributes timing, trigger, reset and calibration signals. The signals of 6 MCCs are converted from electric to optic by the opto-board to the Back Of Crate board, where the opto-board has:

- PIN diode, that converts the signals from optic to electric;
- Digital Optical Receiver Integrated Circuit (DORIC), which adapts the electric signal from LVDS Standard to PIN signals;
- Vertical Cavity Surface Emitting Laser (VCSEL), that converts electrical to optical;
- Virtual Device Context (VDC), that interfaces MCC and VCSEL.

Each module is made of a 256 μm thick crystalline Silicon layer and contains 47232 pixels, where 41984 are 400 by 50 μm and the other are 600 by 50 μm , which are located in the external side to minimize signal loss in the region between the modules.

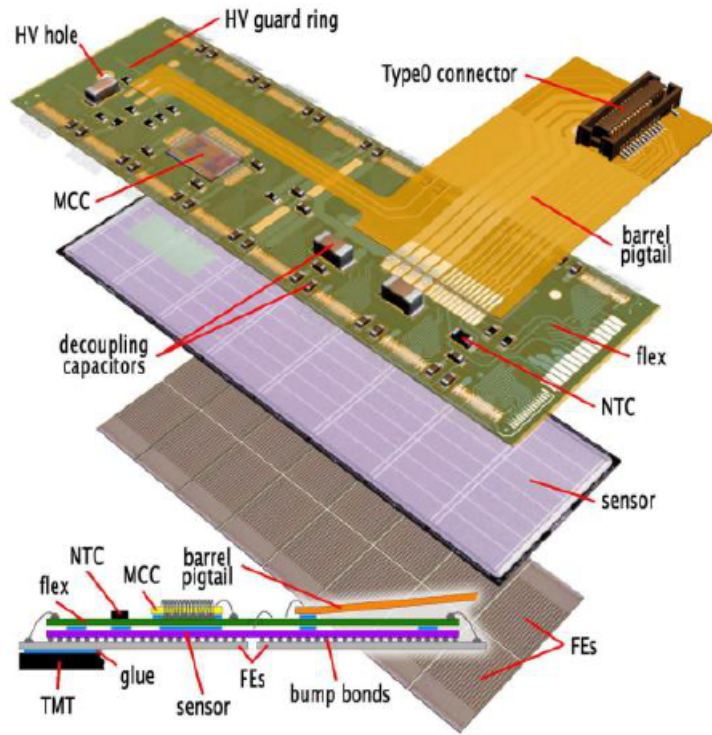


Figure 2.3: Image of the layout of the Pixel Detector modules.

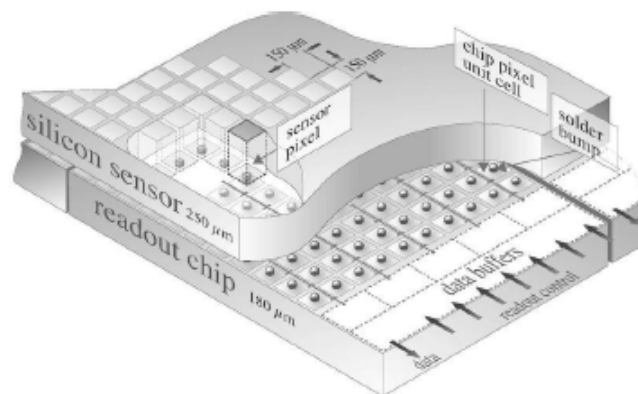


Figure 2.4: Image of the connection between Silicon sensors and FE-I3.

2.2 Sensor

As we said, the Pixel Detector is a semiconductor detector based on Silicon, where a n-doped crystalline semiconductor with p-doped well forms a pixel of the detector. This connection (similar to a diode) is reverse polarized, so the depletion region extends until a ionizing particle pass through the pixel and frees electrons and holes. These drift, for the electric field, toward the metal contacts, which have attracted them, and then they collect these charges. The energy released by the charges is proportional to the collected charges, and, obviously, by the pixel giving a signal, and by the signal itself we can obtain the particle track. The effects of the radiations on the pixels are:

- increment of leakage current, which forces to use a better cooling system;
- the change from n-type to p-type of the substrate, and the consequent shift of the p-n junction to the lower part of the pixel; this problem can be solved enhancing the bias voltage gradually from 150 V to 600 V, fact that decreases the life of the pixel.

To reduce the damages from the radiation it can adding Oxygen atoms to the crystalline structure, keep the Silicon at -20° C because the increase of temperature in the substrate during the data taking is caused by the increase of dopant absorbed by it, creating p-doped dividing zones in the n-doped wells. A scheme of the sensor is shown in Figure 2.5

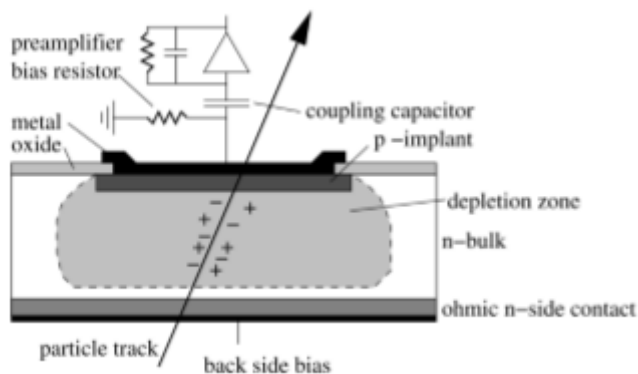


Figure 2.5: *Scheme of the Silicon sensor.*

2.3 IBL

IBL is the first detector of the ATLAS chain and the last that was joined to the Pixel Detector. The motivations are situated in the b-tagging capability decrease of B-Layer due to the luminosity increase of LHC and to the consequent enhance of radiations, and so of damages, in the detector B-Layer, and obviously in the research of a better tracking precision. The IBL technology is different because of the more radiation power and the more surface covered (where this task is performed thanks to the new chip FE-I4).

2.3.1 IBL Sensor and Modules

Sensors and modules of IBL, as we said, use different technologies respect the other layers. There was 2 candidates, shown in Figure 2.6 and 2.7 :

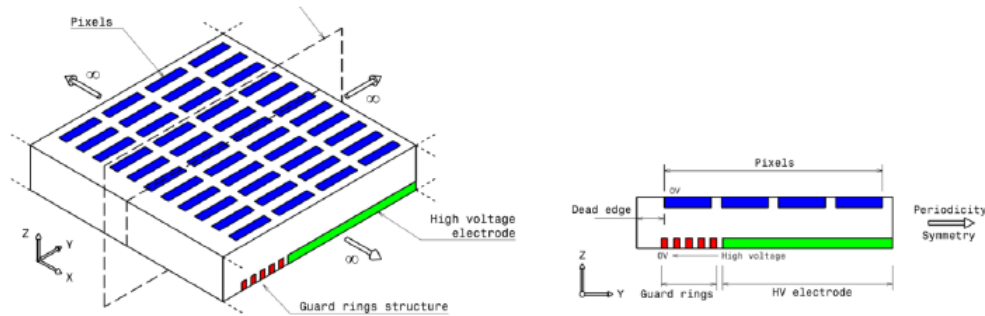


Figure 2.6: *Image of planar sensor in thin border configuration.*

- the Planar Sensors has the layout of a "normal" Silicon sensor. They were used in B-Layer too but in IBL they are slightly different, in fact the inactive border has to pass from 1 mm to $450 \mu\text{m}$ and the NIEL has to double from the old value of $2 \times 10^{15} \text{ neq/cm}^2$ to make acceptable the effects on the signal. Furthermore, from studies made for B-Layer, we now know that a sensor under radiation can double the collected charges if reduces its thickness, because reducing the thickness, consecutively the probability to lost particle's track because of trapping, induced by the radiation damage, reduces too. They request 1000 V for the bias voltage. There are 3 possible configurations: conservative, thin border, thin p-doped substrate.
- 3-D sensors have a completely different geometry from the planar ones; they read the signal from the charges collected from 2 electrodes at once due to the low number of charges, and for this double reading the noise increases. The etching

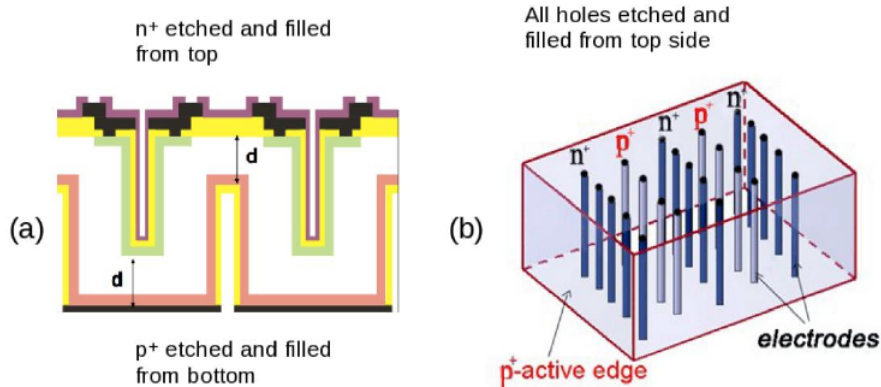


Figure 2.7: *Scheme of 3-D sensor, in double sided (a) and full-3D (b) configuration.*

during the productive process is the difference between full 3-D sensor and double-sided ones. Furthermore the active area of the full 3-D extends much more to the surface reducing the not sensible volume, and these sensors are more closer each other in the plane, reducing the voltage bias (150 V) and so the leakage current. Efficiency decreases of 3.3 % if a particle passes near the electrode, but this only in the case the sensor is in perpendicular position to the particle, that isn't our case because the sensors are tilted of 20°.

2.3.2 FE-I4

The LHC luminosity and the small distance of IBL from the beam pipe has been driven to upgrade the chip for the detection of the particles from FE-I3, that isn't good enough to maintain the request efficiency, to FE-I4. This chip, designed for IBL detector, is built with a 130 nm CMOS technology, a thinning down of the gate oxide for an enhance in the radiation tolerance, and in a 8 metal option with 2 thick Aluminium top layers for better power routing. Furthermore, the current drain architecture of FE-I3 scales badly with high hit rates and increased front-end area; for these reasons the FE-I4 pixel dimensions are $50 \times 250 \mu\text{m}^2$, with an increase of z-axis track resolution and a reduction of the pixel cross-section. The active area of FE-I4 is close to 90 %, given by the active size of 20 mm (z-direction) by 16.8 mm (ϕ -direction) and with 2 mm more foreseen for the periphery, layout containing an array of 80 by 336 pixels each one with an analog and a digital component. The biggest size of FE-I4 takes to important benefits like the reduction of material for the detector layer and the enhance of the physics performance like the b-tagging efficiency vs. the light quark rejection factor, and furthermore obviously the

reduction of costs. The layout of the FE-I4 is shown in Figure 2.8 .

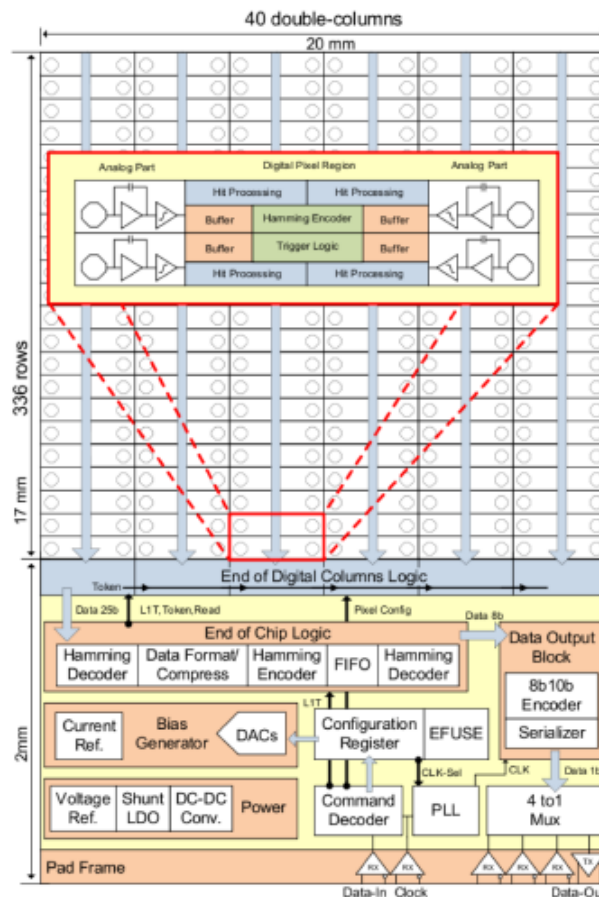


Figure 2.8: *Scheme of FE-I4 read-out chip.*

Digital Section

The FE-I4 uses a different method for the storage of pixel hits, completely different from the column drain architecture. It uses a local storage of the pixel hits in buffer located at pixel level, in a region of 2 by 2 pixels, where the 4 pixels are tied together at digital logic point of view and they shared the same digital processing (see Figure 2.9). This architecture takes advantage from the small feature size of the 130 nm CMOS technology, reducing the loss in hits recording below to 0.6 % at luminosity 3 times more than the actual, saving area and reducing power. Furthermore now the recording of small number of electrons, where this recording is most of the time located near a large signal recording by the pixels, happens without being time-stamped, giving an handle

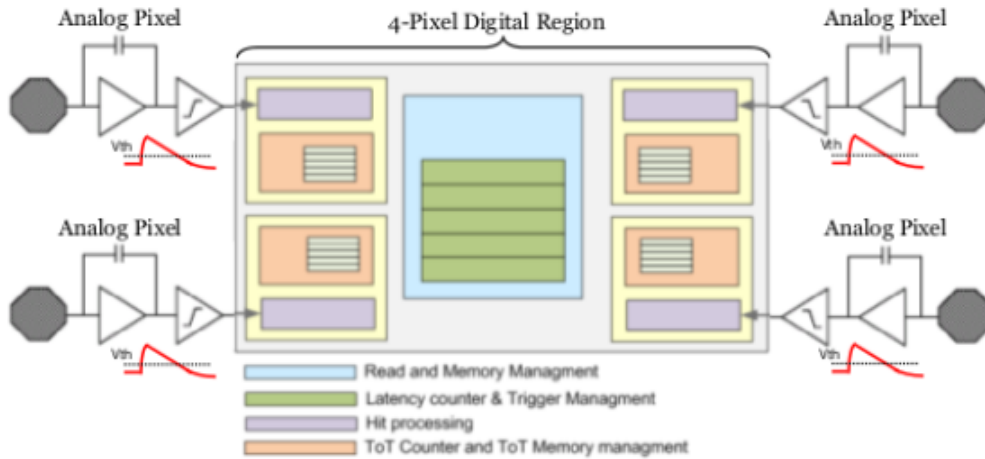


Figure 2.9: *Scheme of FE-I4 Double Column system.*

on time-walk. A Double-Column contains the 4 pixel region and its logic, with a clock of 40 MHz (the bunch-crossing of LHC). Here 4 adjacent pixels share latency counter, trigger, read and memory management units. The 8-bit latency counter counts down a programmable latency. The individual components are the Time Over Threshold 4-bit counters and the hits processing circuitry. A start due to a particular event, coming from any discriminator, starts the latency counter, which is only one even for several signals from several pixels in the same bunch-crossing. The time which the pixel comparators stay above threshold tells us the dimension of the hits. The read-out is based on a dual token between Double-Column and End of Column which make triple redundant with majority voting to yield enhancement. A pixel configuration shift register runs in each Double Column to tune each pixel singly, and redundant always to yield enhancement. The End of Column logic is kept very simple and serves only as a dedicated interface between each one of the 40 Double Column and the digital control block with its FIFO.

Analog Section

The analog section of the chip, shown in Figure 2.10 , which cover 50 by 150 μm^2 of the size of FE-I4, is implemented as a 2-stage architecture, optimized for low power, low noise and fast rise time, followed by a discriminator. The first stage is an adjusted cascode pre-amplifier with a triple-well NMOS input, containing an active slow differential pair, tying the pre-amplifier input to its output, and used to compensate sensor radiation-related leakage current. The section has a 100 nA DC leakage current tolerance. The second stage is AC coupled to the pre-amplifier and implemented as a PMOS input

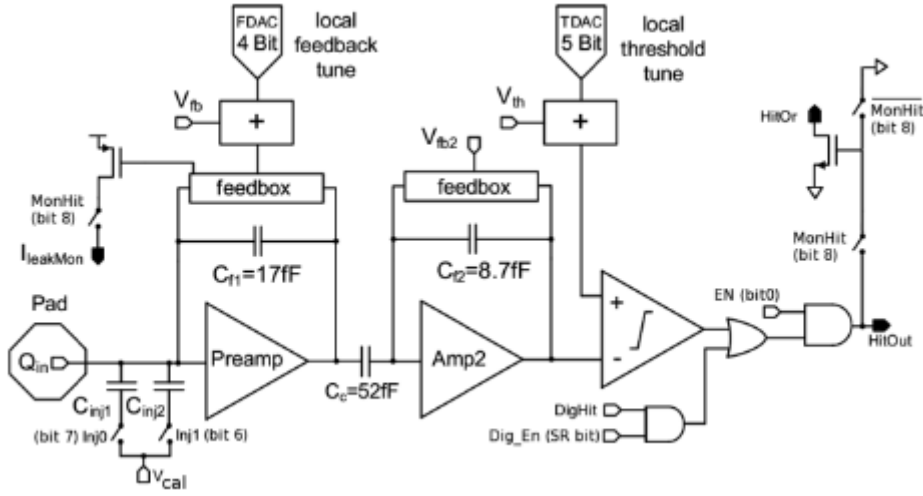


Figure 2.10: *Scheme of FE-I4 Double Column system.*

folded cascode; this coupling comports two benefits: the decouple of the second stage by the leakage current related to DC potential shift and gives an additional gain factor, coupling capacitance to feedback capacitance, always in the second stage, of 6. This permits to increase the feedback capacitance of the first stage without degrading the signal amplitude at the discriminator input, and giving benefits like the enhance of charges collection efficiency, signal rise time and power consume.

Chip Periphery

The FE-I4 periphery has the sequent tasks: communication and operational configuration of the integrated circuit (IC), organization of the data read back and fast data output serialization. New blocks are implemented for future prototype functions and to provide extra testing capabilities like redundant memories, low-speed multipurpose multiplexer. Two LVDS inputs are required to communicate with the FE-I4: the clock at 40 MHz and command input Data-In (40 Mb/s). The command decoder of FE-I4 is based on the architecture for the module control chip. Its stream is decoded into local pixel configuration and global configuration, trigger commands. The local registers for pixel configuration are 13 bits deep, while for global configuration the 32 registers are 16 bits deep. In the bias generator section, based on an internal current reference, DACs convert the stored configuration values to voltages and currents needed to tune all the IC sections. The decoded trigger is sent to the pixels and to the End of Chip Logic block where the readout is initiated. The data stored in the 4-PDR ToT buffers are sent to the periphery only when a trigger signal that confirms an hit is sent, then data

are associated to the specific bunch-crossing corresponding to the specific trigger. The trigger confirms an hit when the coincidence of a trigger with latency counter reaches its latency value in 4-PDR. At this point the 4-PDR address and the 4 ToTs are propagated to the End of Chip Logic, where the transmitted signals are Hamming coded to yield enhancement. Then the data are re-formatted and stored in a FIFO to be sent out. The re-formatting is due to band-width reduction and facilitate the following data processing steps. Further the data, the pixels send even information describing its status for the diagnostic from the global registers. The data are then encoded in an 8b/10b custom protocol, in Data Output Block, and serialized at 160 Mb/s, a fast serialization made possible by the use of a speed clock formed by a Phase Locked Loop clock generator.

