

Contribution to the development of the acquisition electronics for the LHCb experiment

THÈSE N° 3054 (2004)

PRÉSENTÉE A LA FACULTE SCIENCES DE BASE

Institut de physique de l'énergie et des particules

SECTION DE PHYSIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Guido HAEFELI

Dipl. Phil-nat., Université de Berne

de nationalité suisse et originaire de Mümliswil-Ramiswil (SO)

acceptée sur proposition du jury:

Prof. A. Bay, directeur de thèse

Dr. H. Dijkstra, rapporteur

Dr. R. Jacobsson, rapporteur

Prof. U. Straumann, rapporteur

Prof. M. Q. Tran, rapporteur

Lausanne, EPFL

2004

Acknowledgements

I would like to express all my gratitude to Professor Aurelio Bay for his extraordinary support as an expert and colleague during the past five years. His expert guidance and his genuine and sincere support have made all of this possible.

Special thanks for their valuable help and contributions go to my former colleague Yuri Ermoline, the co-workers Federica Legger, Laurent Locatelli, Patrick Koppenburg and the colleagues from the LPHE. Special thanks also go to Raymond Frei, Guy Masson from the electronics and Jean-Philippe Hertig from the mechanics workshop as well as to the secretaries Erika Luthi, Monique Romaniszin and Esther Hofmann. Thanks as well to for the computer support from Japhet Bagilishya.

Among many of my LHCb colleagues I would like to thank to Jorgen Christiansen, Manfred Mücke, Pawel Jalocha and Tatsuya Nakada, the people involved in the DAQ and ECS Clara Gasper, Benjamin Gaidioz, Beat Jost, and especially to Niko Neufeld for the countless debugging sessions he always made to become a pleasure. Thanks to the electronics designers François Bal, Hans Muller, Jean-Michel Sainson, Pascal Vulliez, Angel Guirao, Rui Pimenta at Cern. Thanks for the contributions from Dirk Wiedner, Ulrich Uwer, Achim Vollhardt, Bernhard Spaan, Rainer Schwierz, Cyril Drancourt, Pierre-Yves David, Nicolas Dumont-Dayot, Daniel Boget Thomas Ruf, Massimiliano Ferro-Luzzi. Special thanks also to my colleagues from the Tsinghua University Alex Gong, Hui Gong and Beibei Shaw.

Last but not least I wish to thank the jury of the thesis Benoît Deveaud-Plédran, Hans Dijkstra, Richard Jacobsson, Ulrich Straumann and Minh Quang Tran.

Thanks to all these people for the professional experience and pleasant moments we have shared.

Finally, I would like to express my love and gratitude to my wife Céline, my two children Robyn and Raphaël, my family and friends for their everlasting support and encouragement.

Lausanne 24. August 2004

Abstract

The LHCb experiment is one of the four large particle detectors currently under construction at the LHC accelerator at CERN. It is a forward single-arm spectrometer dedicated to precision measurements of CP violation and rare decays in the b quark sector.

In the Standard Model CP violation arises via the complex phase of the 3x3 CKM quark mixing matrix. The LHCb experiment will test the unitarity of this matrix by measuring in several theoretically unrelated ways all angles and all sides of the unitarity triangle. This will allow to over-constrain the model and - hopefully - to exhibit inconsistencies which will be a signal of physics beyond the Standard Model.

The LHCb detector consists of roughly one million sensors and is read out every LHC bunch crossing at 40 MHz. In the subsequent selection of events a multilevel trigger scheme is applied. The data is required to reside in the radiation environment on the front-end chips until the first level trigger (L0) decision is taken. For the second level trigger (L1) processing, the data is transmitted over long analog copper or digital optical links to the data acquisition board called TELL1 (Trigger ELectronics Level 1 board).

TELL1 is now used by essentially all sub-detectors of LHCb. It provides the interface to the copper and optical link systems and performs intensive processing. This includes event synchronization, link compensation for the analog readout, pedestal calculation and subtraction, common mode suppression, zero suppression, L1 buffering, multi event packaging, and encapsulation into IP compliant Ethernet packets. The output of the board is sent to the Gigabit based event builder network and processed on the the combined L1 and High Level Trigger CPU farm.

The data rate of 30 Gbit/s on the input of TELL1 can be managed using large FPGAs, highest density DDR SDRAM and fast PCB interconnects. In this document a proposal for the processing steps is given, the Level 1 buffer implementation is discussed, and design consideration concerning signal integrity are made.

Version abrégée

LHCb est une des 4 grandes expériences en construction, qui seront installées sur le Large Hadron Collider (LHC) du CERN. LHCb est un spectromètre à un bras qui est consacré à la mesure de précision du phénomène de la violation de CP ainsi qu'à la mesure de désintégrations rares dans le secteur des particules avec un quark b. Dans le Modèle Standard des particules, la violation de CP est engendrée par une phase complexe de la matrice de mélange des saveurs, CKM. LHCb vérifiera l'unitarité de la matrice CKM par différentes méthodes théoriquement uncorrélées. Cela permettra de tester la théorie et, peut-être, d'en montrer les contradictions, ce qui sera un signal de physique au-delà du Modèle Standard.