2.4 BOC-ROD System

After the detection of the particles by the semiconductor detector and the acquisition of the relative signals from the on-detector read-out (the read-out system which is located in the area subjected to radiation damages), these signals are sent to the off-detector part of the readout system of IBL, shown in Figure 2.11 . This infrastructure is composed by:

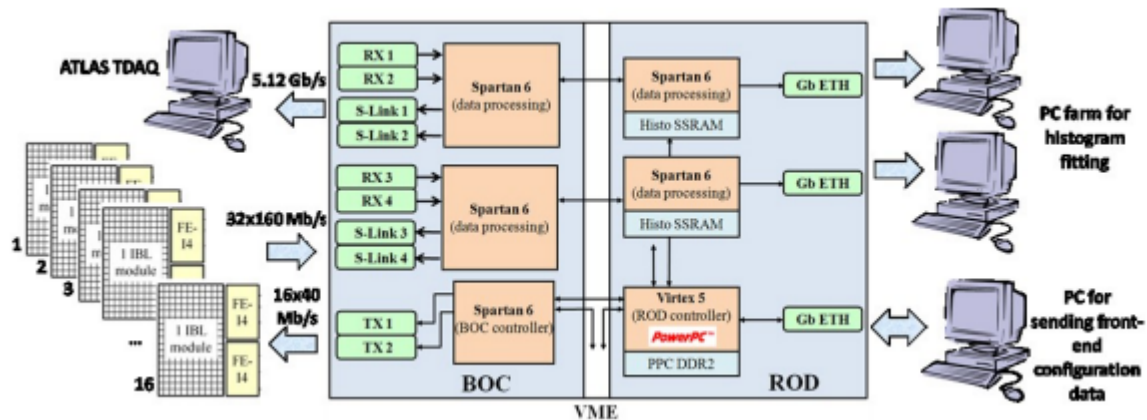


Figure 2.11: *Scheme of BOC-ROD system.*

- Back of Crate (BOC) board;
- ReadOut Driver (ROD) board;
- VME Crate;
- TTC Interface Module (TIM) board;

- Single Board Computer (SBC);
- S-Link to send data from the BOC to the ATLAS TDAQ system;
- optical module to connect the BOC with the Front-End chip (FE-I4);
- Ethernet connections at speed of 1.12 Gb/s to send histograms for calibration analysis and configuration data for FE-I4.

Starting from the received data from 32 FE-I4 (bandwidth of 5.12 Gb/s), these data are sent to the BOC by optical modules, which consequently send it to the ROD encoded with an 8b/10b code, where these data are processed. After that the data are sent to two different ways: to the BOC again and then by 4 S-Link to ATLAS TDAQ system (5.12 Gb/s connection), and to the PC farm for the calibration histograms. Furthermore, thanks to a GB Ethernet connection, it's possible to send data from the BOC to 16 FE-I4 to configure them, where originally the configuration data were produced in the ROD and then sent to the BOC. Each one of these off-detector systems can interface 32 FE-I4 in input and 16 in output.

2.5 BOC-ROD Comunication

The BOC-ROD interface carries all data which has been received from the detector. There are 96 lines with SSTL3 (Sub Series Terminated Logic) I/O standard between the cards. The 96 lines are divided into 8 12-bits wide data busses. Each data bus transfers the data of 4 front end chips at a rate of 80 MHz. Data lines carry the decoded 8b/10b data of the channels and the control lines show if the data is a 8b/10b-k-word.

2.6 IBL-BOC

The IBL-BOC functions are to receive data from the front end, to send configuration data to it and provide the clock for the on-detector part. This clock is generated in the TIM board and it's sent to the BOC where it can be delayed, subsequently the PLL generates copies of this clock to send them where is necessary. The management of the clock is handled by the BOC FPGAs, which are 1 BOC Control FPGA (BCF) and 2 BOC Main FPGAs (BMF). A photo of the IBL-BOC is shown in Figure 2.12

2.6.1 BCF

BCF (a Spartan 6 FPGA) provides at the control of BOC. The central component of the firmware is a Wishbone interconnect which gives to all the peripheral the basis of the configuration for the access. To this interconnect are connected a Setup-Bus connector

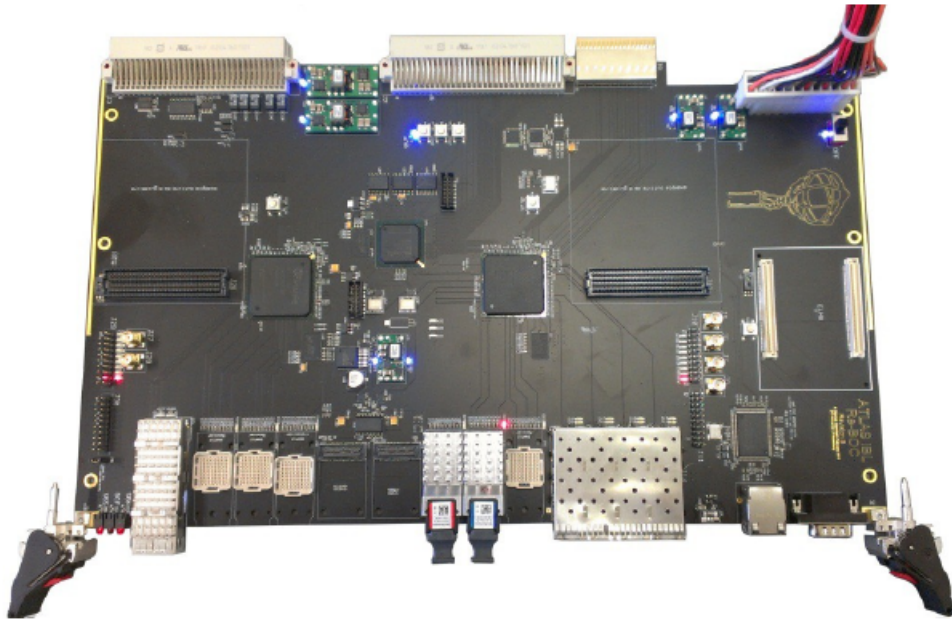


Figure 2.12: *IBL-BOC*.

and a Microblaze processor. The latter configures the ethernet access to the BOC and provides some control tests for the board. Setup-Bus (shown in Figure 2.13) is an asynchronous configuration interface between BOC and ROD with 16 addresses, 8 data and 3 control lines. The FPGA's configuration follows the sequent steps: BCF loads its configuration from a 64 Mbit SPI Flash in Master Serial Peripheral Interface mode; then BCF reads the configuration settings of BMFs from another SPI Flash and downloads it via the Slave Serial configuration ports; in the end BCF loads the software, which depends to the last configuration, from a third SPI Flash.

2.6.2 BMFs

BMFs (2 Spartan 6 FPGAs) encode the configuration data from ROD into a 40 Mb/s serial stream and then send it to the front-end. The TX path is used to send commands and trigger to the modules. In normal detector operation it is used to do the Bi-Phase Mark (BPM) encoding (shown in Figure 2.14) of the incoming data from ROD, to adjust the detector timing using coarse and to delay blocks. The coarse delay has implemented a variable-tap shift register clocked at 160 MHz. The RX path in the firmware is responsible for the reception and decoding of the incoming detector data, after that the decoded data are collected and multiplexed to ROD.

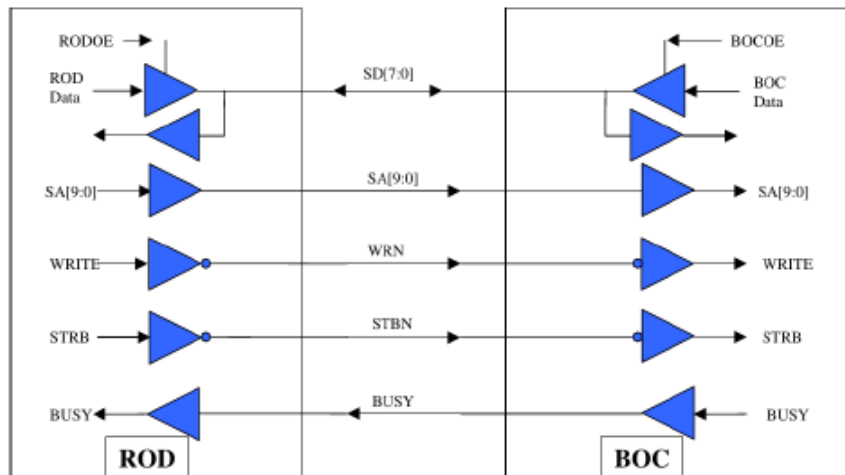


Figure 2.13: *Scheme of the Setup Bus.*

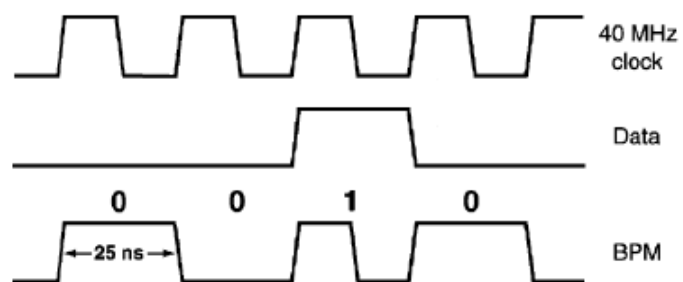


Figure 2.14: *Example of the BPM encoding.*

2.7 IBL-ROD

The second board, which manages the off-detector readout, is the IBL-ROD, that is the upgrade of the ATLAS Silicon Read Out Driver (SROD), used in the SCT and Pixel B-Layer, L1 and L2. Project and firmware of the ROD are mainly developed in Bologna, and it provides the data gathering and subsequently building of the event fragments, and the calibration histograms building. ROD has: 1 Digital Signal Processor (MDSP, currently not used), 1 Program Reset Manager (PRM) FPGA, 1 ROD Controller FPGA, 2 FPGAs slave, 1 Phase-Locked Loop (PLL), 32 MByte SDRAM DDR, 4 Mbit and 64 Mbit FLASH memories, 2 GByte DDR2 SODIMM, 3 Gbit Ethernet connections. A photo of the IBL-ROD is shown in Figure 2.15 .

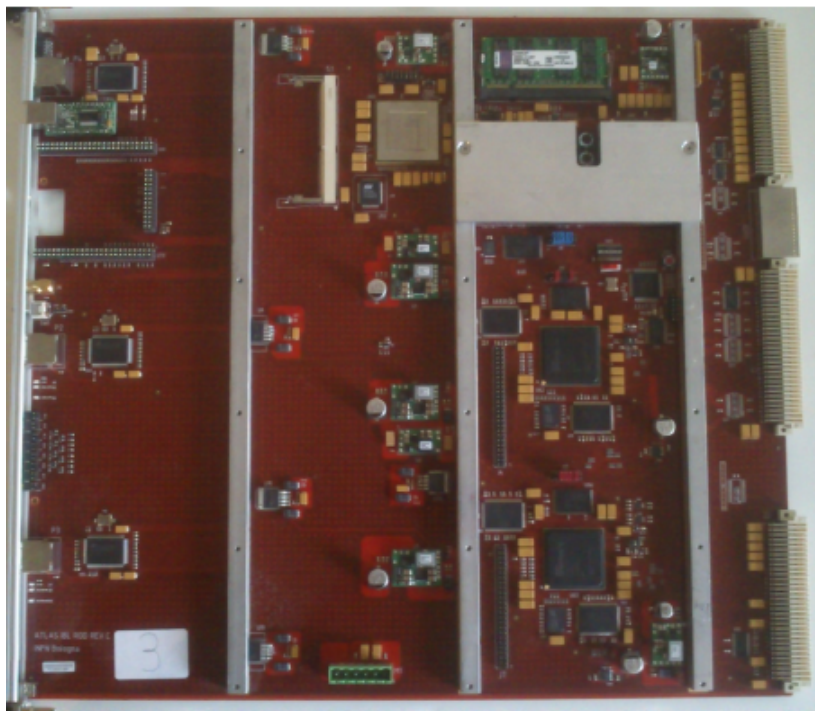


Figure 2.15: *IBL-ROD*.

2.7.1 ROD Controller

The control FPGA in the ROD is made by a Virtex 5 which has the role of Master of the Read Out Driver, FPGA which manages with: the FE-I4s, the triggers that comes from the TTC Module and all the information that refer to the trigger itself. Embedded in the FPGA there's a Power PC (PPC) microprocessor. The Master blocks are:

- Event ID and Trigger Processor, which process the event data and then sends them to the FPGAs Slave, and tells to the FE Command Processor to generate and send the configuration data to the FE. It can be driven by the PPC or by the TIM;
- FE Command Processor, which generates and sends commands to the FE if required, and generates Lv1 trigger if TIM issues a trigger;
- Event Processor, where Event ID, Trigger Type and Bunch Crossing ID are sent to the Event Fragment Builder of the Spartan Slaves.

A logic block of the Master is shown in Figure 2.16 .

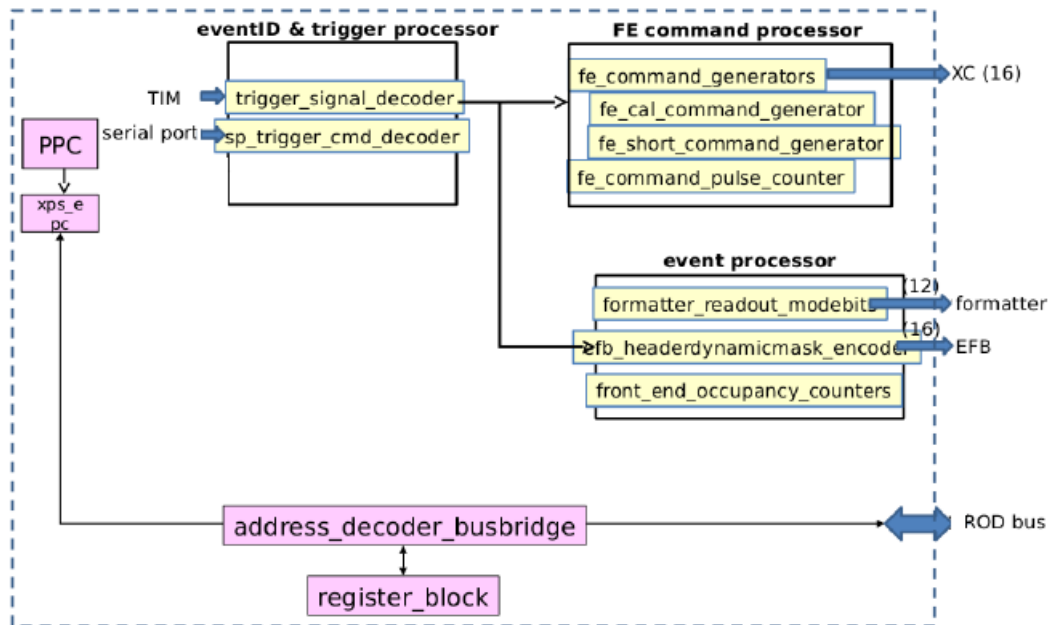


Figure 2.16: Scheme of the ROD Master firmware block.

2.7.2 Spartan Slave

The slave FPGAs are Spartan 6 with a Microblaze processor emulated on them. These FPGAs: process the data took from the FE-I4, collect histograms to a SSRAM and send them to an histogram server, if needed, through a Gb Ethernet connection programmed always by the Spartan 6. The firmware logic blocks of the Slaves are shown in Figure 2.17 and are:

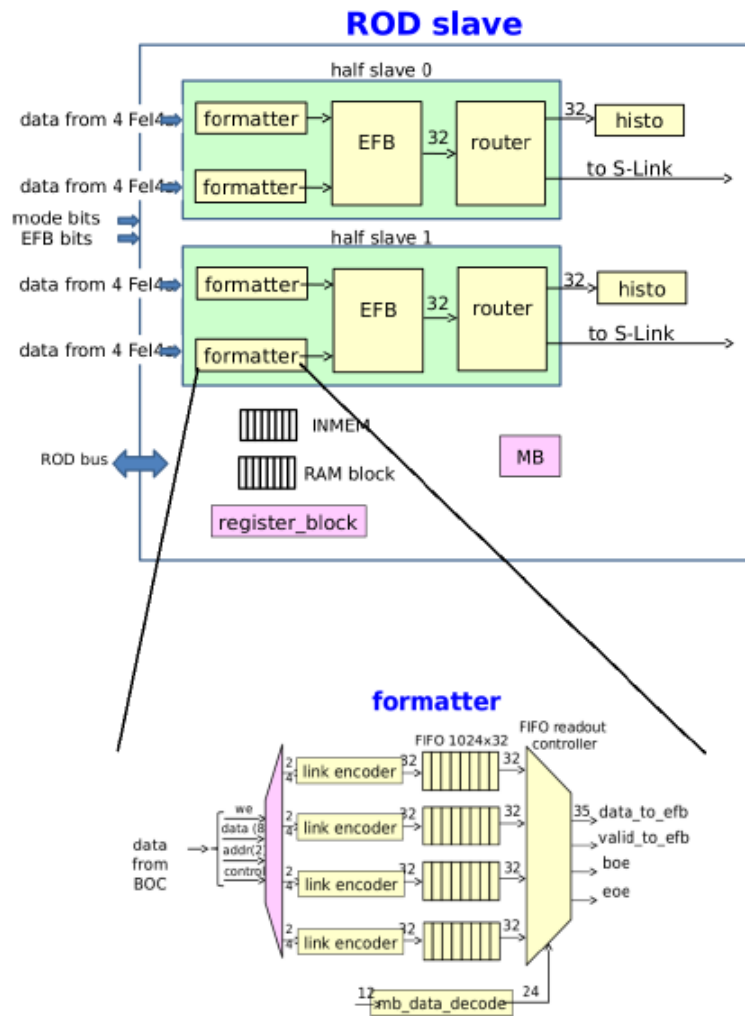


Figure 2.17: Scheme of the ROD Slave firmware block.

- Dual Clock FIFO, which connect the 80 MHz clock of the bus and the 40 MHz clock of the FPGA;
- Event Fragment Builder, where the data coming from the front end are added with header and trailer infos as trigger type, event ID or bunch crossing ID;
- Inmem FIFO (accessible by the PPC), which has the task to debug by collects all inputs from the BOC-ROD buses and verified the data, even before entering the gatherer zone, and giving a perspective of what should happen;
- Histogrammer, where the histograms are collected, histograms which contain information about calibration runs.

2.7.3 PRM

The PRM (a Spartan 6 FPGA) interfaces with the VME bus, the ROD Controller FPGA, the slave FPGAs and the PLL. It has the important role of programming and resetting the ROD FPGAs.

2.7.4 SBC

Single Board Computer is a computer mounted on a 6U board which: programs some ROD components, controls all the operations of the VME on the ROD and monitors the ROD's components temperature.

2.7.5 Lattice PLL

Lattice ispClock 5620 Phase-Locked Loop (PLL) is a control system which generates clocks based on clock connected in input to it, comparing the phase and frequency of the clocks in input and output and varying them until the clocks match. It is composed by the following blocks:

- Phase Detector (PD), which generates the voltage representing the difference in phase between the clocks, and sends this voltage to the Voltage-Controlled Oscillator (VCO) which maintains the difference constant;
- Loop Filter, which has two functions: decides how the loop must behave in case of disturb, limits the reference frequency energy output by the PD so that it can reduce the spurs that can be produced;
- VCO which is an LC oscillator.

2.7.6 S-Link

A Simple LINK (S-Link) is a link that can be thought of as a virtual ribbon cable, moving data or control words from one point to another. It is shown in Figure 2.18 .

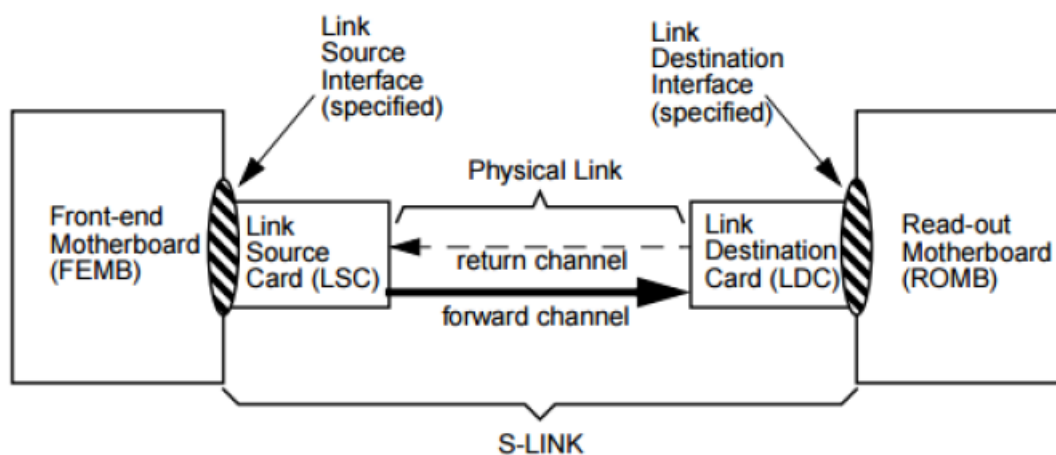


Figure 2.18: *Scheme of the S-Link.*

The specification describes the interface between the Front-end Motherboard (FEMB) and the Link Source Card (LSC), and the interface between the Link Destination Card (LDC) and the Read-out Motherboard (ROMB). It does not describe the physical link itself. Further the simple data moving, S-Link includes:

- Control/Data bit, where all the words transmitted are accompanied by an additional bit which enables the user to mark any word and thus identify it;
- Error Reporting, using an LDERR line, a S-link detects transmission errors and reports them. Furthermore data error LED is illuminated and held until reset;
- Test Function, where LSC and LDC are transformed in a transmitter and a receiver of fixed pattern which are verified by LDC, mode called test mode. If data error are revealed, the data error LED is illuminated by LDC; furthermore LDC can transfers test pattern to the ROMB;
- Reset Function, which provides an hard reset for LSC and LDC.

2.7.7 TIM

The Trigger Timing Controller Interface Module interfaces the ATLAS Lv1 trigger with the Read-Out Drivers of the Pixel Detector using the LC-standard TTC and Busy system. It makes the following tasks:

- propagates the clock of TTC all over the experiment;
- receives and upgrades the triggers;
- keeps updated the TTC with Bunch and Event Counters via Bunch Counter Reset and Event Counter Reset;

TIM has 2 FPGAs where one has generic functions for the TIM (VME Interface, local bus control, board reset, sending of status information to the second FPGA), while the other provides interface with the FE panel and ROD backplane signals. A photo of the TTC board is shown in Figure 2.19 .

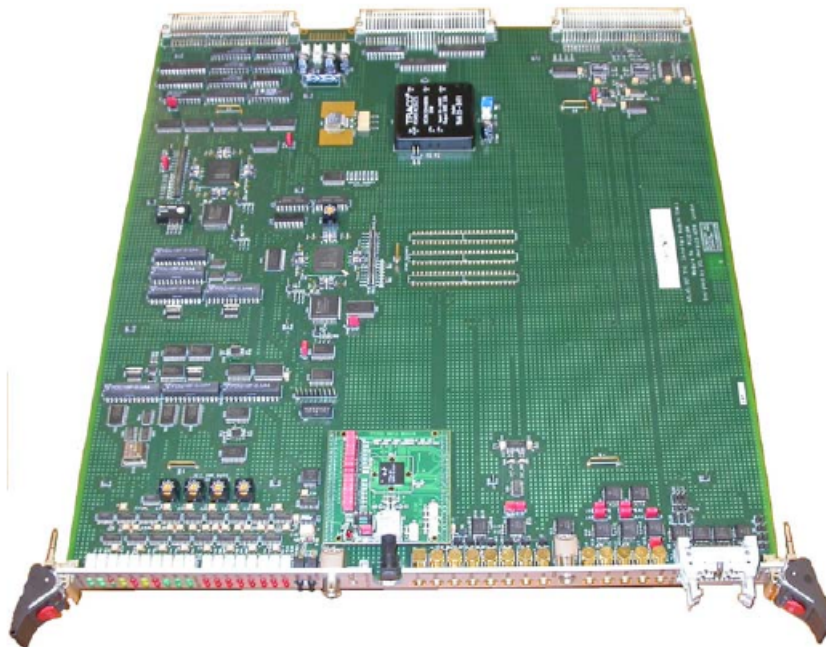


Figure 2.19: *TIM board.*

Chapter 3

Pixel-ROD

The read out system just described did and is doing diligently its work of reading data from the front end, and sending them to the ATLAS TDAQ and PC farm. It was built and configured for the current LHC luminosity of $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ and the current bunch crossing frequency of 40 MHz. It permitted, with all the ATLAS experiment, to reach many important achievements, but now the game rules will change. The future LHC upgrade (after LS2 and LS3) will enhance these numbers significantly, as the Table 3.1 show. In this future situation the BOC-ROD read out system will must be upgraded to

		LHC	HL-LHC
bunch spacing	ns	25	25
peak L	$10^{34} \text{ cm}^{-2}\text{s}^{-1}$	2.3	15.5
protons/bunch	10^{11}	1.7	1.7
peak of events/bunch		44	294

Figure 3.1: *Future LHC features, with High Luminosity LHC numbers.*

provide the efficiency need for the future CERN physics. There are some projects for new read out system working and concurring for the tasks required by the future physics at CERN. The most important technology that the next read out system will must develop are the PCIe connections (because this is the direction in which the read out projects of ATLAS and even CMS are going), the transceiver connections, the capacity to process and transport a bigger amount of data respect at which it was never be produced. Searching for these characteristics, the boards choose to achieve the new physics goal are the KC705 and the ZC702, 2 Xilinx evaluation boards. The project idea of the INFN

laboratory and of the Department of Physics and Astronomy laboratory of the Bologna University was to use the knowledge from the experience achieved with the IBL-ROD and use it in a new board born from the fusion of 2 boards with different characteristics, that united in 1 will substitute the BOC and the ROD and will increase the performance. The name choose for this board is Pixel-ROD (PR), showed in Figure 3.2 .

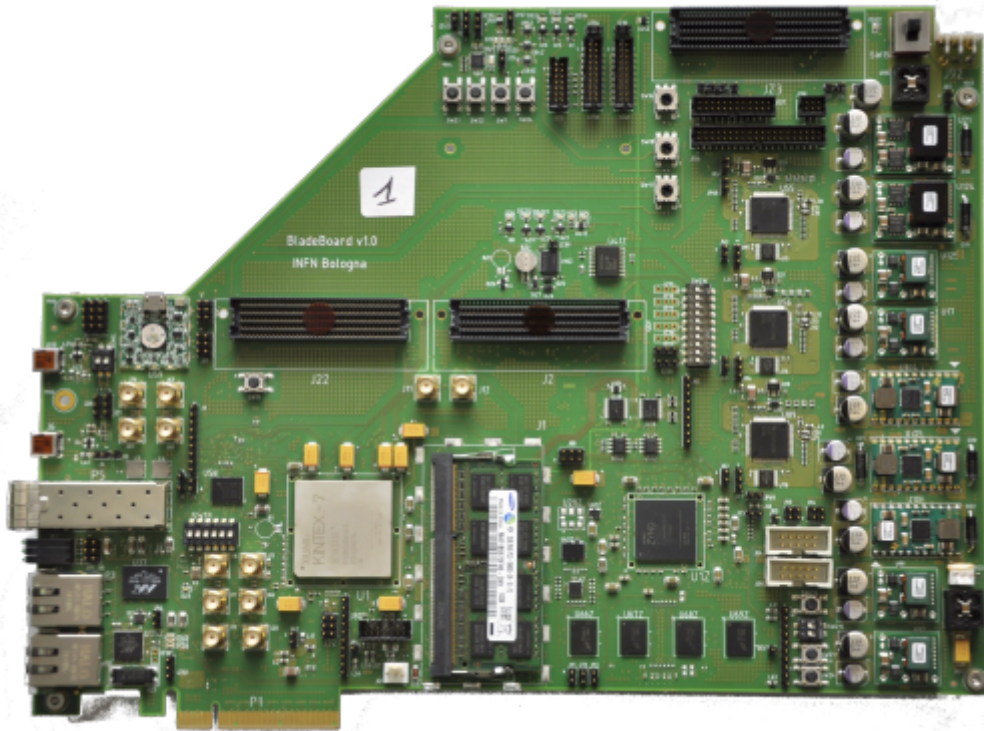


Figure 3.2: *The Pixel-ROD board.*

- KC705 is an evaluation board of the Xilinx with a programmable FPGA and with a high connection capability even with high speed (up to 12.5 Gb/s). Recalling what said in the previous section, the KC705 has a PCIe Gen2 8x connector with a nominal transmission speed of 4 GB/s (2.5 GB/s during the tests), which permits to connect the board directly to a pc (of the ATLAS TDAQ or of the pc farm) and to leave the old VME bus with its 160 MB/s of speed. The high speed connections are the 16, opto-electrical or electrical, input and output connectors which can reach a transmission speed of 12.5 Gb/s each one. In the end the Kintex-7 FPGA is a powerful instrument which can be used to upgrade the slave work of the readout system.

- ZC702 is, like the KC705, a Xilinx board which is the choose Master for this readout system project. Its most important feature is the ARM-Cortex A9 MPCore processor embedded in the Zynq-7000 FPGA which surpasses in brutal strength the PowerPC 440 processor embedded in the Virtex 5 FPGA, indeed we have 667 MHz frequency for PPC against 1 GHz of the ARM, 256 kB level 2 cache against 512 kB, same level 1 cache (32 kB), etc.

3.1 Pixel-ROD

In this section we will talk about the Pixel-ROD specifications, Pixel-ROD defined as a read out PCIe based board with Master-Slave system.

3.1.1 FPGAs

The PR is provided with 2 FPGAs, a Zynq-7000 XC7Z020-1CLG484C and a Kintex-7 XC7K325T-2FFG900C, which now will be described the major features.

Kintex-7

The FPGA Kintex-7 represents the multi-high speed connection controller with the outside of the board. XC7K325T-2FFG900C has the following characteristics:

- advanced high-performance FPGA logic based on real 6-input logic Look-up table (LUT) technology configurable as distributed memory;
- high performance SelectIO technology with support for DDR3 interface up to 1866 Mb/s;
- high speed serial connectivity with build-in multi gigabit transceivers from 600 Mb/s to maximum 12.5 Gb/s;
- a user configurable analog interface (XADC), incorporating dual 12-bit 1MSPS analog-to-digital converters with on-chip thermal and supply sensors;
- DSP slices with 25 x 18 multiplier, 48-bits accumulator, and pre-adder for high-performance filtering, including optimized symmetric coefficient filtering;
- powerful Clock Management Tiles (CMT), combining Phase-Locked Loop (PLL) and Mixed-Mode Clock Manager (MMCM) blocks for high precision and low jitter;
- integrated block for PCI Express® (PCIe), x8 Gen2 Endpoint and Root Port designs;
- 500 I/O pins and 34 Mb Block RAM blocks.

Zynq-7000

The FPGA Zynq-7000 represents the general controller of the PR, the Master of the Kintex-7 to which the configuration board commands move from the users, and from them move toward the Slave. The XC7Z020-1CLG484C features for the Programmable Logic (PL) part are the following:

- 6-input logic Look-Up Table (LUT) technology configurable as distributed memory;
- DSP slices with 25 x 18 multiplier, 48-bits accumulator, and pre-adder for high-performance filtering, including optimized symmetric coefficient filtering, optional pipelining, optional ALU;
- 4.9 Mb Block RAM (140 36 kB blocks);
- a user configurable analog interface (XADC), incorporating dual 12-bits 1MSPS analog-to-digital converters with on-chip thermal and supply sensors.

The major characteristic provides by XC7Z020-1CLG484C is the embedded ARM Cortex A-9 MPCore processor. It's the Processor System (PS) part of the Zynq-7000, and toward are described the most important components.

Application Processor Unit (APU)

The APU contains 2 processors which share, with a NEON co-processor, a 512 kB Lv 2 cache (for instruction and data), while each processor implements a 32 kB Lv 1 cache. The architecture is the ARM v7-A, which supports ARM, Thumb (for which the single instruction multiple data (SIMD) instructions are available) and Java instructions, with full virtual memory support and with the instructions added by the NEON co-processor. The configuration is a MP with a Snoop Control Unit (SCU) which maintains coherency between Lv 1 and Lv 2 caches and manages the Accelerator Coherency Port (ACP) interface from the PL. In parallel with the Lv 2 cache there's a 256 kB On-chip Memory Module (OCM) that provides a low-latency memory. ACP facilitates communication between PL and APU. All accesses through the Lv 2 cache controller can be routed to the DDR controller or can be sent to other slaves in the PS or PL depending on their address. To reduce latency to the DDR memory, there is a dedicated port from the Lv 2 cache controller to the DDR controller. The architecture supports TrustZone Technology to help to create a secure environment to run applications and protect their contents. The Memory Management Unit (MMU) works to memory protection, controlling access to and from external memory and translation virtual memory (addresses) to physical addresses.

Interconnect

The interconnect located within the PS comprises multiple switches to connect system resources using AXI point-to-point channels to communicate addresses, data, and response transactions between master and slave clients. This ARM AMBA 3.0 interconnect implements a full array of the interconnect communications capabilities and overlays for Quality of Services (QoS), debug and test monitoring. QoS resolves contention in the central, master, slave and memory interconnect.

DDR Memory Controller

The DDR memory controller supports DDR2, DDR3, DDR3L and LPDDR2 devices, and consists of three major blocks: an AXI memory port interface (DDRI), a core controller with transaction scheduler (DDRC) and a controller with digital PHY (DDRP). The DDRI block interfaces with 4 64 bits synchronous AXI interfaces to serve multiple AXI masters simultaneously. Each AXI interface has its own dedicated transaction FIFO. The DDRC contains two 32-entry content addressable memories (CAMs) to perform DDR data service scheduling to maximize DDR memory efficiency. It also contains fly-by channel for low latency channel to allow access to DDR memory without going through the CAM.

I/O Peripheral

The PS I/O peripherals, including the static/flash memory interface, share a multiplexed I/O (MIO) of up to 54 MIO pins. Zynq-7000 AP SoC devices also include the capability to use the I/Os which are part of the PL domain for many of the PS I/O peripherals. This is done through an extended multiplexed I/O interface (EMIO). Software programs the routing of the I/O signals to the MIO pins. The I/O peripheral signals can also be routed to the PL (including PL device pins) through the EMIO interface. This is useful to gain access to more device pins (PL pins) and to allow an I/O peripheral controller to interface to user logic in the PL. The I/O multiplexing of the I/O controller signals differ; indeed, some IOP signals are solely available on the MIO pin interface, some signals are available via MIO or EMIO, and some of the interface signals are only accessible via EMIO.

3.1.2 Memory

The physical memory installed on the Pixel-ROD are:

- 2 GB DDR3 RAM SODIMM (Kintex-7);
- 1 GB DDR3 RAM (Zynq-7000);

- 2 128 Mb Quad SPI flash (1 for Kintex-7 and 1 for Zynq-7000);
- 1 128 MB parallel flash memory (Kintex-7).

3.1.3 Internal Bus

The Pixel-ROD has 3 types of internal buses that connect directly the 2 FPGAs:

- a 21 bit differential bus;
- a 5 bit single line bus;
- a 1 bit differential bus dedicated to share the differential internal clock of the board between the FPGAs.

3.1.4 I/O

Some input and output from the original schematic of KC705 and the ZC702 have been removed for a more useful purpose. These removed components are: the HDMI port (KC705 and ZC702), SD card reader (KC705 and ZC702), some GPIOs and LEDs (KC705 and ZC702), 1 of the 2 FMC LPC connector (ZC702), the USB port (ZC702), the PMODS connection (ZC702). The connection from the board and to the board are:

- PCI Express Gen2 x8 (Kintex-7);
- 2 10/100/1000 Ethernet tri-speed (Kintex-7 and Zynq-7000);
- 1 FMC HPC VITA 57.1 (Kintex-7);
- 2 FMC LPC VITA 57.1 (Kintex-7 and Zynq-7000);
- 2 USB-to-UART (Kintex-7 and Zynq-7000);
- 1 USB JTAG interface (using a Digilent module or header connection);
- 2 differential SMA Transceivers (2 in input and 2 in output) (Kintex-7);
- 1 differential SMA dedicated to the transceiver differential clock input (Kintex-7);
- 2 SMA for the user (Kintex-7);
- 4 SMA dedicated for the SI5326 component, an Any-Frequency Precision Clock Multiplier;
- 1 SFP+ (Kintex-7);
- a specific connector called PMBUS to connect the 3 UCD9248 chips (see "Power Supply System") to a pc.

3.1.5 Components

Other important components of the Pixel-ROD are 2 oscillator that generate fixed clocks (1 for Kintex-7 and 1 for Zynq-7000), and 2 programmable demultiplexer (Kintex-7 and Zynq-7000), which can program:

- clock generators (1 for Kintex-7 and 1 for Zynq-7000);
- a PLL to generate clocks based on clocks in input in it (Kintex);
- RAM memory (Kintex-7);
- PMBUS (see later) (Kintex-7 and Zynq-7000).

3.1.6 Switches, Bottoms and LEDs

There are on the board some switches, LEDs and bottoms very important for the configuration of the board and for the knowledge of its status:

- 2 bottom to de-program the 2 FPGAs (SW14 for Kintex-7 and SWZ4 for Zynq-7000);
- 2 LEDs from the "done" signal, which makes enlighten the LEDs when the programming of a FPGA is gone well (DS20 for the Kintex-7 and DSZ3 for the Zynq-7000);
- a switch (SW20) to configure the board to use the Digilent JTAG chip (01) or the JTAG connector (10);
- a switch (SWZ16) to configure the Zynq-7000 FPGA to use a particular connector or memory to upload the firmware.

3.1.7 Power Supply System

Respect the other structure of the board, the power supply system of the Pixel-ROD has to be re-invented. Indeed, to maintain the possibility to insert the board on a pc case, the power supply systems of KC705 and of ZC702 have been merged and not copied. The 12V voltage arrives from a Molex connector and it provides power for 3 Digital Pulse-Width Modulation (DPWM) System Controller (UCD9248) chips, which control the power up of the board. This particular power up system is due because the new FPGA technology requires a particular and precise chain of synchronous power up of all the board components. Every UCD9248 is a 4 rail, 8 phases synchronous buck digital PWM controller designed for non-isolated DC/DC power applications. When multiple power stages are configured to drive the voltage rail, UCD9248 distributes

the phases to all the DPWM output in order to minimize the ripple. The UCD9248 integrates dedicated circuitry for DC/DC loop management with RAM, flash memory and a serial interface to support configurability, monitoring and management. In order to facilitate the configuration of these devices, a PC based Graphical User Interface (GUI), named Fusion Digital Power Designer, is provided by Texas Instrument. This tool allows to configure the system operating on parameters for the application, storing the configuration to on-chip non-volatile memory. Furthermore, it is possible to get real time measurements of sensed voltages, once the device is configured and correctly interfaced. The connection from the chip and the pc is made by a dedicated USB adapter and a specific device named PMBUS connector. After the configuration files have been uploaded on the 3 devices, each device can control 4 DC/DC Switching regulators, which are of 3 different types:

- PTD08A020W, a single output switching regulator which supply 20 A;
- PTD08A010W, a single output switching regulator which supply 10 A;
- PTD08D210W, a double output switching regulator which supply 20 A.

Each UCD9248 manage 4 rails and 4 of the possible phases, and they do it using 4 signals:

- a DPWM output signal through which the switching regulator defines the output voltage;
- a EAN/EAP differential signal which provides the control of the voltage value choose for the output of the switching regulator;
- a Current Sense (CS) input signal which controls the choose range of the output of the switching regulator;
- a Fault (FLT) signal which reports an error in the parameter choose by the user.

Chapter 4

Preliminary Tests

As the description just exposed shown, the Pixel-Rod is a complex device with important features and incredible versatility. The final task of this device will be to work hours and hours constantly with a continuous stream of data, always active check system and continuous functioning processing system. So, to be sure of the correct functioning of the board, the first step to do to ensure the board capacities is to test its components.

4.1 Power On

The power system of the Pixel-ROD is a complex architecture custom built to allow the union of the KC705 and the ZC702. Because of the complex architecture of the Pixel-ROD, the power on system must not only "give power" to the board, but it must do it in a particular way to protect the board's components from damages. To ensure the correct power on, the Pixel-ROD uses 3 chips called UCD9248, power controllers chips which can be programmed by a specific tool designed by the Texas Instrument, the manufacturer of the chips. To reduce the complexity of the test and the danger of damages, the Pixel-ROD power system has been sectioned using solder pads, so that all the parts of the board has been tested separately. After ascertains that all the parts could receive the power, the UCDs were programmed using the software Fusion Digital Power Designer, provided by Texas Instrument, connecting a pc to the board by the PMBUS connector. Because of the Pixel-ROD complexity, the power on it isn't a normal "switch and all the board is ready", but the rail that transmits the voltage in all the board must follow a particular ignition chain. In Figure 4.1 and 4.2 are shown the simulation and a real image, took with the oscilloscope directly by the board, of this chain. The principal problems founded during these tests were: the "forgot" connection of 2 solder pads, a DC/DC mounted backwards, a couple of feedback signals reversed, a block of 3 signals with the names inverted. This bad lists, tough it may seems a defeat, in fact demonstrates the solidity of the project, because the only problems found have been only

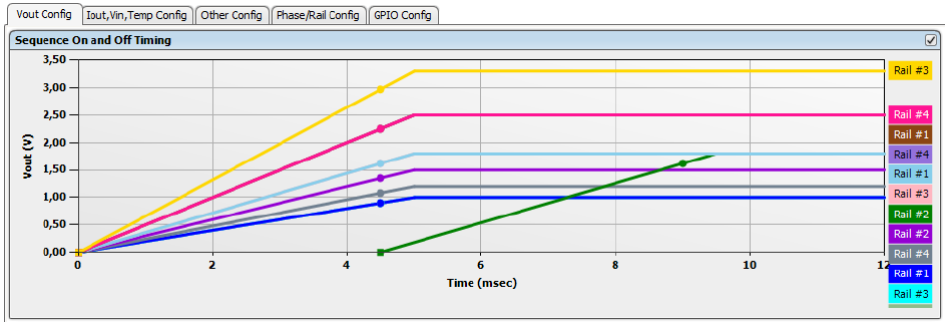


Figure 4.1: Simulation of the ignition chain.