Le détecteur LHCb est constitué de un millions de senseurs qui seront lus à chaque collision du LHC, soit chaque 25 ns. Un " trigger " à 3 niveaux (nommés L0, L1 et HLT) va par la suite sélectionner les événements physiquement intéressants. Les données doivent être stockées dans les chips de " front-end " pendant que le trigger de premier niveau, L0, calcule sa décision. Pour le niveau L1, les données sont transportées vers une région sans radiation par des câbles optiques ou électriques, vers les circuits d'acquisition, nommés TELL1 (Trigger ELectronics Level 1 board). Le développement de TELL1 est le sujet de ce travail. TELL1 a été adopté par tous les sous-détecteurs de LHCb (excepté le RICH). Il va fournir l'interface optique ou électrique vers le front-end. Il sera capable d'effectuer des calculs compliqués en mode pipeline: synchronisation, filtrage des signaux, calcul et soustraction des piédestaux, correction pour le bruit commun, suppression des zéros, stockage pendant le second niveau de trigger, regroupement en une structure à plusieurs événements et encapsulation pour transmission sur réseau Ethernet. Les données à la sortie du TELL1 sont par la suite transmises à un réseau Gigabit Ethernet pour qu'elles soient assemblées en événements complets et transmises aux processeurs de trigger L1 et HLT.

Le taux des données de 30 Gbit/s à l'entrée de chaque TELL1 peut être géré par l'utilisation de FPGA de grande taille, de DDR SDRAM de la plus haute densité disponible sur le marché, ainsi que de techniques d'interconnexion rapide sur circuit imprimé. Dans ce document nous proposons une séquence d'opérations à réaliser dans le TELL1 ainsi que l'implémentation de la mémoire du niveau L1. Nous parlons aussi des précautions dans le design, pour garantir l'intégrité de la propagation des signaux rapides.

Contents

Introduction	1
1 Experimental apparatus: LHCb at LHC	5
1.1 Accelerators and the LHC	5
1.1.1 The Large Hadron Collider	6
1.1.2 The LHC - Challenges	7
1.2 The LHCb detector	9
1.2.1 The Vertex Locator VeLo	10
1.2.2 RICH	11
1.2.3 The magnet	12
1.2.4 The tracker	12
1.2.5 The calorimeters	13
1.2.6 The muon detector	13
1.2.7 Read-out, Data Acquisition and Triggering at LHCb	13
1.2.8 The Level 1 electronics and the TELL1 board	14
2 The trigger and the data acquisition systems	17
2.1 Why do we need a trigger system?	17
2.1.1 The LHCb trigger from the physics point of view	19
2.2 Overview of the data acquisition	20
3 Data processing for the VeLo: the TELL1	23
3.1 The VeLo front end. The Beetles.	24
3.2 Input data processing for the VeLo	26
3.2.1 VeLo link synchronization	26
3.2.2 The FIR filter	30
3.2.3 Pedestal calculation and subtraction	33
3.2.4 Channel reordering	34
3.2.5 Other processing options	38
3.3 Data processing for the L1 trigger	39
3.3.1 L1 Trigger Common Mode Suppression (L1T-CMS)	39
3.3.2 L1T Data zero suppression (sparsification)	49
3.3.3 L1 buffering	50
3.3.4 L1T Data Linking	54
3.4 Data processing for the HLT	57
3.4.1 HLT Data Linking	57

3.4.2	HLT Common Mode Suppression	58
3.4.3	HLT Zero Suppression	59
3.5	Multi Event Packing (MEP)	60
3.6	L0 and L1 throttling (buffer overflow prevention)	62
3.6.1	L0 throttle	62
3.6.2	L1 throttle	63
4	TELL1 for optical readout	65
4.1	Input data processing for the optical readout	65
4.2	L1 Trigger Common Mode Suppression for IT and TT	66
4.3	L1T Data Linking for OT	67
4.4	HLT readout	67
5	The development of TELL1	69
5.1	The ancestors of TELL1	69
5.1.1	RB1	69
5.1.2	RB2	69
5.1.3	RB3	71
5.2	TELL1	72
5.2.1	Analog receiver card (A-RxCard)	73
5.2.2	Optical receiver card (O-RxCard)	73
5.2.3	Event builder Network interface (GBE RO-TxCard)	73
5.2.4	ECS interface (CCPC and Glue Card)	74
5.2.5	FEM	75
5.3	The Signal Integrity (SI) problems	76
5.3.1	The PCB	77
5.3.2	Termination	79
5.3.3	Parallel termination for DDR SDRAM address bus	81
5.3.4	Point-to-point termination for DDR SDRAM data bus	83
5.3.5	QDR bus signals	84
5.3.6	Inter FPGA connects for data linking	86
5.3.7	Driving strength control for SPI-3 interface to TxCard	87
5.3.8	ECS bus “Local bus”	87
5.4	PCB Routing	89
5.5	L1 Buffer implementation studies	94
5.5.1	Principle of operation of the L1 buffer	94
5.5.2	Data access characterization	94
5.5.3	Buffer size	95
5.5.4	Memory interface	96
	Conclusion	97
	A The Zoo of memories	99
	B Introduction to processing implementation techniques with FPGA	103
	Abbreviations	109

Introduction

For centuries man has been interested in understanding the world around him. In High Energy Physics (HEP) we try to find answers to the questions: What are the building blocks of matter, the “elementary particles”? How do they interact? In the field of Astrophysics and Cosmology we look for answers to questions such as how did the Universe evolve and what will be its (our) destiny?