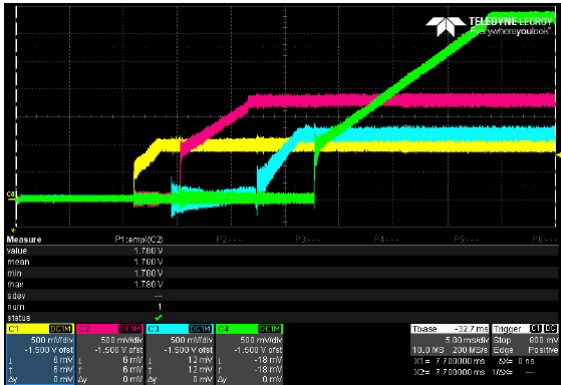


Figure 4.2: Image through oscilloscope of the real ignition chain.

little distraction problems, not great design problems.

4.2 System Clock and Internal Bus Test

Here start my work, indeed my thesis job has been to test some of the Pixel-ROD components and implements some of the particular firmware for important implementations of this device. As we said, the 2 Pixel-ROD FPGAs are connected directly with 3 type of internal buses. To test these buses, firmware written in VHDL has been used. The idea, for the 5 single line and the differential clock line, was to generate 2 counter with the same frequency in each FPGA, send the data from the Zynq-7000 to the Kintex-7, and start a subtraction operation between the data produced. To synchronise the data to subtract, a signal has been sent from the Zynq to the Kintex. Then, some GPIOs of the Kintex has been used to check the goodness of the transmission, which is shown by a not-asserted signal of the GPIOs on the oscilloscope, made at 200 MHz using as clock the system oscillator. For the 21 bit differential bus, a similar operation has been made, but in this case the subtraction was built subtracting the "n" data received by the Kintex from the Zynq, and the "n-1" data. So the result will be always a logic "1", apart when the bus will be full of logic "1" and it will restart the count, or in other word, "0->21 bit=0" minus "1->21 bit=1". In the first case the Kintex will emit a logic "0" by a GPIO, in the latter case it will emit a logic "1". The logic "1" emission will occur every

$$t_{period} = 2^{21} * \frac{1}{200 * 10^6 Hz} \simeq 10.5 ms$$

. So, using an oscilloscope, it is possible to see the logic "1" about every 10.5 ms, to be sure of the good success of the test.

4.3 FPGAs RAM Tests

The next test was about the performance of the DDR3 RAM memories installed in the board. For the test was used the Intellectual Property (IP) Integrator, a tool of the Vivado suit, which is a Xilinx software, which permits the production of firmware using cores provided by a catalog available in the Vivado suit, and connecting them with the AXI4 interface. This type of programming method consents to build big and complex structure, like emulating processor, with a relatively simplicity. In this particular case, for the Kintex FPGA, the work has been to implement a Microblaze, a 32 bits soft processor, to allow to "transform" the Pixel-ROD in a programmable computer where, based on the block needed to test, this block has been instantiated and, as we said with the use of the AXI4 interface, connected to the emulated processor. The most important IP cores used for this test are following described.

Microblaze

This is the core which represents the emulated processor. Completely customizable, the signals with which it works are:

- Clk: represents the reference clock of the design;
- Reset: the reset used in all the design;
- Interrupt: this signal manages all the interrupts sent by the other IP cores;
- Debug: input always connected with the IP core Microprocessor Debug Module, which connects the Microblaze to the Xilinx System DeBugger, which permits to send command directly to the soft processor like read or write on its registers;
- DLMB and ILMB: Data Local Memory Bus and Instruction. This signals allow the primary access to the on-chip block RAM;
- M_AXI_DC and M_AXI_IC: Master AXI Data Cache and Instruction Cache. This 2 signals allow the interface of the Microblaze with the memory cache, using the AXI4 interface;
- M_AXI_DP: Master AXI Data Peripheral. It allows the interface of the Microblaze with the peripherals connected to it, always using the AXI4 interface.

Memory Interface Generator

This is the controller and physical layer to interface 7 series FPGA and other AXI4 slave devices, to DDR3 memory. Completely customizable, the most important signals with which it works are:

- S_AXI, which is the input of the AXI4 interface used for all the transactions like configuration's commands and data that will be written on the DDR3;
- SYS_CLK: input of the clock. Possible frequencies are 100 MHz or 200 MHz;
- ui_clk: User Interface output Clock. Reference clock for all the design produced multiplying the SYS_CLK;
- DDR3: the interface towards the DDR3 memory.

After building the firmware using the cores just described and others, the final part of the preparation for the test was to implement in the "machine", built with the firmware, the software, in C++ language, to perform the test. This was made using another Xilinx tool called System Development Kit, which, recognising the "machine" built with all its

registers and components, permits to implement the C++ program that will make the test, and allows to control the status of it and the sending of commands to the Microblaze by the Xilinx System Command-line Tool (XSCT) terminal. The test has consisted of writing the same word (AAAA5555) in all the possible registers of the DDR3 with the possibilities to be written. The success of the test was stated: through a terminal available by the UART port, and using the XSCT terminal to write directly in a register of the memory. Figure 4.3 and 4.4 show the results of the tests. For the Zynq's FPGA,

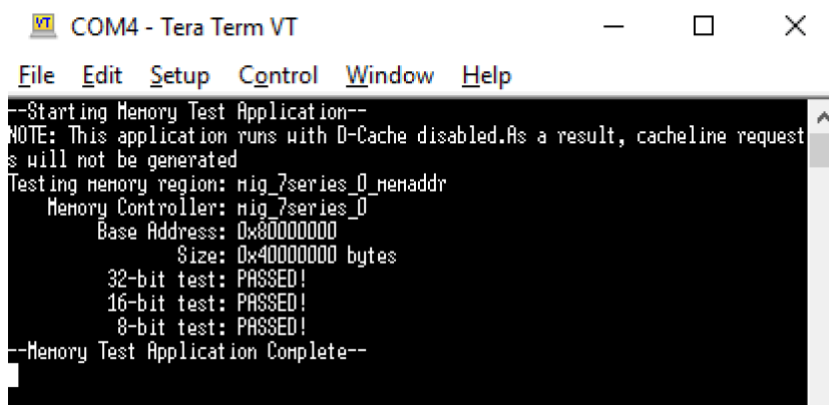


Figure 4.3: Image of the terminal showing the results of the DDR3 test on Kintex.

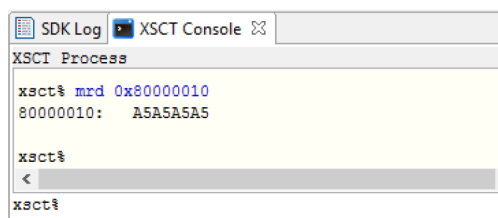


Figure 4.4: Image showing the XSCT terminal during a write operation on the Kintex RAM.

the procedure from the SDK is the same used for the Kintex's FPGA. The difference is in the IP core used, which in this case was a single core which implements all the necessary functionalities. This is due to the fact that, as we already said, the Zynq's FPGA has an ARM processor integrated, which can be instantiated using a single core, and which has all the useful connections to the Zynq's components and memory already done. In Figure 4.5 and 4.6 are shown the results of the test.

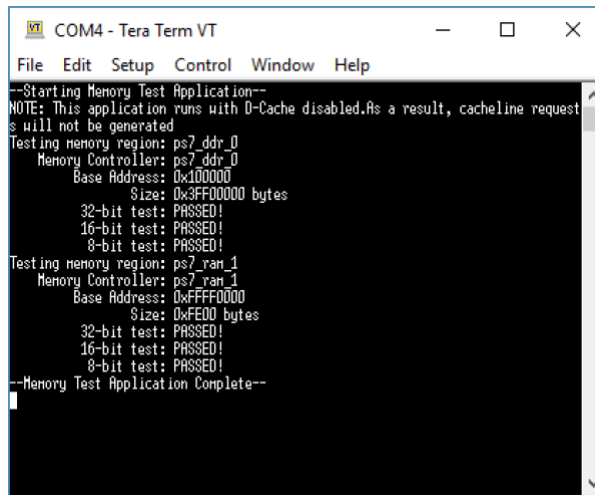


Figure 4.5: Image of the terminal showing the results of the DDR3 test on Zynq.

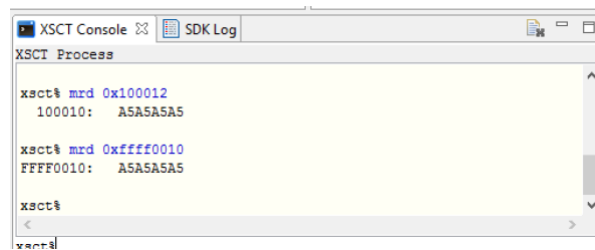


Figure 4.6: Image showing the XSC terminal during a write operation on the Zynq RAM.

4.4 Ethernet Ports Tests

The next components tested were the ethernet ports. The ethernet port is an important connection that allows to connect the board to local nets or global nets. Also in this case the IP Integrator has been used, with the instantiation of a Microblaze for the Kintex's FPGA, and the instantiation of the Zynq core for the Zynq's FPGA. The only difference in the firmware was the use of a core called AXI Ethernet Subsystem in the Kintex architecture, a customizable core where the most important signals are:

- S_AXI, used to configure the subsystem;
- S_AXI_TXD and S_AXI_TXC, respectively Transmit Data and Transmit Command, which are the reserved AXI4 input to communicate with the subsystem;
- M_AXI_RXD and M_AXI_RXS, the reserved AXI4 reception for data and status of the subsystem;
- MDIO, the Management Data Input Output used to configure the PHY;
- GMII, the Gigabit Media Independent Interface, which is connected to the board's ethernet port.

Like in the memory tests, the SDK tool provides a test for the ethernet port, test that consists in the implementation of an "echo" that permits to write by keyboard on a terminal and see what someone wrote on the same terminal, which terminal is on a pc connected via ethernet with the Pixel-ROD. In other words, the symbols sent to the board through ethernet connection are, after the sending, sent from the board to the pc and read on the terminal. In Figure 4.7 and 4.8 are shown the images of this echo test.

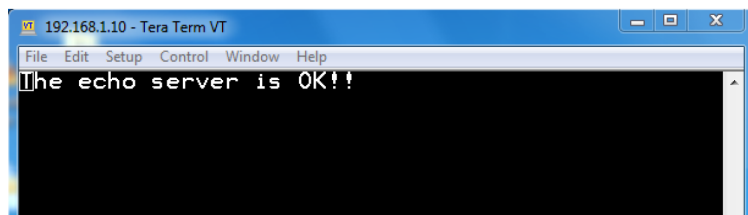


Figure 4.7: Image of the terminal showing the opening of the echo link for the Kintex FPGA.

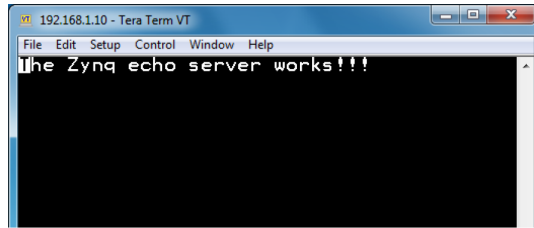


Figure 4.8: *Image of the terminal showing the opening of the echo link for the Zynq FPGA.*

4.5 I2C LVDS Oscillator Tests

The PR contains 2 fixed clock generators which produces a 200 MHz clock (1 for the Kintex-7 and 1 for the Zynq-7000), a PLL generating clocks based on clocks in input in it (Kintex), 2 clock generators (1 for the Kintex-7 and 1 for the Zynq-7000), as we already said; furthermore, is available a particular FMC HPC connectible mezzanine, named FM-S14, which its most important feature is to multiply by 5 the number of optical transceivers available on the board; furthermore it has the important feature to have 2 clock generators, components which can generate any clock frequency between 15.48 MHz and 1.3 GHz changing the values of their registers, using the I2C protocol (Appendix). To test these 2 latter components an I2C master was used, which can write and read on the registers of the components using the VHDL language. The I2C master firmware wrote consists of a VHDL program with a state machine for the write operation and read operation on 1 or 2 slaves, using to write on the slave the Page Write method. Instead to read data from the slave the "normal" method has been used. The state machine provided allows a write-read operation sequentially for all the registers (for example: the FM-S14 has 24 registers, so the master firmware consists of 24 write operations and successively 24 read operations), to be sure that the 8 bits words just wrote are correct. The firmware permits to check all the words read using 2 signals sent from the FPGA to 3 General Purpose Input/Output (GPIO). The SCL value is set at 100 kHz. To simulate the goodness of the master before use it on the components, it has been used another firmware which, even using 2 FIFOs, simulates an I2C slave. It allows to write an amount of data choose by the user on the FIFOs, and read it when the FIFOs are not empty. In Figure 4.9 is shown an image taken with an oscilloscope of the write-read operation in action (using 3 GPIOs), and of 2 GPIO signals showing a correct write-read operation. The clock built, showed in Figure 4.10 , has a frequency of 256 MHz.

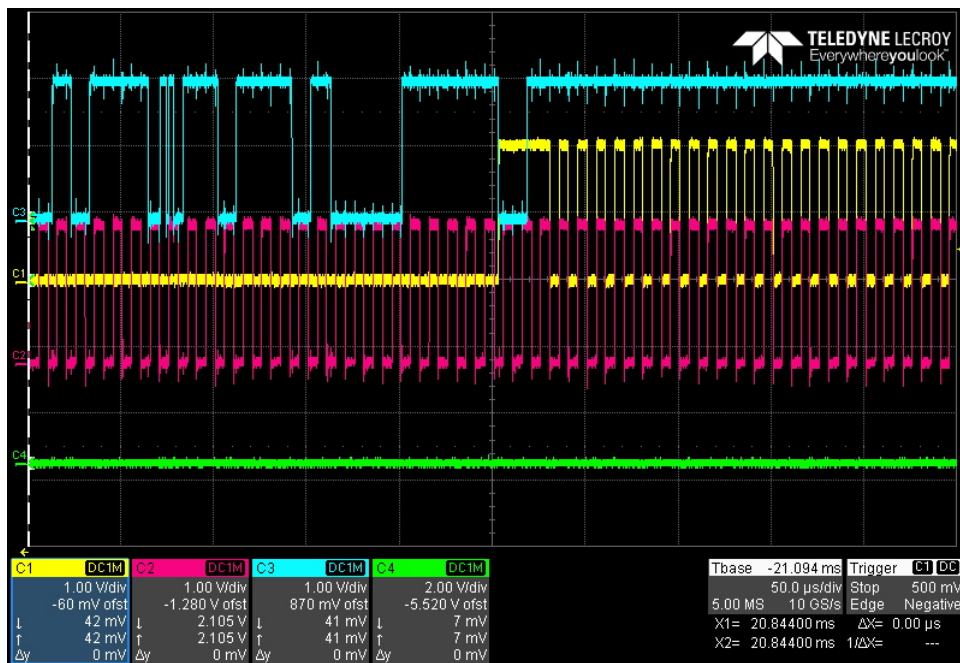


Figure 4.9: Image of the write and read operation and of the 2 checking signals (2 equal counters at 100 kHz) on the oscilloscope.

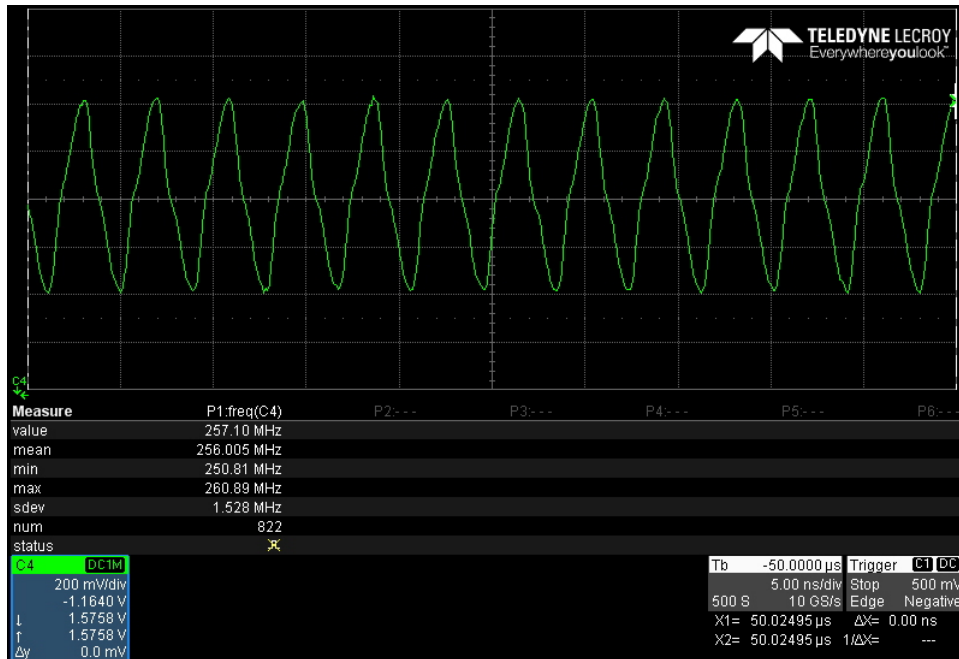


Figure 4.10: Image of the 256 MHz clock generated measured by oscilloscope.

4.6 Fast Links Tests

The most important tests on the Pixel-ROD were the tests of the functioning and of the performance of the 16 fast links present on the PR and managed by the FPGA Kintex. These links, together with the ARM processor, are the most important features of the Pixel-ROD. The Vivado Design suit provides, to test the transceiver of the Xilinx's boards, a special core called IBERT (Appendix), which allows to form a link between 2 transceivers or a loop-back on 1 transceiver. The instrumentation available allowed the test of the 2 differential SMAs and of the differential SMA dedicated to the reference clock, of the 4 transceivers present in the FMC HPC connector using the FM-S14, and of the integrated optical connection. In the SMA case the test consisted in a loop-back test, transmitting data from the transceiver and connecting the SMA coaxial cables used at the transceiver reception, then checking the transmission quality. For the 5 SFP+ connectors instead, a link between the PR integrated SFP+ and each one, individually, of the SFP+ connectors of the FM-S14 mezzanine was made. To produce the reference clock, at frequency of 250 MHz, we have used the FM-S14 mezzanine. While the SYS_CLK_I has been produced by the Pixel-ROD integrated oscillator, at 200 MHz. The work has started instantiating the IBERT core and customizing it. The customization interface is divided in 3 pages. In the first page ("Protocol Definition") the protocols used for the tests are provided; they are always custom protocols with:

- rate: 5 Gb/s or 10 Gb/s (the first rate was to be sure of the correct functioning of the hardware, the second rate to test it at maximum speed);
- data width: 40 bits;
- reference clock: 250 MHz (it must be equal to $\frac{rate}{n*10}$ with the range of Natural number n depending on the PLL used);
- quad count: 1 (SMA) or 2 (SFP+, because the SFP+ integrated transceiver and the FMC HPC transceivers are in different quad);
- QPLL: YES (it is used the Quad PLL shared by the 4 transceivers of 1 quads).

In the second page of the customization interface ("Protocol Selection"), the transceiver or transceivers, based on which are under test, are chosen. In the third page of the customization interface ("Clock Settings") are provided other clock settings. For all the 2 tests the same settings have been chosen:

- Clock Type: System Clock (the clock provided by the board);
- Source: External (meaning not present in the transceiver quad);
- I/O Standard: LVDS (Low Voltage Differential Signal);
- P Package Pin and N Package Pin: AD12 and AD11;
- Frequency: 200 MHz.