Although these sciences look at very different scales, from the smallest to the largest structures in Nature, there is still a common philosophical basis and a large share of “physics”. We are not to forget that Astronomy is one of the pillars on which modern physics science is standing.

Currently it is believed that the fundamental building blocks of matter consist of 6 quarks (u, d, c, s, t, b) and three leptons with associated neutrinos ($e, \mu, \tau, \nu_e, \nu_\mu, \nu_\tau$). For all quarks and electrons there are corresponding anti-particles, the mirror images of the particles. Anti-particles were predicted by P. A. M. Dirac in his attempt to marry Quantum Mechanics and Relativity, leading to the famous Dirac equation. The current theories on particle interactions describe the different forces between particles by the exchange of specific “messenger particles”: the photon is the mediator of the electromagnetic force, the gluons are responsible of the “strong” force, and so on. The sum of our knowledge is collected in the so called “Standard Model of Particles” (SM).

At the beginning of the twentieth century Hubble discovered that the Universe is expanding. If the direction of time would be reversed, everything in our Universe would collapse in one point known as the Big Bang - the beginning of time. Models describing the Big Bang assume that the early stage of the Universe was compact and hot. With the precise knowledge of the fundamental particle interactions in such an environment, the subsequent evolution of the Universe could be predicted and compared with the observations. The predictions are a Universe containing a large amount of photons together with an equal amount of matter and anti-matter particles. The observation of the Cosmic Microwave Background (predicted in 1940 by Gamow, observed in 1965 by two engineers of the Bell laboratories, A. Penzias and R. Wilson), a very isotropic radio-wave Planck spectrum corresponding to a black body temperature of 2.726 K, shows that there are indeed a huge amount of photons present in the Universe. There also is a sizeable amount of matter (us, for instance) but in a ratio of about 1 billion photons per matter particle. On the other hand no anti-matter is observed (the amount of positron, anti-protons, etc., which are observed in cosmic rays can be understood as secondary production, and not of cosmological origin). This observation implies that there is a matter/anti-matter asymmetry in nature. The source of this asymmetry has been an intriguing question far many years. In 1967 A. Sakharov [2] wrote down three conditions needed to explain

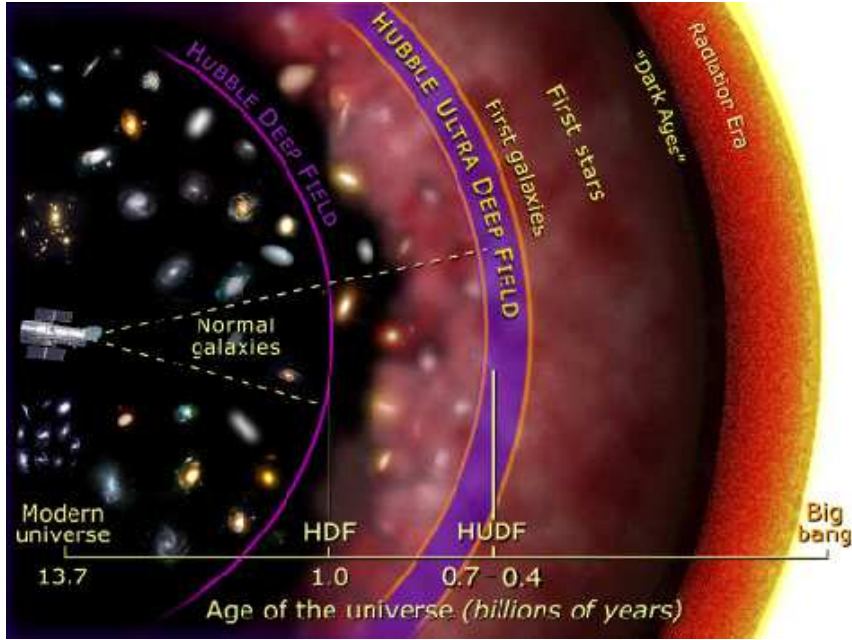


Figure 1: The Hubble Space Telescope (HST) can probe the history of the Universe when it was about half a billion years old. Called the Hubble Ultra Deep Field (HUDF), the million-second-long exposure reveals the first galaxies to emerge from the so-called "dark ages", the time after the Big Bang when the first stars reheated the cold, dark Universe (illustration taken from [1]). The Dark Ages and beyond can be explored by other means (like the COBE experiment). High Energy Physics and Cosmology work together to shed light on the first seconds of the Universe.