In the last page there is the summary of all the customization. After that, automatically the settings are saved and the Output Product is generated. This file permits to build the firmware in VHDL, and then, if there aren't any modifications to do at the firmware, and after the instantiation of the file constraint (the file in which: are instantiated the ports that will be connected to the signals used and are written some settings in TCL Language automatically generated), the architecture is ready to become a binary file that will be uploaded in the Pixel-ROD. The interface provided by the IBERT core is shown in Figure 4.11 . This interface is accessible if the link between the transceivers was correctly created, and after choosing which transceivers the user wants to check. This interface allows to:

- see the actual speed rate, the bit transmitted, the erroneous bit and the BER value ($\frac{erroneous\ bit}{trasmitted\ bit}$);
- choose the pattern to send and the pattern that must be checked;
- inject erroneous bit during the transmission;

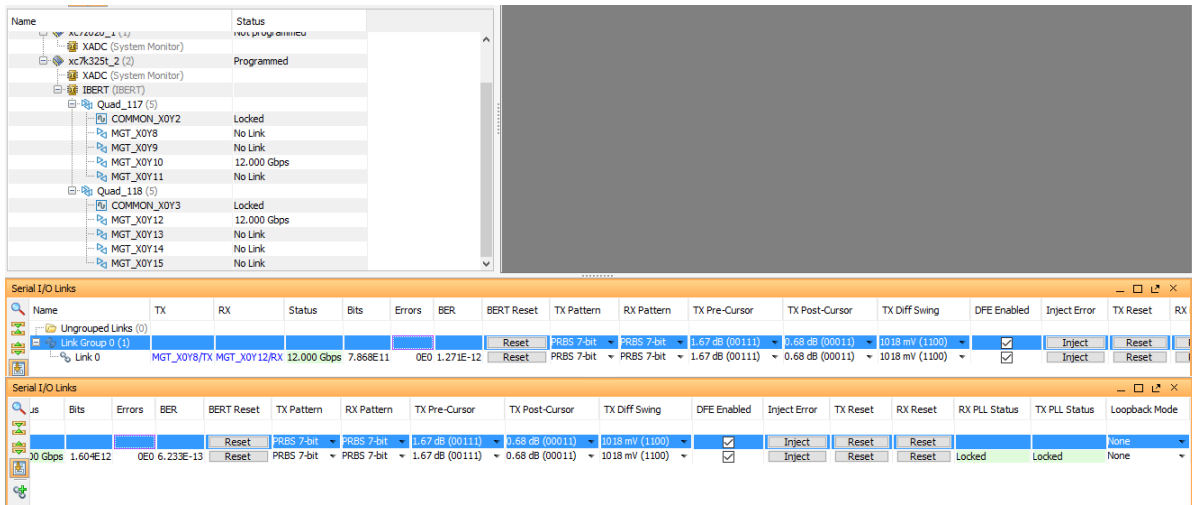


Figure 4.11: Core IBERT interface, where the 2 "Serial I/O Links" parts are in fact 1 divided in 2 for layout motivations.

- reset the BER operation, the transmission and the reception (these latter are very important for the Pixel-ROD because at the first connection of the link the BER is high ($\simeq 10^{-9}$), and to drive it at good values ($\simeq 10^{-13}$), the tx and the rx must be reset);
- choose if to do a loop-back Near-End (inside the FPGA), Far-End (through the SFP+ connections), at the PMA layer level, or at the PCS layer level.

In the Figure 4.12 and 4.13 are shown the results of the test for the SFP+ components,

Name	TX	RX	Status	Bits	Errors	BER	BE
Ungrouped Links (0)							
SFP to FM-S14 (1)							
Link 0	MGT_X0Y10/TX	MGT_X0Y15/RX	10.000 Gbps	3.987E12	0E0	2.508E-13	
FM-S14 to SFP (1)							
Link 1	MGT_X0Y15/TX	MGT_X0Y10/RX	10.000 Gbps	3.961E12	0E0	2.524E-13	

Figure 4.12: SFP+ BER.

and because the SMA components results are the same, they aren't be shown. In particular the Scan Eye images, which denotes the performance of a fast connection by the dimension of the "eye" formed by the overlap of the transmission and of the reception signals, show the goodness of the transceiver connection tested; as well as the "low"

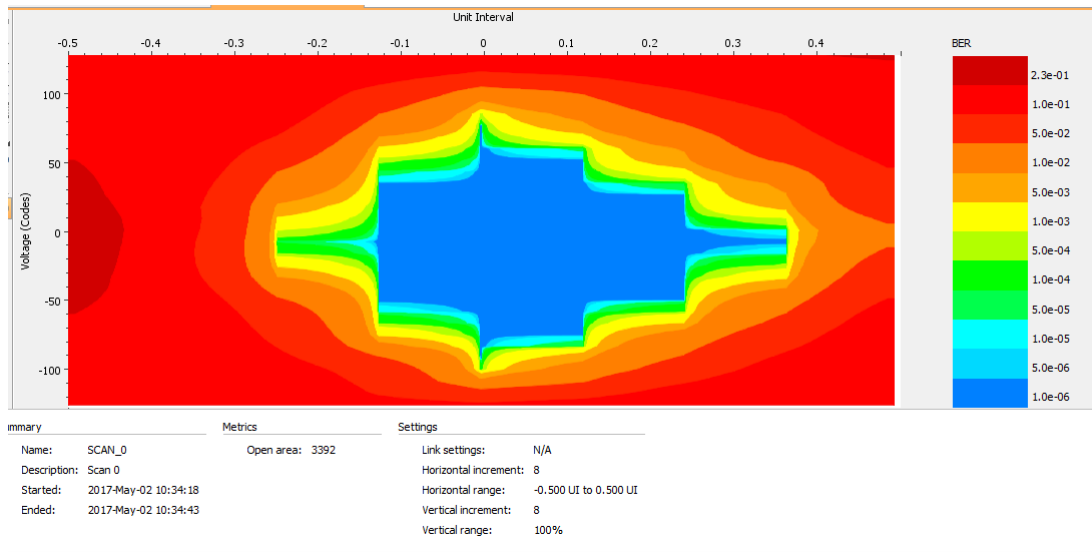


Figure 4.13: *SFP+ Scan-Eye*.

BER and the number of erroneous bit (0 error). A particular reminder at which must be pay attention, is the fact that each SFP+ has a signal, named TX_DISABLE, which is always in pull up. This signal disables the transmission through SFP+ connector, so, to enable it, TX_DISABLE must be de-asserted. In the case of the integrated SFP+ this is possible using a "jumper", a little component which can connect 2 parts of a trail on the board, jumper which connects the TX_DISABLE signal to mass, de-asserting it. In the case of the FM-S14's SFP+s, the signal must be de-asserted using a firmware, which de-asserts the 4 TX_DISABLE signals, one for each of the SFP+ of the mezzanine.

4.7 PCI Express Test

The PR is, as we already said, a PCIe based board, with a PCI Express 8x of second generation, which has a nominal data rate of 4 GB/s, 2.5 GB/s real (during the test). To test this type of communication, an IP core of the Xilinx, the DMA/Bridge Subsystem for PCI Express, has been used. This core, synthesizing its complexity, uses a firmware created for a KC705 board, and software made for a Linux machine at which the testing board must be connected. This test writes pattern of bits on all the RAM memory of the board under test, a Pixel-ROD in this case, passing through the PCIe connector, and then read all the RAM just wrote, calculating the time spent to make this "read and write" operation.

Chapter 5

Implementation

After the tests on the most parts of the Pixel-ROD have been concluded, it's started a work of many implementation on the PR, to demonstrate the versatility and the potential of this custom board. Until today the implementations have concerned the FPGA Kintex, and its fast connections.

5.1 Aurora 64b/66b Connection

The first use of the PR was the implementation of a communication protocol named Aurora 64b/66b (Appendix), a standard protocol in particular used for the output of a new pixel sensor called RD53a (Appendix) which will be used for the upgrade of the Pixel Detectors of ATLAS and CMS. The implementation of this transmission protocol, in this case via optical connection, has been done using an Intellectual Property of the suit Vivado, which instantiates an Aurora 64b/66b protocol and an error checker system to control the goodness of the transmission. This core (Appendix) allows to build a communication between 2 boards, or to make it using only one board that can do the transmitter and the receiver at the same time (loop-back). The purpose of this implementation is to emulate the one lane at rate 5.12 Gb/s protocol transmission of the RD53a using a commercial electronic board, the KC705, and use the Pixel-ROD to receive and read the data, casually generated, by the transmission core; while the 4 lanes at 1.28 Gb/s protocol has been tested using one PR and implemented a transmitter and a receiver on the same PR. For all the 2 protocols the mezzanine FM-S14, connected to the Pixel-ROD, has been used for the optical connection and for the generation of the reference clock, used by a Phased-Locked Loop of the transceiver to produce the correct bit rate. In Table 5.1 are shown the parameters used in the core interface of the Aurora 64b/66b IP of the Vivado suit. In the 2 protocols some changes have to be made to allow the operations. The same changes made for all the 2 protocols have been the use of 2 different GPIOs for the signals pma_init and reset, because the GPIOs automatically

Type of Protocol	1 lane at 5.12 Gb/s	4 lanes at 1.28 Gb/s
Line Rate (Gb/s)	5.12	1.28
GT Reference Clock (MHz)	256	256
Initial Clock (MHz)	200	200
GT DRP Clock (MHz)	100	100
Dataflow Mode	Simplex (TX or RX in base at the board)	Duplex
Interface (how the data must be send)	Streaming	Framing
Flow Control	None	None
AXI4 Lite (type of interface used)	Asserted	Asserted
Vivado Lab Tools (enable the debug cores)	Asserted	Asserted

Table 5.1: *Customization of the Aurora 64b/66b core.*

instantiated in the constraint of the IP core are pulled down in the KC705 (the board for which this IP core is built for by Vivado), while in the PR these GPIOs aren't pulled down, but pulled up. So to solve the problem 2 floating GPIOs, meaning that these 2 aren't pulled up or down, have been used, and pulled down using the constraint file.

Regarding the single protocol, instead:

- for the 1 lane, the management of the reference clock for all the 2 boards has been made in this way: for the PR by connecting the correct pins in the constraint file, for the KC705 the connection has been different; the reference clock of the KC705 has been taken by the "clock SMA" connection, the connectors used to give to the SMA transmission the reference clock, and the clock signal has been taken by the PR, copying the reference clock for the PR and sending it to 2 user SMA, 2 SMA connectors usable by the user for his purposes. After that the user SMA and the clock SMA has been connected by 2 coaxial cables. This operation, in the PR, has been made using 2 VHDL primitives: the primitive Output Double Data Rate (ODDR), to copy the reference clock signal, and the Output Buffer Differential Signal (OBUFDS), to send the signal to the 2 user SMAs;
- for the 4 lanes protocol instead, since in this case the FM-S14 must be both a receiver and a transmitter, for this latter situation the signal TX_DISABLE, signal that control the possibility to transmit data from a transceiver, must be de-asserted.

The core used a core ILA and 3 cores VIO for the debugging, where in the core VIO there are "virtual buttons" that permit to reset the transmission, both through a lane reset and through the channel reset, or resetting the pma_init signal. The signals data_err_count, soft_error and hard_error are the weapons given by the core to control the correct reception of the data, and during the tests these signals have given 0 error. In 5.1 and 5.2

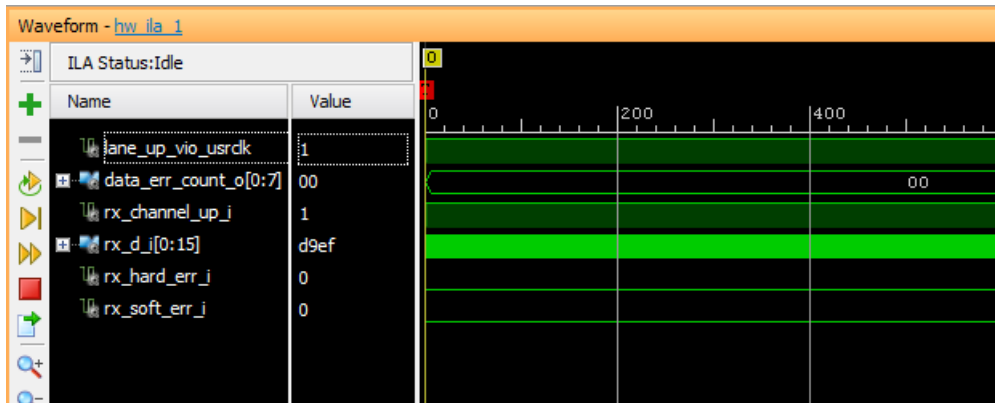


Figure 5.1: *ILA core interface of the 5.12 Gb/s protocol.*

are shown the waveform interface of the core ILA of the 2 protocols showing, among

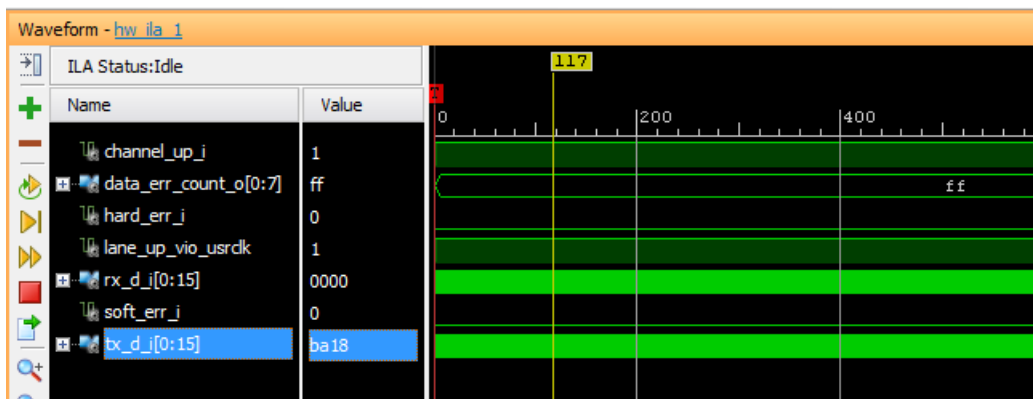


Figure 5.2: *ILA core interface of the 1.28 Gb/s protocol. data_err_count has the value "ff" because the optical fibers available aren't of the same length, so the frames that arrive in the 4 rx lane don't belong to the same word, and so they don't check.*

the other things, the data received and the error during the transmission; while in Figure 5.3 and 5.4 are shown the VIO cores with the signals lane_up and channel_up in high position, which confirm the connection.

5.2 GBT Protocol

The GigaBit Transceiver protocol transmission is an important protocol conceived by the CERN laboratories to allow the communication between the "on-detector" part of

Name	Value	Activity	Direction	VIO
lane_up_vio_j	[B] 1		Input	hw_vio_1
channel_up_in_initdk	[B] 1		Input	hw_vio_1
sysreset_from_vio_j	[B] 0		Output	hw_vio_1
greset_from_vio_j	[B] 0		Output	hw_vio_1

Figure 5.3: *VIO core interface of the 5.12 Gb/s protocol.*

Name	Value	Activity	Direction	VIO
lane_up_vio_j	[B] 1		Input	hw_vio_1
greset_from_vio_j	[B] 0		Output	hw_vio_1
channel_up_in_initdk	[B] 1		Input	hw_vio_1
sysreset_from_vio_j	[B] 0		Output	hw_vio_1

Figure 5.4: *VIO core interface of the 1.28 Gb/s protocol.*

a read-out system, subject to radiation damage (50-100 Mrad for the Pixel Detector read-out system), and the "off-detector" part, protected by these type of damages. The CERN protocol consists of an ASIC, named GBTx, situated in the on-detector part, which sends the Front End data to electronic boards situated in the off-detector part, and the transmission protocol is, in fact, the GBT protocol, transmitted by particular optical fibers, radiation damage resistant. To test this type of transmission has been used 2 Pixel-ROD, one to emulate the GBTx ASIC and one to receive the data transmitted.

Chapter 6

Future Developments and Conclusion

The tests carried out so far on the Pixel-ROD have proved the reliability of the board and its capability to be interfaced with other electronics through many different electrical connectors and optical fibers. Furthermore, the first two implementations of important transmission protocols such as the GBT and the Aurora 64b/66b, used in the High Energy Physics research area, have been also successful. Now the next tests on this board are oriented towards a variety of directions. Besides the completion of some hardware tests (mainly memories) that have not been exhaustively performed yet, we are starting to design a set of firmware blocks to interface the board even more easily. A first task to meet very soon is the collaboration with the Felix Group, i.e. a group of universities and laboratories which has developed a series of electronic boards called Felix. This collaboration started in July 2017 with our tests at NIKHEF in Amsterdam to create the basis of a data acquisition system using the Pixel-ROD directly interfaced with Felix card. In addition, the Pixel-ROD can interface directly with the future front-end readout chips, as the RD53a. For this task, we are also creating a data acquisition system based on synchronized clock lanes working for all the synchronous components of the Pixel-ROD, primarily the Zynq-7000 as Master and the Kintex-7 as Slave component. Moreover, we need a set of software tools to control, configure, test, and in general use the entire system. As possible applications for this board we are thinking also to make something available to develop a test-stand for the ATLAS community to qualify the RD53a chips that are currently under fabrication. In this case the board will interface the chip and will summarize its functional performance. Other institutes are designing other test systems so that there is also the change to converge to a multi-platform test-stand. In any case, Bologna might provide part of the work. Another feasible application is to interface again with the Felix card, either for data-acquisition of trigger functions, to update any detector that will be upgraded before the LHC phase-2. For example, in 2019 a readout update is foreseen for the ATLAS New Small Wheel of the muon detector

and for the Liquid Argon electromagnetic calorimeter. It might be possible that Bologna will help to build something, hardware or firmware, whatever the request. Finally, for the time being, we are also providing and simulating firmware to be used within the Pixel-ROD board as front-end emulator. In fact, many detectors need a front-end compatible data provider in advance with respect to the actual front-end construction time. This is useful for example to test, in advance, the readout chain for calibration and data taking purposes.

Chapter 7

Appendix

7.1 FPGA

A FPGA is a device containing a circuitry made of an array of reconfigurable logic gates. After configured, in the internal logic are built connections that provide the software application from the hardware implementation. A FPGA uses dedicated hardware for processing. They are parallel in nature, so different processes do not have to compete for the same resource, and so the performances of a single part are not affected by the addition of another process. FPGA-based control systems can enforce critical interlock logic and can be designed to prevent I/O forcing by an user. A FPGA contains, in his integrated circuit (IC), millions of gates. The main components of a FPGA are the Configurable Logic Blocks (CLB), the Programmable Interconnects (PI) and the I/O Blocks (I/OB). A scheme of it is shown in Figure 7.1 . The logic blocks are implemented using multiple level low fan-in gates, and can be configured even to emulate a microprocessor. They can be implemented by:

- transistor pairs;
- combinational gates like basic NAND gates or NOR gates;
- n-input Look-Up Tables;
- multiplexers;
- wide fan-in AND-OR structure.

Routing in FPGA consists of wire segments of varying lengths which can be interconnected via electrically programmable switches. The configuration of the FPGA's components, CLB, PI and I/OB, can be of 3 types:

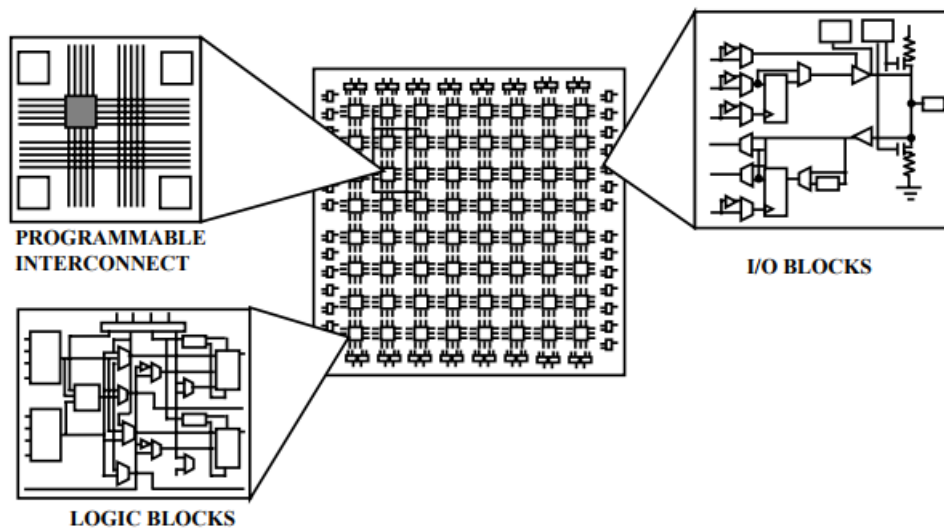


Figure 7.1: *Scheme of an FPGA layout.*

- Symmetrical arrays: here CBLs are arranged in rows and columns of a matrix and interconnect by PIs. At the periphery there are the I/OBs which provide the communications to and from the outside. Each CLB consists of an n-input Look-Up Table and a pair of programmable flip flops. I/O blocks also control functions such as tristate control, output transition speed. Interconnects provide routing path. Direct interconnects between adjacent logic elements have smaller delay compared to general purpose interconnect.
- Row based Architecture: here the structure is of alternating rows of logic modules and Programmable Interconnect tracks. I/O are at the end of each row. Rows connect each other vertically by interconnect. Combinatorial modules contain only combinational elements while sequential modules contain both combinational elements along with flip flops. Routing tracks are divided into smaller segments connected by anti-fuse elements between them.
- Hierarchical Programmable Logic Device: this architecture is designed in hierarchical manner with top level containing only logic blocks and interconnects. Each logic block contains number of logic modules and each logic module has combinational as well as sequential functional elements. Each of these functional elements is controlled by the programmed memory. Communication between logic blocks is achieved by Programmable Interconnect arrays. I/O Blocks surround this scheme of logic blocks and interconnects.