the cosmic abundance of matter. One of these three requirements is the existence of "CP violation". Here "P" stands for Parity, the symmetry operation which corresponds to space reflection and "C" is the Charge conjugation, the symmetry between particles and anti-particles. The Dirac theory, for instance, is invariant under the CP symmetry. In Dirac's Universe there is no room for CP violation, and an equal amount of matter and anti-matter (within statistical fluctuations) should exist at any time. On the other hand, a clear CP violation (of the order of 10^{-3}) was observed in 1964 in the decay of the "neutral kaon" K_L^0 . This particle (which is its own antiparticle) has the choice to decay into two CP related channels, one with an electron and the other with a positron. It was found that the channel with a positron was more frequent. The asymmetry was $\frac{N(K_L^0 \rightarrow \pi^- e^+ \nu) - N(K_L^0 \rightarrow \pi^+ e^- \bar{\nu})}{N(K_L^0 \rightarrow \pi^- e^+ \nu) + N(K_L^0 \rightarrow \pi^+ e^- \bar{\nu})} \approx 3 \times 10^{-3}$. This result provides an **absolute definition for the sign of the electric charge** in the Universe (or at least in the part of the Universe reachable by our observation) and qualitatively satisfies one of the criteria of Sakharov.

Over the last 40 years the Standard Model theory has been able to describe with great precision (often at the level of 10^{-8}) the observed phenomena in electro-magnetic, strong and weak interactions. In all these years many high energy physics experiments have tested the Standard Model for inconsistencies. None has been found so far. The Cabibbo-Kobayashi-Maskawa (CKM) mechanism within the Standard Model provides

a first step toward an explanation of CP violation. The CKM matrix parameterizes the mixing between quarks and is the mathematical link to the Higgs mass generation mechanism in the fermionic sector. The possibility to introduce a complex phase in the CKM matrix allows the SM to introduce some amount of CP violation, which well corresponds to observations. From the history of Science point of view it is very important to remember that the CKM mechanism invoked in 1972 by Kobayashi and Maskawa to explain the CP violation led to the prediction of the existence of the third family of quark (t and b). Indeed no complex phase can exist with only 4 quarks. The b was discovered in 1977 and the t in 1994. The CKM mechanism of CP violation has been verified at a few percent level by experiments with Kaons (in particular at CPLEAR) and more recently by the BaBar and BELLE experiments with neutral B's (mesons containing a b quark).

The Standard Model is not a complete theory (that's why we just call it a "model"). First of all gravity is ignored, and there at least 18 free parameters (more if the neutrino masses and mixing are included). The set of parameters includes the masses of all the particles and the 4 parameters of the CKM matrix. Several theories are on the market to increase the predictivity of the model. Each has its prediction of particular observables. For instance supersymmetric theories (SUSY) double the number of particles: for each known particle we should observe a supersymmetric partner. None has been observed so far!... We hope that some "New Physics" will be discovered in the near future, to guide the fantasy of our colleagues theorists.

Beside the possible direct observation of new exotic particles, one way to probe New Physics is to carry out very high precision measurements of specific processes, believed to be affected by the presence of New Physics. In this spirit an intense campaign of measurement is carried out to over-constrain the CKM matrix, in particular by testing its predictivity in the domain of CP violation. New Physics should alter the amount of CP violation and this will appear as deviations in precision measurements of CP violation processes.

The conclusions of these measurements will give new input to Cosmology. It is believed that the amount of CP violation of the Standard Model is not enough to explain the observed asymmetry in the Universe [3]. Hence the study of the CKM matrix plays a central role in our attempts to discover the origin of our Universe.

More in detail, the CKM matrix is a complex matrix with 3×3 elements (3 is the number of families of quarks). In the SM the matrix is unitary, to conserve probability of transitions inside the 3 families. Incidentally a deviation from unitarity could mean that a new quark family is hiding somewhere, for instance.

Today, the CKM elements are known to various levels of precision. Large uncertainties remain in the sector which involves processes with b quarks. Hence B meson decays play an important role. Theoretical calculations for decays of B mesons within the SM framework predict large CP violation effects. As previously stated this phenomenon has indeed been observed by BaBar and BELLE. We are now interested in pushing the precision of the measurement to a level at which hopefully the SM predictions will fail. This will be the smoking gun which will prove that some kind of New Physics is active.

CERN is now building a new machine, the Large Hadron Collider (LHC), as a replacement of the LEP. The LHC is a proton-proton collider with a 14 TeV energy in the center of mass. At this energy, one of every 160 interactions is expected to produce B particles, from which typically one in 100,000 decays exhibits possible CP violation.

The unstable B-particles have a short life time and travel a distance of a few centimeters before decaying into stable particles (the mean life of a neutral B is about 1.5×10^{-12} s, but in their fast moving frame this time expands by a factor of 10-100 because of Lorentz time dilation). By combining the properties of the observable final state particles, the kinematic properties of the initial B-meson can be reconstructed. The CP violation phenomena can subsequently be measured by comparing the decay times spectra of the reconstructed B and anti-B particles for specific decay channels.

How can this be performed in an experiment? The experiment specially optimized for B-decay reconstruction at the LHC is the LHCb detector, for which the research in this thesis has been done. In order to build a challenging experiment as LHCb, many years of R&D are needed. The LHCb collaboration began in August 1994. The preliminary description of the detector is documented in the Technical Proposal (1998) [4]. The project has matured over the years and the optimized LHCb detector is described in the ‘‘Detector Design and Performance’’ Technical Design Report [5] (TDR). Moreover each sub-system is the subject of a specific TDR[6]. After 10 years of work by hundreds of physicists and engineers, LHCb is now in the construction phase. We are all making a maximum effort allow its operation to start in 2007, together with the first LHC beam collision.