As we said, FPGAs are based also on a supply of uncommitted wires to route signals,

wires that are connected by the user and therefore must use an electronic device to connect them. 3 types of devices have been commonly used to do this: pass transistors controlled by a SRAM cell, a flash memory or EEPROM cell to pass the signal, or a direct connection using anti-fuses.

7.2 JTAG

The devices used in these type of researches and work need a debug and testing structure safe and performing. In order to overcome this, the best system is the standard JTAG, in particular Standard Test Access Port (TAP) and Boundary Scan (BS) Structure.

7.2.1 Boundary Scan

The main advantage offered by using Boundary Scan technology is the ability to set and read the values on pins without direct physical access. All the signals between the device's core logic and the pins are intercepted by a serial scan path known as the Boundary Scan Register (BSR) which consists of a number of boundary scan "cells". In normal operation these boundary scan cells are invisible. However, in test mode the cells can be used to set and/or read values from the device pins (or in "internal" mode from values of the core logic). The collection of boundary scan cells is configured into a parallel-in, parallel-out shift register. A parallel load operation, called a "capture" operation, causes signal values on device input pins to be loaded into input cells, and signal values passing from the core logic to device output pins to be loaded into output cells. A parallel unload operation, called an "update" operation, causes signal values already present in the output scan cells to be passed out through the device output pins. Data can also be shifted around the shift register in serial mode, starting from a dedicated device input pin called Test Data In (TDI) and terminating at a dedicated device output pin called Test Data Out (TDO). The Test Clock (TCK) is fed through another dedicated device input pin, and the mode of operation is controlled by a dedicated Test Mode Select (TMS) serial control signal. The tasks of these signal are:

- TCK: this signal synchronizes the internal state machine operations.
- TMS: this signal is sampled at the rising edge of TCK to determine the next state.
- TDI: this signal represents the data shifted into the device's test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.
- TDO: this signal represents the data shifted out of the device's test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state.

- TRST: this is an optional pin which, when available, can reset the TAP controller's state machine.

In Figure 7.2 is shown a scheme of BS and TAP structure. Thanks to these components

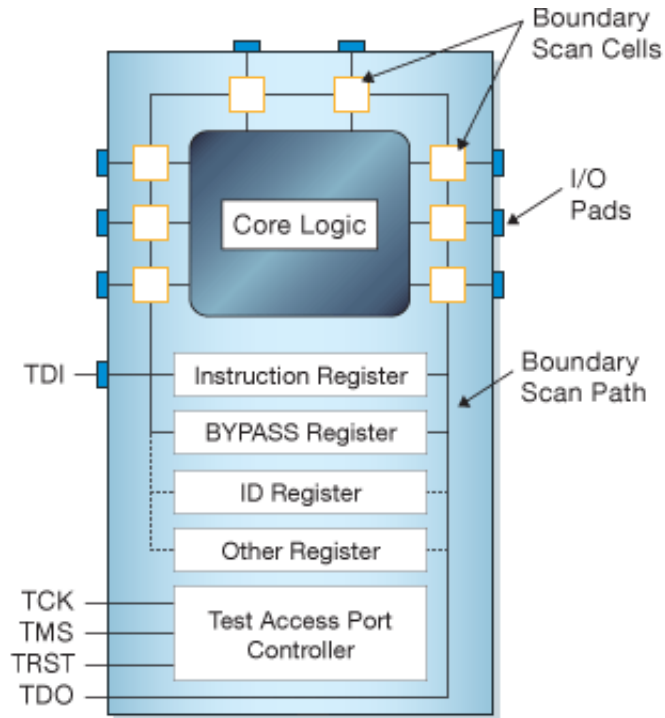


Figure 7.2: *Scheme of the BS and TAP architecture.*

installed in the device, particular tests can be applied to it, after interconnects it via the global scan path, and successively: by loading the stimulus values into the appropriate device-output scan cells via the edge connector TDI (shift-in operation), applying the stimulus (update operation), capturing the responses at device-input scan cells (capture operation), and shifting the response values out to the edge connector TDO (shift-out operation). A Boundary Scan cell has four modes of operation: normal, update, capture, and serial shift. During normal mode, DataIn is passed straight through to DataOut. During update mode, the content of the output register is passed through to DataOut. During capture mode, the DataIn signal is routed to the shift register and the value is captured by the next ClockDR. During shift mode, the ScanOut of one register flip-flop is passed to the ScanIn of the next via a hard-wired path. Note that both capture and shift operations do not interfere with the normal passing of data from the parallel-in terminal to the parallel-out terminal. This allows the capture of operational values "on the fly" and the movement of these values for inspection without interference with functional

modes of operation. The use Boundary Scan cells to test a device's core functionality is called "internal test" or simply InTest. The use the Boundary Scan cells to test the interconnect structure between two devices is called "external test" or simply ExTest.

7.2.2 Registers

There are two types of registers associated with Boundary Scan: Instruction Register and Data Register, where each compliant device has one instruction register and two or more data registers. The Instruction Register holds the current instruction. Its content is used by the TAP controller to decide what to do with signals that are received. Most commonly, the content of the instruction register will define to which of the data registers signals should be passed. Regarding the Data Registers, we have:

- Bondary Scan: this is the main testing data register. It is used to move data to and from the I/O pins of a device;
- BYPASS: this is a single-bit register that passes information from TDI to TDO. It allows other devices in a circuit to be tested with minimal overhead;
- Identification (IDCODES): this register contains the ID code and revision number for the device. This information allows the device to be linked to its Boundary Scan Description Language (BSDL) file. The file contains details of the Boundary Scan configuration for the device.

7.2.3 Test Access Port Controller

The TAP controller, a state machine whose transitions are controlled by the TMS signal, controls the behaviour of the JTAG system. In Figure 7.3 is shown the state-transition diagram. All states have two exits, so all transitions can be controlled by the single TMS signal sampled on TCK. The two main paths allow to set or retrieve information from either a data register or the instruction register of the device. The data register operated on (e.g. BSR, IDCODES, BYPASS) depends on the value loaded into the instruction register.

7.2.4 Boundary Scan Instructions

The IEEE 1149.1 standard defines a set of instructions that must be available for a device to be considered compliant. These instructions are:

- BYPASS: this instruction causes the connection of TDI and TDO lines via a single-bit pass-through register (the BYPASS register). This instruction allows the testing of other devices in the JTAG chain without any unnecessary overhead;

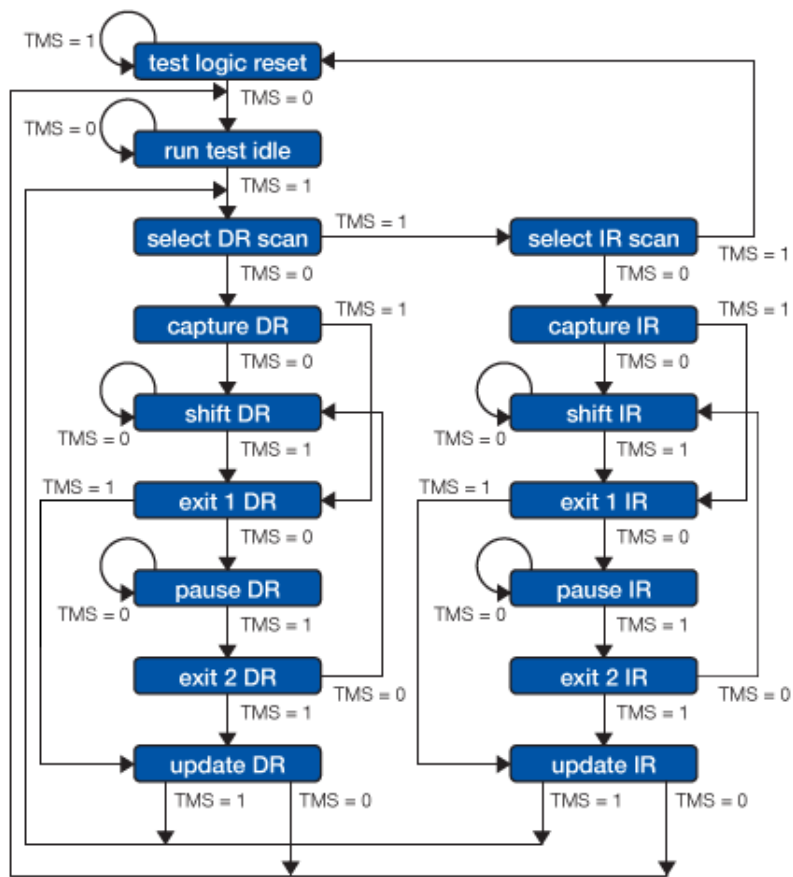


Figure 7.3: *Scheme of state transition of the TAP controller.*

- **EXTEST:** this instruction causes the connection of TDI and TDO to the Boundary Scan Register (BSR). The device's pin states are sampled with the "capture dr" JTAG state and new values are shifted into the BSR with the "shift dr" state; these values are then applied to the pins of the device using the "update dr" state;
- **SAMPLE/PRELOAD:** this instruction connects the TDI and TDO to the BSR. However, the device is left in its normal functional mode. During this instruction, the BSR can be accessed by a data scan operation to take a sample of the functional data entering and leaving the device. The instruction is also used to preload test data into the BSR prior to load an EXTEST instruction.

Other commonly available instructions include:

- **IDCODE:** this instruction causes the connection of TDI and TDO to the IDCODE register;
- **INTEST:** this instruction causes the connection of TDI and TDO lines to the Boundary Scan Register (BSR); while the EXTEST instruction allows the user to set and read pin states, the INTEST instruction relates to the core-logic signals of a device.

7.3 AXI4

The AXI4 is part of ARM AMBA 4.0, a family of micro controller buses. There are 3 types of AXI4:

- AXI4, which is developed for high-performance memory-mapped requirements;
- AXI4 Lite, which provides simple, low-throughput memory-mapped communication (for example: to and from control and status registers);
- AXI4-Stream, which provides high-speed streaming data.

With memory mapped we intend that all the transactions involve the concept of a target address within a system memory space and data to be transferred. The AXI4 work is to interface and connect Intellectual Property (IP) cores of the Xilinx suit, and it does it with some benefits like:

- standardizing the AXI interface, so the user must learn only one protocol for IP;
- AXI4 allows burst of up to 256 data transfer cycles with just a single phase, while AXI4 Stream allows unlimited data burst size and it hasn't phases for interfaces or transfers, so they are not considered to be memory-mapped. AXI4 Lite allows only 1 data transfer for transaction.

In the following lines we will describe briefly the method with which AXI interface works. The AXI specifications describe how an AXI Master and an AXI Slave, each representing IP cores, connect and transfer informations with each other. Memory-mapped AXI Masters and Slaves are connected by a structure called Interconnect block, which contains AXI-complaint master and slave interfaces, which it use to connect one or more masters with one or more slaves. AXI4 and AXI4 Lite use the following interface channels: Read Address, Read Data, Write Address, Write Data, Write Response. Data can move in both direction between master and slave simultaneously, and data can carry many transfer sizes (1 up to 256 and unlimited, as said before). The simultaneous bidirectional data transfer is possible thanks to the fact that AXI4 provides separate data and address connections to read and write, with the possibility to use a single address and then burst data. In Figure 7.4 and 7.5 are shown the schemes of write and read

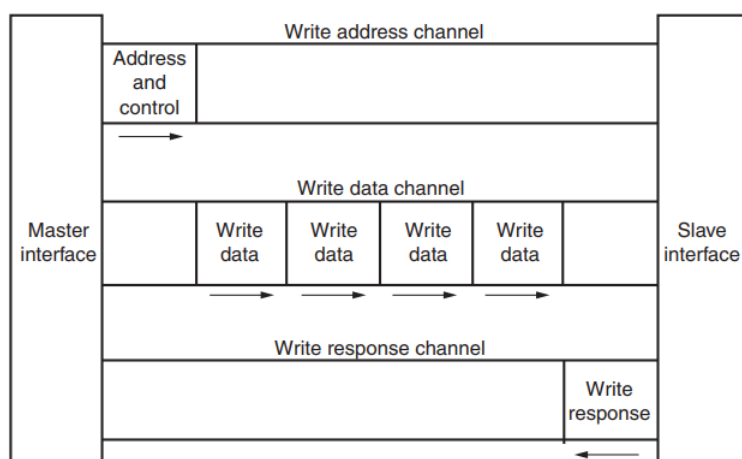


Figure 7.4: Scheme of the write operation for the AXI4 interface, where the time line is from left to right.

operation through AXI4. In addition to the features just said, AXI4 protocol permits to achieve high data throughput even thanks to: data upsizing and downsizing, multiple outstanding address and out-of-order transaction processing. The clock connection in AXI4 is provided for each master-slave pair, each has its clock. The AXI4 protocol allows the insertion of register slices, called pipeline stages, to aid in timing closure. AXI4 Lite and Stream are slightly different from AXI4:

- AXI4 Lite not support bursting;
- AXI4 Stream defines a single channel for transmission of streaming data, channel which is modelled after the write data channel of AXI4. Furthermore transfer can not be reordered. AXI4 Streaming can burst an unlimited amount of data and

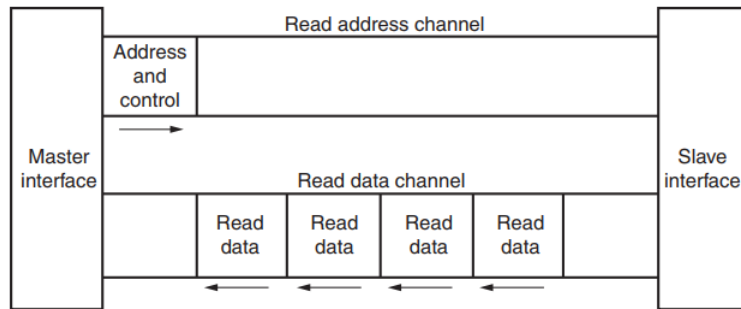


Figure 7.5: Scheme of the read operation for the AXI4 interface, where the time line is from left to right.

the AXI4 Stream-complaint interfaces can be split, merged, interleaved, upsized, downsized.

The AXI specification provides a framework that defines protocols to move data between IP using a defined, standard signal structure. This standard ensures that IP can exchange data with each other and those data can be moved across a system.

7.4 IBERT LogiCore IP

IBERT (Integrated Bit Error Ratio Tester) LogiCore IP is an Intellectual Property of the Vivado suit to evaluate and monitor the GTX transceivers of the 7 series FPGA, and is based on the functionality of the IBERT type core. This core provides a Physical Medium Attachment (PMA, first layer in the OSI model of the computer network theory) evaluation and demonstration platform. Logic that communicates with the Dynamic Reconfiguration Port (DRP) of the GTX transceiver permits to manage the transceivers, change registers and attributes that control the port values. The communication at run time is carry out by a JTAG connection. The IBERT has particular features like:

- provides a user-selectable number of transceiver;
- customizes the transceiver for the desired number of lines, line rate, clock rate, reference clock source and datapath width;
- system clock available from pins or one of the enabled GTX transceiver;
- data pattern generators and checkers for each GTX transceiver;
- TX pre-emphasis and post-emphasis;

- TX differential swing;
- RX equalization;
- Decision Feedback Equalizer (DFE);
- Phase-Locked Loop (PLL) divider settings.

In particular, the reference clock (signal name is REFCLK) can be sourced from either CPLL or QPLL, where a Channel PLL (CPLL) is a PLL for each transceiver, and a Quad PLL (QPLL) is a PLL shared between all the four transceivers of a quad, where a "quad" is where a set of 4 transceivers are united in one bank of the FPGA. Talking about the pattern generation and checking, the possible patterns that can be generated are: Pseudo-Random Binary Sequence (PRBS) of 7-bit, 15-bit, 23-bit or 31-bit width, clock patterns (clock with frequency = system clock frequency/2 or /10). The pattern sent is compared with pattern internally generated in the receiver. If the receiver receives 5 cycles of correct patterns then the LINK signal is asserted, otherwise, when it is asserted, if receives 5 cycles of incorrect patterns the LINK signal is de-asserted. From an operative point of view, in a single design can be implemented 3 line rates, where each one can be a pre-provided industry standard protocol, or can be a customize line, specifying:

- speed of transmission;
- width of the words to send;
- reference clock frequency;
- in which quads the line rate must be used;
- if use the QPLL.

After deciding which protocol to use, the user can activate the transceiver in base at a location choice, which is provided by a table, and which quad reference clock to use. After that, there is the choice of the source clock, if it must be provided by FPGA pin or by the reference clock itself. If the source clock frequency is higher then 150 MHz, a Mixed-Mode Clock Manager provides the timing constraints. At last the receiver output clock permits to pull out a recovered clock from any serial transceiver by a probe, and resets are available for the BERT counters and all GTX transmissions and receptions. Following are showed the IBERT core ports and their functions:

- SYSCLK_I: an input clock that locks all communication logic. This port is present only when an external clock is selected in the generator;
- TXP_O, TXN_O: an output which transmits differential pairs for each of the GTX transceivers used;

- RXP_O, RXN_O: an output which receives differential pairs for each of the GTX transceivers used;
- GTREFCLK0_I, GTREFCLK1_I: the GTX transceiver reference clock. The number of GTREFCLK ports can be equal to or less than the number of TX and RX ports because some GTX transceivers can share clock inputs;
- RXOUTCLK_O: output of the quad based RX output clock.

7.5 Aurora Protocol

The Aurora 64b/66b protocol is a link-layer protocol that can be used to move data point-to-point across one or more high-speed lanes. The protocol describes how to transfer user data through serial mode connection using Aurora channel, consisting of one or more Aurora lanes. The transfer can be full-duplex or simplex. Aurora channels have the following properties:

- data are transferred through Aurora channel in frames;
- frames share the same channel with control information as flow control messages, clock compensation sequences and idles;
- frames can be of any possible length and can have any possible format. Only the delineation of frames is defined by the specifics;
- frames can be interrupted at any time by flow-control messages or idles.

A schematic overview of the protocol is shown in Figure 7.6 .

7.5.1 Data Transmission and Reception

The transmission of data in Aurora 64b/66b is performed by using 64-bit codes called block, where each one of these blocks can be transmitted through an Aurora channel, one per cycle per lane. Since this situation, the blocks are prioritized, to resolve the conflict that can be born if two or more blocks need to be sent in the same lane. The blocks are:

- Clock Compensation: this idle block can be used to prevent data corruption dues to small differences between recovered clock and local reference clock; this problem could happen if different clock sources are used to drive TX and RX clocks;
- Not Ready: this idle block is sent while attempting to align data from the channel and perform channel bonding;

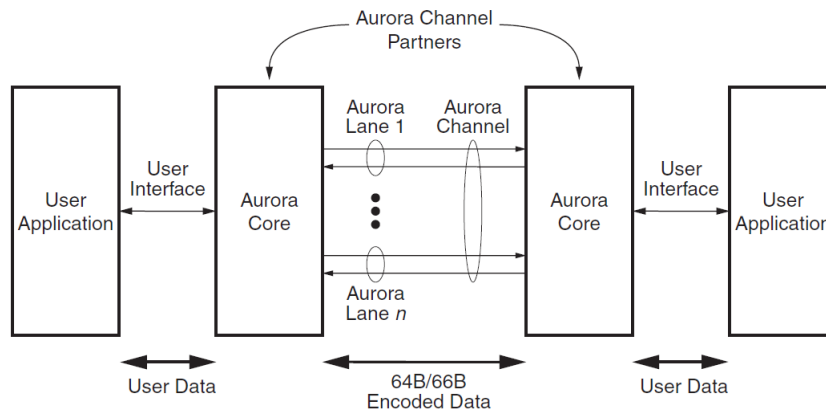


Figure 7.6: Overview of the Aurora 64b/66b protocol.