In modern HEP experiments a central role is played by the read-out electronics and data-acquisition sub-systems. In short: the particles leave an electric signal in the sensors (solid-state, scintillators, gas detectors,...). Then we need an the electronic circuit to collect the analogue signals from the sensors, amplify and digitize them before storage for the subsequent ‘‘off-line’’ analysis. Not all the proton-proton collisions at LHC are of interest. An electronic sub-system called ‘‘trigger’’ is in charge of filtering the events which are worth saving. This thesis is dedicated to the development of a part of such circuitry for LHCb.

The outline of this thesis is as follows:

Chapter 1 is an introduction to the experimental setup, which includes the LHC accelerator and the LHCb experiment. Chapter 2 pays special attention to the description of the LHCb trigger system. Chapter 3 describes the data processing implemented on TELL1 with respect to the VeLo. Chapter 4 discusses the adaption required for other sub-detectors making use of the TELL1. Chapter 5 gives an overview of the development done in the past and for TELL1.

Chapter 1

Experimental apparatus: LHCb at LHC

Our study of CP violation will be done at CERN. CERN (Convention Européenne de la Recherche Nucléaire) has installed a large physics research center for high energy particle physics in Geneva [7]. The laboratory founded in 1954 by originally 12 European states is counting 20 member states today. The international collaboration was the answer to cope with the ever growing size, complexity and cost of high energy particle physics experiments.

For our measurements we will make use of the Large Hadron Collider (LHC) and of the LHCb experiment. The aim of this chapter is to give an introduction to accelerators, the LHC in particular, and to LHCb.

1.1 Accelerators and the LHC

Accelerators for high energy particle physics experiments are large facilities to accelerate and therefore increase the kinetic energy of a particle. Why do we need to accelerate particles? There are two parallel reasons. First of all the equivalence of kinetic energy and the mass of a particle can be used in a fundamental way. To create heavy particle in the LEP experiment, electrons and positrons have been accelerated up to 100 GeV each and created particles much more massive than the originally accelerated ones. For example the W production in the reaction

$$e^+ + e^- \rightarrow W^+W^- \quad (1.1)$$

shows the production of the 160000 times more massive particle compared to the mass of the electron. The second aspect can be understood in terms of the capability of a particle to probe the structure of a target. Like in a microscope, the spatial resolution increases when the wavelength becomes smaller. Following de Broglie, the resolution power of a particle probe with momentum p is of the order of h/p (h is the Planck constant). We might say that a 1 TeV electron can probe structures of the order of 10^{-4} fm and to time domains of the order of 10^{-12} fs.

To create more massive particles and explore even smaller space-time regions, more energy is needed and this asks for a continuous innovation on the accelerator design.

1.1.1 The Large Hadron Collider

As previously said, the next generation accelerator that is constructed in the tunnel of the LEP experiment is the Large Hadron Collider (LHC). With a proton-proton collision

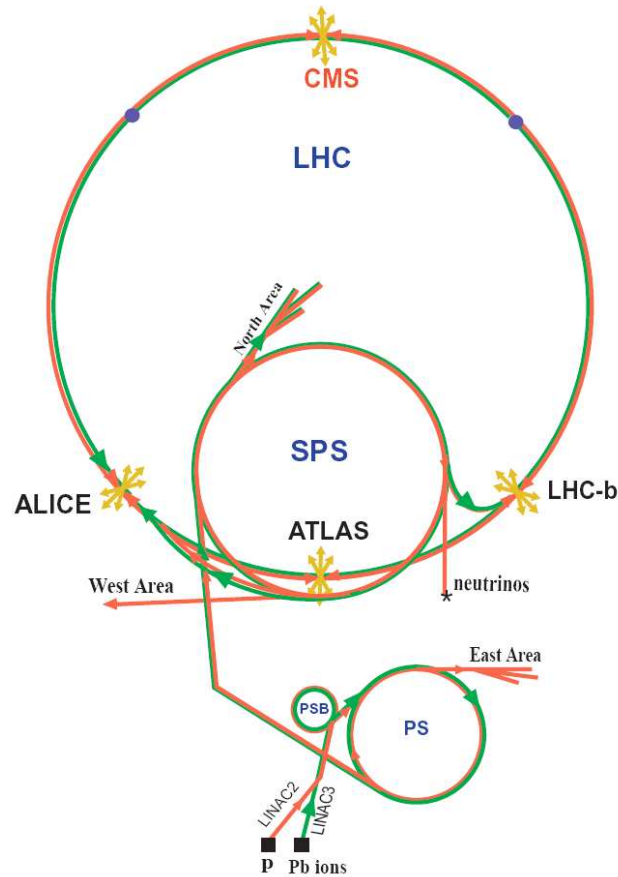


Figure 1.1: *The LHC accelerator complex. Picture taken from [8].*

at a rate of 40 MHz, a center of mass energy of 14 TeV¹ and a nominal luminosity² of $1 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$. In figure 1.1 an overview of the whole LHC accelerator complex is shown. Acceleration of protons starts in one of the linear accelerator (Linac) up to 50 MeV. Then two circular accelerators boost the particles energy up to 1 GeV (Booster) and 26 GeV (PS) before the particles enter the Super Proton Synchrotron (SPS) and enter the LHC with 450 GeV energy. Also indicated in the figure are the location of the detectors using the LHC beam. The ATLAS and CMS experiments located at Interaction Points IP1 and IP5 are general-purpose detectors covering a wide range of physics. In particular they are optimized to detect the Higgs particle as well as the supersymmetric partners of the SM particles. These are the experiments demanding for the highest possible luminosity. At

¹Due to the fact that the proton is build of 3 quarks, the effective energy available in a quark - quark collision is reduced by the fraction of energy carried by each quark, a quantum variable with an average which is roughly 1/3 of the total energy.

²The “luminosity” is a measure of the rate of collisions. The luminosity multiplied by the cross section of a process gives the measurable rate of the process.

IP2, the ALICE detector is designed to study quark gluon plasma in heavy ion collisions. Indeed the LHC can also be used as a heavy ion collider producing Pb-Pb collisions, for instance. The LHCb experiment is located at IP8 and is dedicated to study CP violation in the B mesons systems. It will use only a fraction of the available luminosity of LHC.

1.1.2 The LHC - Challenges

In the LHC [9] the energy available in the collisions between the constituents of the protons (the quarks and gluons) will reach 7 TeV, that is about 10 times that of LEP and the Fermilab Tevatron. In order to maintain an equally effective physics program at a higher energy E the “luminosity” of a collider should increase in proportion to E^2 . As seen before the de Broglie wavelength associated to a particle decreases like $1/p \sim 1/E$. This allows to increase the resolution power of the system but at the same time the cross section of the particle decreases like $1/E^2$. and the probability of interaction becomes small very fast. The record luminosity in a collider has been obtained with the KEKB machine (instrumented by the BELLE experiment) with values beyond $1 \times 10^{33} \text{cm}^{-2} \text{s}^{-1}$. Nevertheless this machine works at quite low energy (“only” about 10 GeV). The Tevatron p-pbar collider at Fermilab has of the order of $1 \times 10^{32} \text{cm}^{-2} \text{s}^{-1}$. The LHC is designed to reach $1 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$. The luminosity is proportional to the product of the two beams intensities and to the inverse of their transversal section. This LHC design luminosity will be achieved by filling each of the two rings with 2835 bunches of 10^{11} particles each. Only protons will be used to obtain a very high current in both beam (this is not the case of the Tevatron at Fermilab or in the old SPS at CERN, in which a unique ring is used for the circulation of protons and anti-protons). The resulting large beam current ($I_{LHC} = 0.53 \text{ A}$) is a particular challenge in a machine made of superconducting magnets operating at cryogenic temperatures. The two beam pipes needed will be embedded in a single super-conducting magnet with a magnetic field strength of 8.3 T operated at a temperature of 1.9 Kelvin. This strong magnetic field is needed to to keep the particles on the 27 km long circular trajectory during the final acceleration.

There are several phenomena limiting the maximal available luminosity. We will give a short summary of these problems.

The capability to create a very well focused beam at the collision point (where a detector is sitting) is limited by mechanical instabilities. The transversal section of the two beams will be of $17 \mu\text{m}$ r.m.s. at the collision point. This requires a mechanical stability of the focusing magnets of the same order.

One cannot increase the bunch particle density at will because of “beam-beam” interactions. When two bunches cross at the collision point only a tiny fraction of the particles interact in a strong way. All the others are deflected by the electromagnetic field of the opposing bunch. These deflections accumulate turn after turn and may eventually lead to particle loss. The LHC injectors (the PS and SPS) are being refurbished to provide exactly the required beam density, not exceeding the stability limit.

While travelling down the 27 km long LHC beam pipe each of the 2835 proton bunches leaves behind an “electromagnetic wake-field” which perturbs the succeeding bunches. In this way any initial disturbance in the position or energy of a bunch is transmitted to the next and under certain conditions the trajectories are disturbed in such a way that the beam can be lost. Countermeasures have to be taken to suppress or at least reduce these

effects: the electromagnetic behavior of the materials surrounding the beam must be kept under control. Residual instabilities can be corrected by sophisticated feedback systems.