- Channel Bonding: idle block, used to bond the channel, it is sent to every lane in the channel simultaneously, and the receiver lanes receive this block if their channel bonding FIFOs are adjusted;
- Native Flow Control: it requests native flow control from the Aurora interface on the other side of the channel;
- User Flow Control: the same as before but the flow control messages are customized by the user;
- User K-Block: this not decoded block is passed directly to the user and can be implemented with specific control application functions;
- Data: this block, with the following 2, creates frames carrying user data; in particular this block carries eight octets of data;
- Separator: this block indicates the end of the current frame; the next frame begins in the next block. Separator blocks carry from 0 to 6 octets of data;
- Separator-7: it does the same of Separator, but always carries 7 octets of data;
- Idle: it is transmitted when no other higher priority blocks can be transmitted. It is the lowest priority block.

7.5.2 Frame Transmission Procedure

Like the Figure 7.7 show, the procedure of transmission of a frame through the user application to an initialized Aurora channel is:

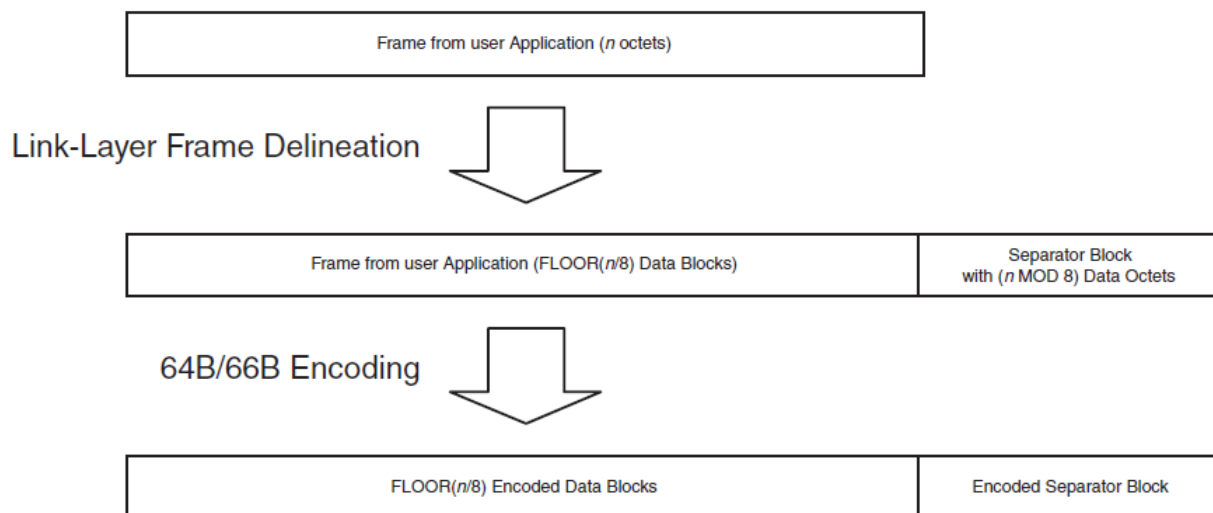


Figure 7.7: Scheme of the transmission procedure of a frame in the Aurora 64b/66b protocol.

- frames are delineated at their end using Separator and Separator-7 blocks, fact that permits the channel partners to distinguish between different frames;
- data and separator block are 64b/66b encoded by a Physical Coding Sub-layer (PCS) prior the transmission, transforming 64-bit blocks to 66-bit blocks;
- the encoded blocks, comprising the frame, are serialized for transmission, using a differential non-return-to-zero (NRZ) format.

7.5.3 Frame Reception Procedure

Like the Figure 7.8 shown, the reception procedure of a frame from an initialized Aurora channel, with the final passage to a user application, is:

- de-serialization of the data stream;
- decoding of the 66-bit block into 64-bit;
- the data and separator blocks decoded are intermingled with control blocks such as Control Compensation, Channel Bonding, Flow Control, User K_Block and idles.

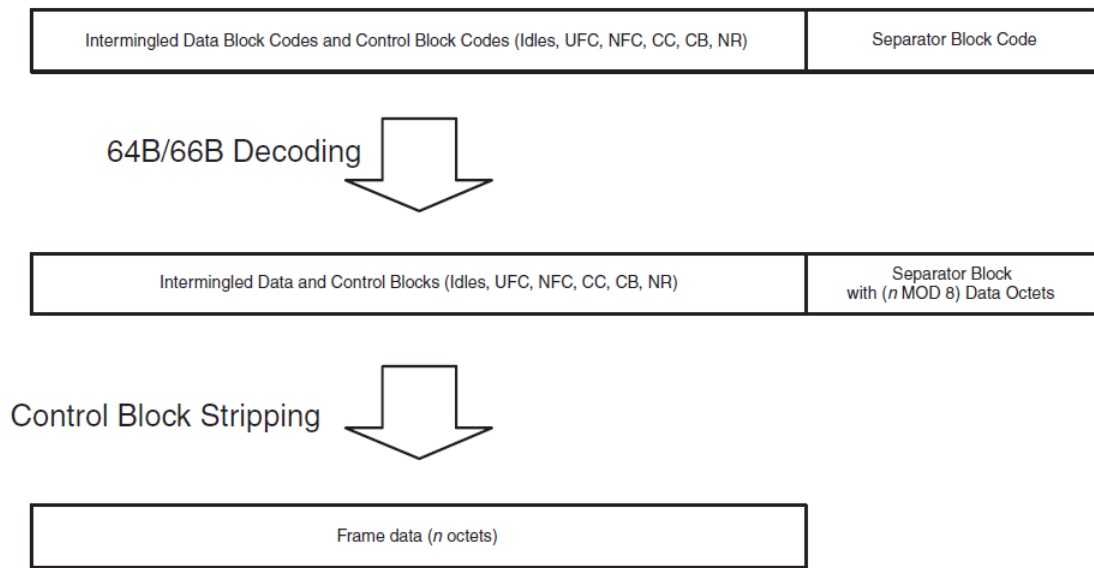


Figure 7.8: Scheme of the reception procedure of a frame in the Aurora 64b/66b protocol.

7.5.4 Flow Control

Aurora 64b/66b supports the following 2 flow control mechanisms:

- Native Flow Control: a link-layer flow control mechanism which allows receivers to request that their channel partners transmit idle instead if data;
- User Flow Control: this mechanism allows short, high-priority control messages to be sent through the Aurora channel.

NFC messages are not cumulative, indeed if a new NFC message arrives while an Aurora interface is still processing a previous NFC request, a new Pause value immediately replaces the old. A Pause value is carried by the NFC message and tells to the channel to elevate the priority of the idle on the data priority until the number of data blocks requested by Pause are transmitted on each of its lanes. UFC can be sent at any time as long as the channel is initialized, and it is made of UFC header blocks and a set of Data Blocks. The transmission of NFC or UFC messages modified the priority of the 10 blocks.

7.5.5 Initialization and Error Rate

The procedure to prepare (initialize) an Aurora 64b/66b channel is composed by 2 stages:

- Lane Initialization: during this stage, each serial lane in the channel is individually reset and aligned to the block boundaries of incoming data;
- Channel Bonding: during this stage Aurora interface uses Channel Bonding block to compensate for the skew between each of the individual lanes. After the success of the bonding, the lanes are treated as a single communication channel.

When the lanes in the Aurora interface receive Channel Bonding blocks, they must adjust their PCS latency, so that the Channel Bonding blocks could be all available at the RX data interface of the PCS simultaneously. When the Aurora channel control logic "knows" that all the lanes in the channel are simultaneously delivering Channel Bonding blocks, channel bonding is complete. Full-duplex Aurora interface go to the Wait For Remote state when their channel partner appears not ready to receive data. Simplex Aurora interface go to Channel Ready state if they can send and receive data freely (like full-duplex), depending on their type. Talking about the errors that can occur during the transmission, these can be of 2 type:

- soft errors, which are transient, statistical errors, expected in a normal transmission situation;
- hard errors, which regard catastrophic or irrecoverable errors, like channel disconnection, buffer overflow, or hardware failure. A cascade of soft errors could represent an hard error, like loss of lock.

7.5.6 PCS Layer and PMA Layer

In Figure 7.9 is shown the scheme of the physical layers of Aurora 64b/66b.

PCS Layer

- Aurora Encoding: specifies how data and control information must be encoded before transmission through an Aurora channel, and decoded upon reception. All data and control information in Aurora 64b/66b are encoded in 64-bit blocks. Each 64-bits block is marked with a sync header value, indicating whether or not it is a Data, or Control block. The sync header is combined with the block to form a block code;
- 64b/66b Scrambling: whenever a lane transmits a block code, the 8 octets, following 2 bits sync header, must be scrambled using a self-synchronizing scrambler with the polynomial $G(x) = 1 + x^{39} + x^{58}$. When a lock code is received, the 8 octets, following 2 bits sync header, must be de-scrambled using a self-synchronising de-scrambler with the same polynomial. The 2 bits sync header don't undergo the scrambling or the de-scrambling.

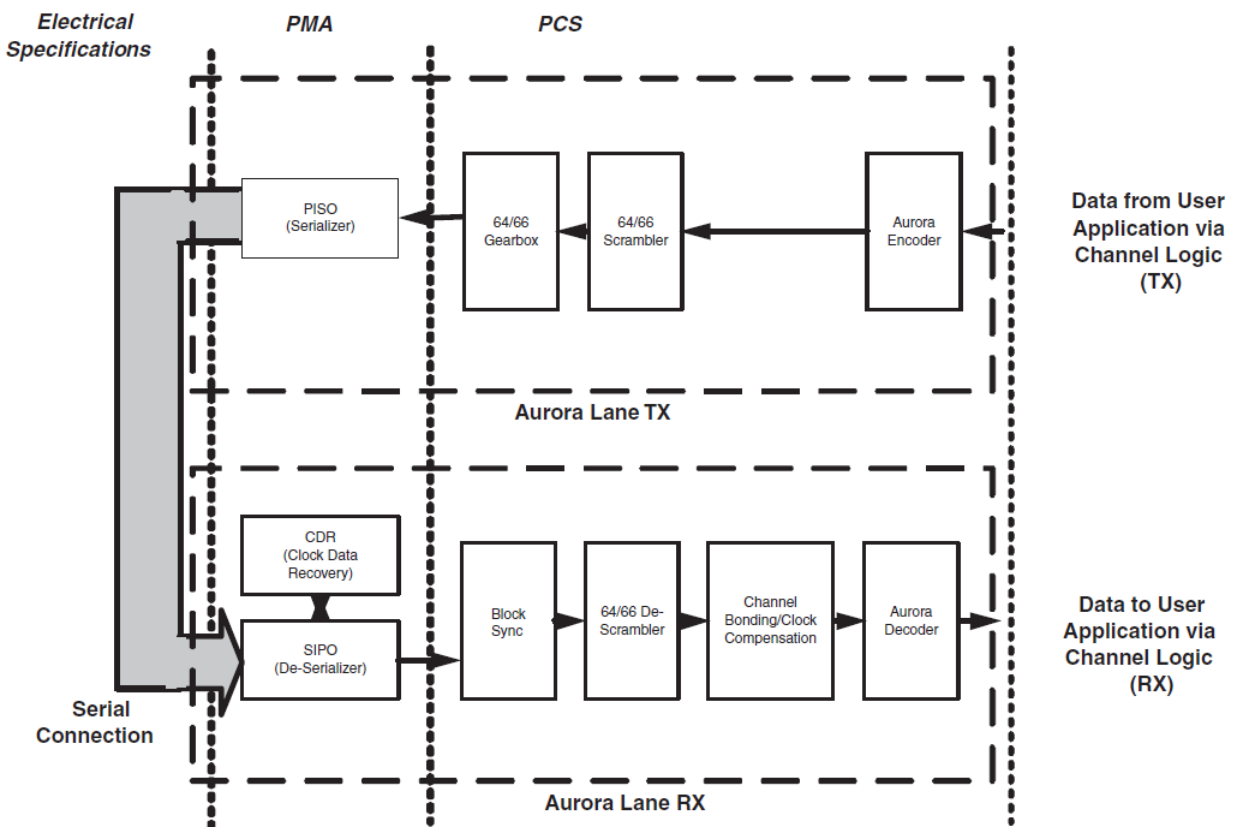


Figure 7.9: Scheme of the Aurora 64b/66b logic blocks and physical layers.

- 64b/66b Gearbox: after exits from the scrambler, the gearbox combines the 64 bits block and the 2 bits sync header to present a 66 bits block to the PMA layer. Otherwise, before entering the de-scrambler, the gearbox separates the 66 bits block to send at the de-scrambler the 64 bits block and the 2 bits sync header;
- Channel Bonding: as we already said, the Channel Bonding has the task to connect the partners of the transmission and reception, using the Channel Bonding Blocks;
- Clock Compensation: Clock Compensation idle blocks can be used to prevent data corruption due to small differences between the recovered clock and local reference clock. These differences occur when independent clock sources are used to drive the clocks on the TX and RX side of a connection. If a shared clock is used, Clock Compensation blocks are not required.

PMA Layer

- Bit and Byte Ordering Convention: the leftmost bits of the encoded block code are the sync header bits. These are the most-significant bits of the block code. The leftmost byte of the block is the most significant byte. The rightmost byte is the least-significant byte;
- Serialization: each lane must transmit the most-significant bit of each block code first, followed in order by the remaining sync bit and the bits of the block code (Figure 7.10);

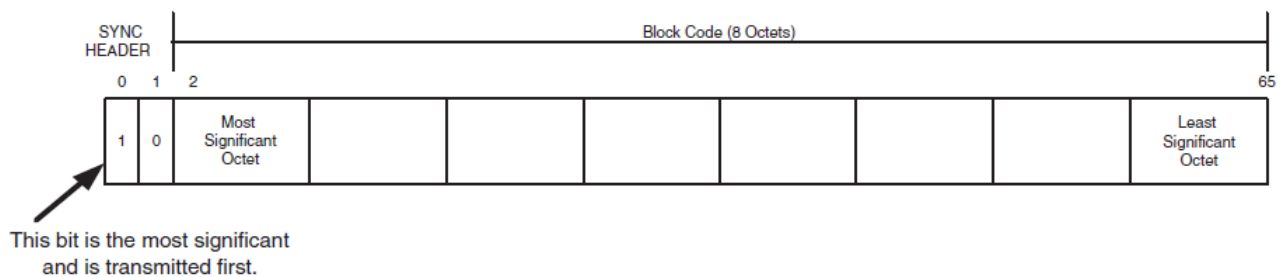


Figure 7.10: *Ordering of the Aurora 64b/66b serialization.*

- De-serialization: each lane should expect the data that it de-serializes to be in the serialization order presented in "Bit and Byte Ordering Convention";
- Clock Data Recovery: the recovery of the high speed serial clock incoming from the transmission stream.

7.6 Aurora Logicore IP

Aurora 64b/66b Logicore IP is an Intellectual Property of the Vivado suit which provides a functional Aurora 64b/66b protocol for every type of 7 series transceiver (GTX, GTH, GTY), for a part of Xilinx's devices. This core can run up to 16 lanes at total rate from 500 Mb/s to 400 Gb/s, and it uses AMBA protocol AXI4-Stream user interface. Aurora 64b/66b core automatically initializes a channel if connected to another Aurora 64b/66b channel partner. After initialization, data can be sent in frames or through streams, where frames can be of any size and streams are unending frame. Excessive bit errors, disconnections or equipment failure cause the core reset. Thanks to the use of the 64b/66b encoding, this core has a transmission overhead of 3 %. The core is made by 4 big blocks:

- lane logic, which drives the GTX and GTH transceiver, handles the encoding and the decoding, and performs the control errors;
- global logic, which performs the channel bonding for channel initialization;
- RX and TX user interface, which uses the AXI4-Stream interface to receive and transmit data, and furthermore both perform the flow control and the TX the clock compensation.

This core provides a method to measure its latency in number of user_clk cycles, using the AXI4-Stream user interface.

7.6.1 Clock and Reset Management

The principle signals for the clock system are:

- user_clk, which synchronizes all signals between the core and the user application;
- tx_out_clk, which is selected such that the data rate of the parallel side of the module matches the data rate of the serial side of the module, taking into account 64b/66b encoding and decoding;
- sync_clk, which is used to drive txusrclk port of the serial transceiver;
- txusrclk, which drives the serial transceiver.

The principle signals which manage the reset system and the power on are:

- reset_pb, which is the first to de-assert to start the core, furthermore it is used (asserting it) to restore the Aurora 64b/66b core to a known starting state;

- pma_init, which is the second to de-assert to start the core (reset_pb must be de-asserted), and it is used (asserting it) to reset the entire serial transceiver which eventually resets the Aurora 64b/66b core as well.

Figure 7.11 show the initialization overview, which it must be accomplish for all the data

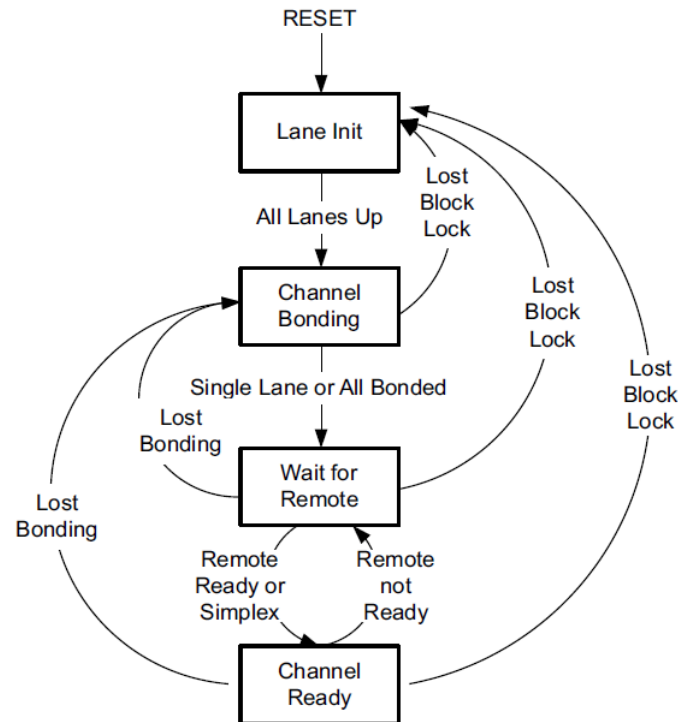


Figure 7.11: *Scheme of the Aurora 64b/66b initialization procedure.*

transmission (which takes the channel_up signal to high), instead to receive data the lane initialization is enough (lane_up signal to high).

7.6.2 User Signals in Transmission and Reception

Frame Data

The frame transmission and reception rules are written here.

Transmission

The Aurora 64B/66B core samples the data only if both s_axi_tx_tready and s_axi_tx_tvalid are asserted. The AXI4-Stream data are only valid when they are framed. Data outside

of a frame are ignored. To end a frame, the user must assert `s_axi_tx_tlast` while the last word (or partial word) of data is on the `s_axi_tx_tdata` port, and use `s_axi_tx_tkeep` to specify the number of valid bytes in the last data beat. The steps followed by the core during the data transmission are:

- to accept data from the user application on the `s_axi_tx_tdata` bus;
- to indicate the end of frame when `s_axi_tx_tlast` is asserted along with `s_axi_tx_tkeep` and to stripe data across the lanes in the Aurora 64b/66b channel;
- to insert idle or pause cycles on the serial line when the user application de-asserts `s_axi_tx_tvalid`.

Reception

The `m_axi_rx_tvalid` signal is asserted concurrently with the first word of each frame from the core. The `m_axi_rx_tlast` signal is asserted concurrently with the last word or partial word of each frame. The `m_axi_rx_tkeep` port indicates the number of valid bytes in the final word of each frame using the same byte indication procedure as `s_axi_tx_tkeep`. All bytes valid are indicated (all 1s) when `m_axi_rx_tlast` is not asserted and the exact number of bytes valid is specified when `m_axi_rx_tlast` is asserted. The framing efficiency is affected by two factors: the size of the frame and the data invalid request asserted by the gearbox that occurs every 32 `user_clk` cycles. Furthermore the gearbox in the GTX and GTY transceiver needs one periodic pause to accomplish the encoding and the clock divider ratio, pause taken every 32 `usr_clk` cycles de-asserting the `s_axi_tx_tready` signal for one cycle.

Stream Data

In the streaming of data there aren't frame delimiters, and the Aurora 64b/66b channel is used as a pipe. The streaming Aurora 64b/66b interface expects data to be filled for the entire `s_axi_tx_tdata` port width (integral multiple of eight bytes). When `s_axi_tx_tvalid` is de-asserted, gaps are created between words which are preserved except when clock compensation sequences are being transmitted. For both frame and stream data interface, there is not a buffer in the reception side, so the data must be immediately read.

7.6.3 Flow Control

To control the data flow during transmission and reception, Aurora 64b/66b core implements a Flow Control interface, which is composed of 3 types: the Native Flow Control, the User Flow Control, and the User-k Block Control. Here in Figure 7.12 and 7.13 and 7.14 are shown the scheme of the NFC, UFC and UBC port interfaces.

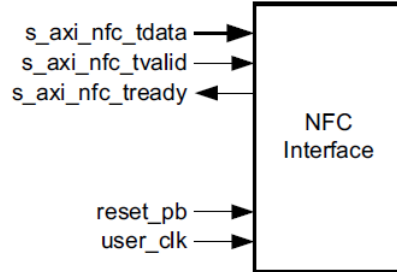


Figure 7.12: Scheme of the NFC logic block.

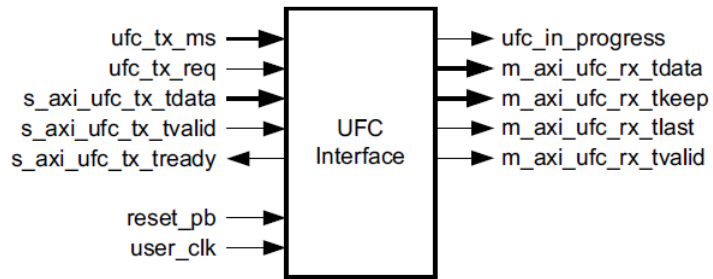


Figure 7.13: Scheme of the UFC logic block.

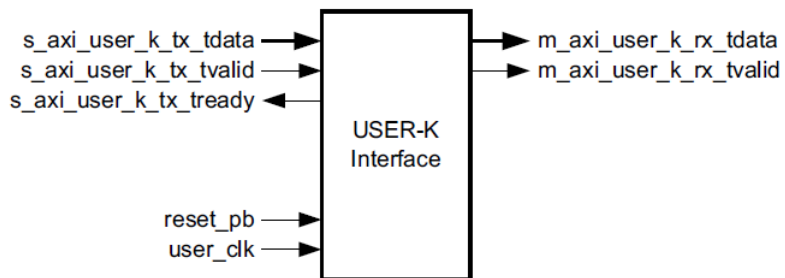


Figure 7.14: Scheme of the UBC logic block.

7.6.4 Error Signals

The Aurora 64b/66b core allows to control the errors that could be made during transmission or reception by the use of two signals:

- `hard_err`: this signal is asserted when an overflow error or an underflow error occurs in transmission or reception, or when the user clock and the reference clock aren't synchronise. This type of error provokes a system reset.
- `soft_err`: this signal is asserted when a bit error occurred. This type of error don't cause a system reset, unless a cascade of `soft_err` signal asserted occurs.

7.6.5 System Debug

The IP Aurora 64b/66b has a debugging system incorporated, which allows to check the assertion of the transmission and reception, to reset at many levels the data transmission, to check for any error during the communication. These cores are 1 core ILA and 3 cores VIO, which allow:

- to check the data transmitted and received with the signals: `tx_d_i`, `rx_d_i`;
- to reset the transmission and reception at many level: resetting the `pma_init` signal, resetting the core generally, to prepare a series of resets that will be made when the user will want to; some of the signals doing this are: `sysreset_from_vio`, `gtreset_from_vio`;
- to check if the lane or lanes are bonded and, consecutively, if the channel is bonded, the signals are: `lane_up_vio_usrclk`, `lane_up_vio_i`, `channel_up_initclk`;
- to check if error in the frame received (`data_err_count`), soft error or hard error are present during the communication;
- to instantiate and check the loop-back mode.

7.6.6 Other Features

An important feature in the simplex mode of transmission and reception is the Auto Link Recovery, which is based on the reception of Channel Bonding patterns. This method allows the RX simplex core to come up independently of the TX simplex core bring-up; instead, without it, the Aurora 64b/66b core must follow a specific reset sequence, where the TX simplex core needs to be in reset or should keep sending initialization sequences until the RX simplex core is up. Other features are implemented in the core. The DRP interface allows the user to monitor and modify the transceiver status. The Transceiver Debug Interface permits the debug for 7 series and UltraScale devices. The CRC (Cyclic Redundancy Checker) interface is a further frame checker.

7.7 I2C Protocol

The I2C protocol is a synchronous serial multi-master multi-slave protocol which is composed by 2 lines, where the transmission and the reception go on a single line, while the other line is for the clock transmission. It works thanks to 2 signals, SCL and SDA, where:

- SCL (Signal CLock) is an output of the master, giving the functioning clock to the slave. Possible values of clock generated by the master are 100 kHz or 400 kHz (or a value depending on the component used);
- SDA (Signal DAta) is the input and output signal used by master and slave to exchange data.

The drivers using the I2C bus are "open drain", meaning that they can change SDA only from high to low, in other words the system can pull the corresponding signal to low, but not to high. This is very important so the contention between the devices which try to drive the line high, and those which try to pull it low, are eliminated, reducing the potential damage to the drivers or the excessive power dissipation. Each driver is implemented with a pull-up resistor to take the line high when no device drives it to low. Furthermore, thanks to the fact that the devices on the bus don't actually drive the signals high, I2C allows to connect devices with different I/O voltages. In general, in a system where one device is at a higher voltage than another, it may be possible to connect the two devices via I2C without any level shifting circuitry in between them. The trick is to connect the pull-up resistors to the lower of the two voltages. This only works in some cases, where the lower of the two system voltages exceeds the high-to-low-level input voltage of the higher voltage system. In Figure 7.15 and 7.16 are shown

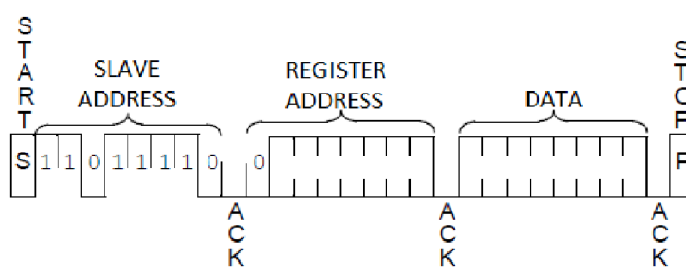


Figure 7.15: *Scheme of the I2C transition state write operation.*

the I2C protocol to write data to the slave and to read data from the slave. Now it will be described the levels, timing and states of SCL and SDA to initiate, implement and finish a write or read operation.

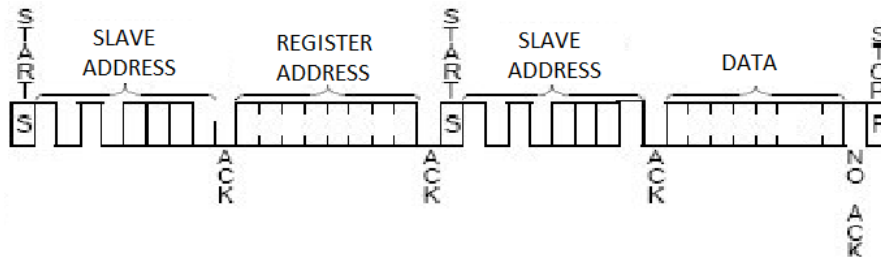


Figure 7.16: *Scheme of the I2C transition state read operation.*

- Idle state: this is the state of the I2C bus when no start condition has been implemented, and where SCL and SDA are in high level. The only way to exit from the idle state is by a start condition, because the other states or condition will be ignored;
- Start condition: this condition initiates the transmission of frame from the master to the slave. It can be implemented if the SDA signal, in high level from the idle, do a transition from high to low, while SCL is in high level. In the multiple master case, the master device which makes the Start condition first, wins the possibilities to control the slave or slaves. It is even possible to repeat the Start and so initiate a new frame transmission;
- Address Slave frame: in this case, the master sends to the slave a frame of 8 bits where: the first 7 bits (from the Most Significant Bit to the bit which precedes the Least Significant Bit) are the slave device address, which allows the master to choose to which slave communicates; while the LSB, of the 8-bits frame, is a command bit which tells the slave if the next frame that will be received by it will be a frame writes from the master to the slave (meaning in transmission, in particular the Address Register), or it will be a frame sends from the slave to the master (meaning in reception, in particular the value saved on a register). In the first case the command will be "0" (low level), in the second case "1" (high level).
- Address Register frame: in this case the master sends an 8 bits frame which represents the register in which the data, writes from the master or reads from the slave, is collocated; the transmission is always from the MSB to the LSB.
- Data frame: this is the 8 bits data writes from master to slave or reads from slave to master, dependently by the value of the command bit.
- Acknowledge bit: after every frame transmission from master to slave, there is a bit sends from slave (so the slave takes the SDA control) that determines if the

transmission is gone well or not, telling it by sending a "0" (acknowledge) or a "1" (not acknowledge).

- Stop condition: after the data frame has been read from the slave or it has been written to the slave, and the acknowledge bit has been sent (only in case of write command), there is a Stop condition, implemented by a transition from "0" to "1" of SDA with SCL in high level. In case of read command, there isn't an Acknowledge bit sends from the slave before the Stop condition, but from the master a "1" is sent to the slave, or in other words a "not acknowledge" value from the master to the slave.

So, at the end, the sequence to write to the slave, or read from it, are shown in Table 7.1. A device programmable by I2C samples data only in the high value of the SCL clock, so

Write Operation	Read Operation
Idle	Idle
Start	Start
Address Slave frame (with command="0")	Address Slave frame (with command="0")
Acknowledge bit	Acknowledge bit
Address Register frame	Address Register frame
Acknowledge bit	Acknowledge bit
Data frame	Start
Acknowledge bit	Address Slave frame (with command="1")
Stop	Acknowledge bit
	Data frame
	Not Acknowledge (from the master)
	Stop

Table 7.1: *Scheme of the state machine of the I2c protocol.*

the bit changing during transmission must be during the low value of the SCL clock. Even during the reception from the slave the data are established by the value corresponding at high value of the SCL clock. There are other types of possible methods to write or read from the master:

- Page write: the Address Slave frame, Address Register frame and the first Data frame are transmitted to the slave receiver in the same way as in a normal register

write sequence. But instead of generating a Stop condition, the master transmits up to 127 (depending by the device) additional bytes, which are temporarily stored in the on-chip page buffer and will be written into memory after the master transmitted a Stop condition;

- Sequential read: the initial sequence is in the same way as a normal read sequence, except that after the slave transmits the first data byte, the master issues an Acknowledge bit as opposed to the not Acknowledge used in a normal read case. This Acknowledge bit directs the slave to transmit the next sequentially addressed 8 bits word. Following the final byte transmitted to the master, the latter will not generate an Acknowledge bit, but will generate a not Acknowledge bit.

In every one of the 2 cases just described, the possibility to transmit or receive data without the assignment of the Address register is possible thanks to an Address counter inside the slave, which increments by one every time one data is sent or received.

7.8 FM-S14

The FM-S14 Quad SFP/SFP+ FMC HPC module is a particular device that allows: the simultaneously transmission and reception via 4 optical transceiver, to generate any 2 frequency between 15.48 MHz up to 1300 MHz programming via I2C 2 clock generators (Quad-Frequency Programmable XO), to use 4 LED integrated, to program via I2C an EEPROM. The 2 clock generators have each one 24 registers programmable and they are setted by default with a series of 4 frequencies selectable by a switch on the top of the mezzanine. The clock generators use a 114.285 MHz crystal oscillator as seed to generate the frequencies, and have a phase jitter of 1 ps.

7.9 RD53a

The RD53A pixel readout integrated circuit is a project of the RD53 Collaboration, an international group of some Universities. This chip has been developed for the ATLAS and CMS upgrade of the 2024 (phase 2 of LHC), in particular for the upgrade of the Pixel Detectors. This device is 20.0 mm by 11.8 mm, which is built with 65 nm CMOS technology, including a pixel matrix of 400 pixels by 192 pixels.

7.9.1 Floorplan and organization

Figure 7.17 and 7.18 show the RD53A layout. This ASIC is a 9 metal layer stack, with in addition 28K AP layer for power lines distribution. The sensitive part of this chip is on the top of it and is composed by 192 by 400 pixels of $50\ \mu\text{m} \times 50\ \mu\text{m}$. The top

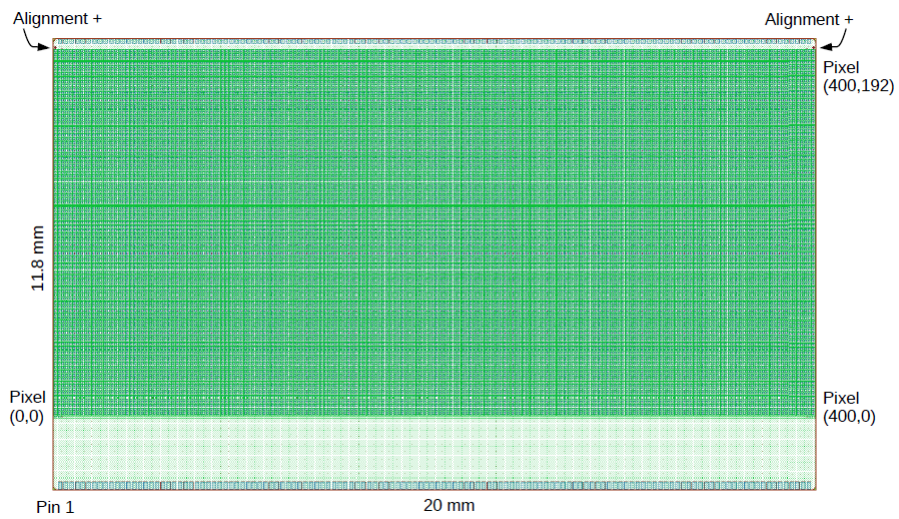


Figure 7.17: *Scheme of the RD53a upview.*

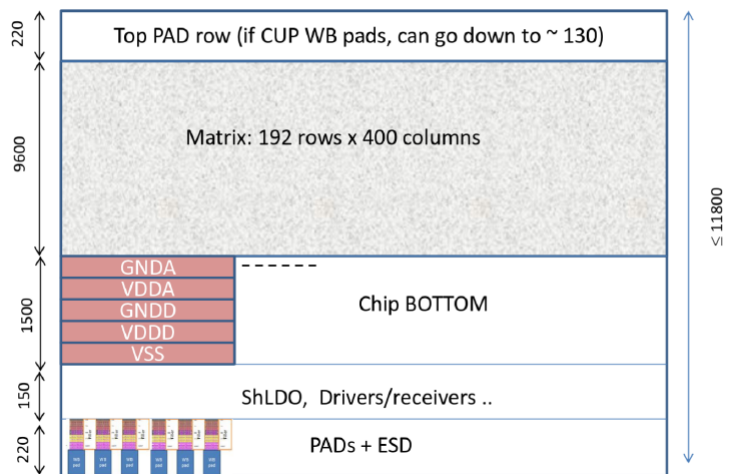


Figure 7.18: *Scheme of lateral view of the RD53a.*

row is now a test pads for debugging purpose, which will be removed in the final chip configuration, while the bottom part contains all the global analog and digital circuitry needs to bias, configure, monitor and read-out the chip. The pixel matrix is built up as 8 x 8 pixel cores, where each 4 are placed in the "analog island" (16 total), where all these islands are embedded in a flat digital synthesized "sea". In the chip periphery all the analog building blocks are grouped in macro-block called Analog Chip Bottom (ACB), for analog environment, and each one of them are surrounded by a synthesized block called Digital Chip Bottom (DCB), which is built to implement the input, output and configuration digital logic.

7.9.2 Analog Front End

In this moment RD53A contains 3 different front end designs to study and achieve the highest performance possible. They are the BGPV (stands for INFN Bergamo and Pavia), LBNL (stands for Lawrence Berkeley National Lab), and TO (stands for INFN Torino). They share some common design features to allow an easy interchangeable on the pixel matrix layout. The analog "quad" layout area is $70\ \mu\text{m} \times 70\ \mu\text{m}$ and contains 4 front ends and 4 bump pads on a $50\ \mu\text{m} \times 50\ \mu\text{m}$ grid. Another common component is the calibration injection circuit, important to direct performance comparisons. Furthermore the bias distribution configuration is the same for all 3.

7.9.3 Digital Matrix

The pixel matrix is built up of digital cores with 8 by 8 pixel channels each contained in 4 by 4 analog islands. One core is one digital circuit. It provides static configuration bits to the analog islands and receives 4 binary outputs from each island. The digital core handles all processing of the binary outputs, including masking, digital injection, Time over Threshold (ToT) counting, storage of ToT values, latency timing, triggering and readout.

7.9.4 I/O and Configuration

To communicate, the RD53a uses different protocols for the input, the output and the configuration. The output consists in 2 ports sending data in protocol Aurora 64b/66b, with different speed rates: 1 lane at 5.12 Gb/s, 4 lanes at 1.28 Gb/s (each lane). For the protocol used for the input and the configuration, see [1].

Bibliography

- [1] RD53 Group, *The RD53A Integrated Circuit*. CERN-RD53-xx-xx-xx, 30 January 2017
- [2] Xilinx, *KC705 Evaluation Board for the Kintex-7 FPGA, User Guide*. ug810, 8 July 2016
- [3] Xilinx, *ZC702 Evaluation Board for the Zynq-7000 FPGA, User Guide*. ug850, 4 September 2015
- [4] Xilinx, *Aurora 64b/66b Protocol Specification*. sp011, 1 October 2014
- [5] Xilinx, *Aurora 64b/66b v11.1 LogiCore IP Product Guide*. pg074, 5 October 2016
- [6] FE-I4 Collaboration, *The FE-I4 Integrated Circuit Guide v2.3*. 30 December 2012
- [7] Faster Technology, *FM-S14 User Manual*. 5 November 2014
- [8] IDT, *Quad Programmable Frequency XO IDT8N4Q001*.
- [9] Xilinx, *Integrated Bit Error Rate Tester 7 series GTX Transceiver v3.0 LogiCore IP Product Guide*. pg132, 8 June 2016
- [10] Xilinx, *Microblaze Processor Reference Guide*. ug 984, 2 April 2014
- [11] Xilinx, *7 Series FPGAs GTX/GTH Transceivers*. ug476, 19 December 2016
- [12] Xilinx, *AXI Reference Guide*. ug761, 19 January 2012
- [13] Xilinx, *7 Series FPGAs SelectIO Resources*. ug471, 27 September 2016
- [14] CERN, *The ATLAS Inner Detector Commissioning and Calibration*. 7 June 2010
- [15] CERN, *Electron Performance Measurements with the ATLAS Detector using the 2010 proton-proton Collision Data*. CERN-PH-EP-2011-117 Eur. Phys. J.C., 21 March 2012

- [16] CERN, *Performance of the ATLAS Track Reconstruction Algorithms in Dense Environments in LHC Run 2*. CERN-EP-2017-045 Eur. Phys. J.C..26 April 2017
- [17] Stefania Stucci, on behalf of the ATLAS Collaboration, *Optical Links for the ATLAS Pixel Detector*. ATL-INDET-PROC-2015-005, 1 July 2015
- [18] Leonardo Rossi, Peter Fischer, Tilman Rohe, Norbert Wermes, *Pixel Detectors: From Fundamentals To Application*. Springer, 24 January 2005
- [19] ATLAS LARG Unit, *Liquid Argon Calorimeter Technical Design Report*. CERN/LHCC 96-41, 15 December 1996
- [20] Jonatham Valdez, Jared Becker, *Understanding the I2C Bus*. Texas Instruments SLVA704, June 2015
- [21] Jochem Snuverink, *The ATLAS Muon Spectrometer: Commissioning and Tracking*. University of Twente, Enschede the Netherlands, 16 October 2009
- [22] M. Wensing, B. Bergenthal, T. Flick, A. Kugel, N. Schroer, *IBL BOC Design and Firmware Manual*.
- [23] Gabriele Balbi, Davide Falchieri, Riccardo Travaglini, Alessandro Gabrielli, Luca Lama, Samuele Zannoli, *IBL ROD Board rev C Reference Manual*. November 2012
- [24] Jonathan Butterworth, John Lane, Martin Postranecky, Matthew Warren, *TTC Interface Module for ATLAS Read-Out Electronics: Final Production Version based on Xilinx FPGA Devices*.
- [25] ATLAS IBL Community, *Insertable B-Layer Technical Design Report*. ATLAS TDR 19, CERN/LHCC 2010-013, 15 September 2010
- [26] S. Baron, J. Mendez, *GBT FPGA User Guide*. 13 April 2016
- [27] Béjar Alonso I. et al., *High-Luminosity Large Hadron Collider (HL-LHC) Preliminary Design Report*. CERN-2015-005, 17 December 2015