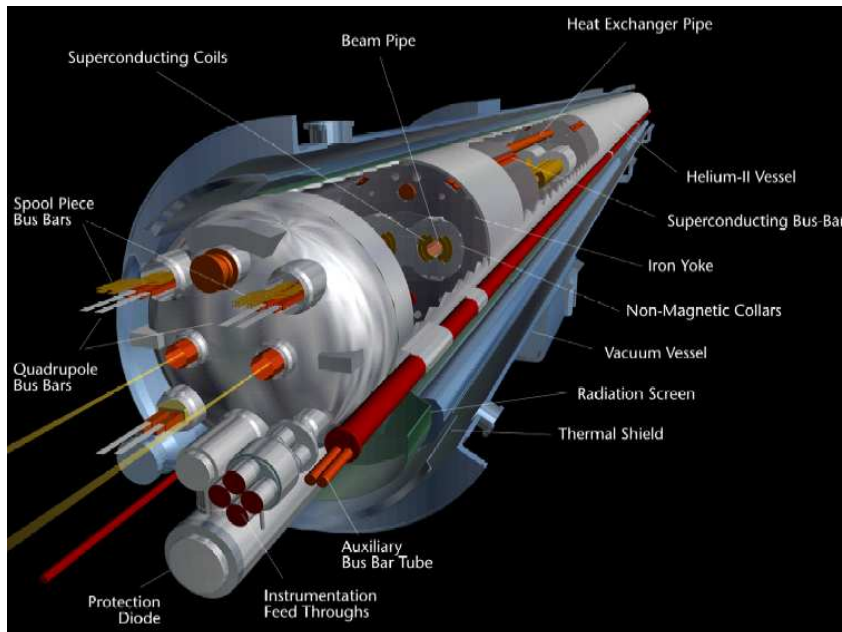


Figure 1.2: *Cross section of the LHC beam pipe. Picture taken from [8].*

In the present acceleration scheme the beams will be stored at high energy for about 10 hours. During this time the particles make four hundred million revolutions around the machine. Small “orbit instabilities” would be amplified leading to a reduced beam lifetime. In addition to beam-beam interactions and wake fields, instabilities could be generated by small non linearities in the deflection and focusing magnets. It is impossible to analytically calculate the effects of such non linear effects. Transport equations are encoded in computer programs for tracking the particles for up to 1 million turns. Results are used to define tolerances for the quality of the magnets at the design stage and during production.

A catastrophic total beam lost is not unlikely and diagnostics tools are studied to counteract these events. In any case a small fraction of the beam is lost per turn because of the previously described effects. It is possible that a lost particle energy will convert into heat in one of the superconducting magnets with subsequent quench. To reduce the probability of these events the LHC is equipped with a collimation system to intercept the particles travelling too far from the authorized orbit.

The closed orbit of LHC is achieved by the bending power of electro-magnets. The drawback of this configuration is that the beam losses energy in the bending process, by emitting the so called “synchrotron radiation”³. The power loss is proportional to γ^4/r (γ is the Lorentz factor). In electron machines this effect becomes impossible to treat beyond 100–200 GeV or so, and Linear ACcelerators (LINACs) are considered. For protons the limit is around 100 TeV. However in the LHC the power emitted is already of about

³The synchrotron radiation is now used in solid state physics, crystallography, etc.

3.7 kW. This power has to be efficiently evacuated. The synchrotron U.V. photons hit the beam pipe and can release absorbed molecules, with increase of the beam-pipe gas pressure. The beam-gas interactions can also affect the beam stability.

In conclusion accelerator physics is a very complex domain and particle physicists are in dept with our colleagues who spend their life to bring us fabulous machines like certainly will be the LHC. To conclude, we have to stress that accelerator physics was initially developed by particle physicists. Now accelerators have become widespread tools, used in many fields for fundamental and applied research not only in HEP but also in solid state physics, crystallography, chemistry, . . . They play an important role in industrial and biomedical applications. Beams are used directly or in the production of isotopes which are then used for powerful functional imaging techniques like the Positron Emission Tomography. They also play a role in art and archeology [10]. In the domain of energy production, “Accelerator Driven Systems” (ADS) are now considered as candidates for the treatment of nuclear waste, etc.

1.2 The LHCb detector

The LHCb experiment [4] is designed to study CP violation and other rare phenomena using the copious production at LHC of hadrons with b quarks (B mesons in particular). The $b\bar{b}$ -production in pp-interaction is dominated by fusion processes of gluons and partons (gg and $q\bar{q}$). The simulation of this processes show that at high energies both the b and \bar{b} -particles are predominantly produced in the same forward cone (see figure 1.3). This fact leads to the specific geometry of the LHCb detector. It is a single-arm spectrometer (only one of the two possible forward regions is employed) and has an angular coverage from 10 mrad to 300 (250) mrad in the bending (non-bending) plane.

The beam luminosity will be locally defocused such that the average pp-interaction per bunch crossing is about one. The luminosity needed by LHCb is about $2 \times 10^{32} \text{cm}^{-2} \text{s}^{-1}$ and can be delivered by the LHC machine from the very beginning of operation. Multiple pp-interaction are not desired since they can lead to mis-interpretation by the trigger (section 2.1.1).

The detector layout after the “LHCb light” optimization is given in figure 1.4. To avoid major civil construction the detector has been adapted to the existing experimental hall used by the DELPHI experiment of the LEP era. LHCb is 20 m long and 10 m wide.

The experiment must be capable to efficiently select events in which b quarks are present. As previously said, b hadrons have a quite long flight path, due to their average life combined with Lorentz boost. The detector must be capable to recognize decay

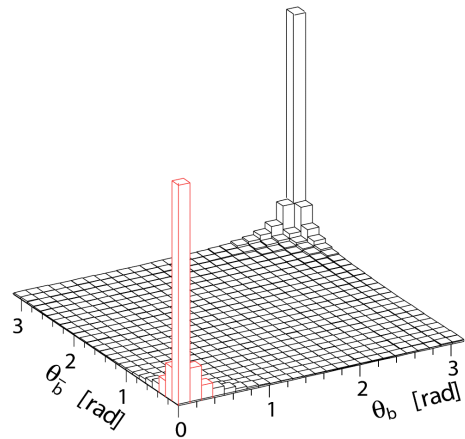


Figure 1.3: Polar angles of the b and \bar{b} calculated by the PYTHIA event generator. Simulation taken from the LHCb technical proposal [4].

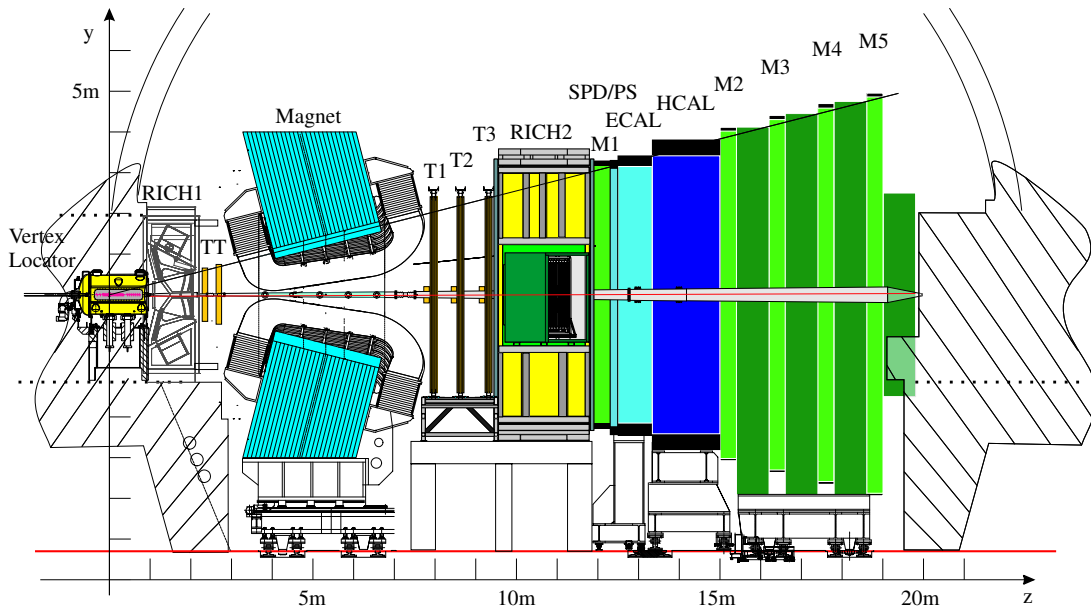


Figure 1.4: Side view of the detector (non-bending plane). Picture taken from [5].

vertexes displaced from the original p-p interaction point. To do that, first the p-p interaction point must be reconstructed from prompt tracks, then the presence of a b hadron can be tagged by a secondary vertex. This task will be performed by the VERticeX LOcator (VeLo), a solid-state detector. The momentum information of charged particles will be reconstructed by the magnetic spectrometer: a dipole magnet and 4 tracking stations. LHCb has a calorimetric detector and muon chambers. Of great importance is the particle identification based on RICH. In addition the experiment provides the readout-electronics and data-acquisition system and a trigger. We give in the following some details of each sub-system.

1.2.1 The Vertex Locator VeLo

The LHCb “VeLo” vertex detector is a silicon strip detector that provides precise measurements of the track coordinates of charged particles close to the interaction region [11]. It consists of 21 stations arranged perpendicular to the beam and is the only detector providing information about the tracks in the backward direction. To minimize the material between the interaction and the detectors, the beam pipe is replaced with a thin aluminum box (see figure 1.5). The box provides the separation between the beam vacuum and the vacuum of the detector container as well as a shield against RF pick up from the beam. The small radial distance from the beam requires that the sensors can be retracted during the LHC beam injection phase. The assembled detector is shown in figure 1.6. Each station of the VeLo measures the r and ϕ position of traversing charged particle. Using single sided silicon sensors, a station is build of two silicon planes separated 2 mm from each other. The total number of channels to be read out is about 200'000.

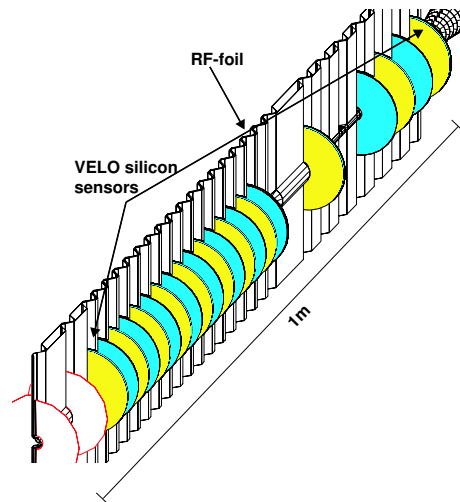


Figure 1.5: Arrangement of the detectors along the beam axis. Only the silicon sensors on one side of the RF-foil, which separates the LHC vacuum from the detector vacuum, is shown. The first two detectors (unshaded) belong to the Pile-Up system. Picture taken from [5].

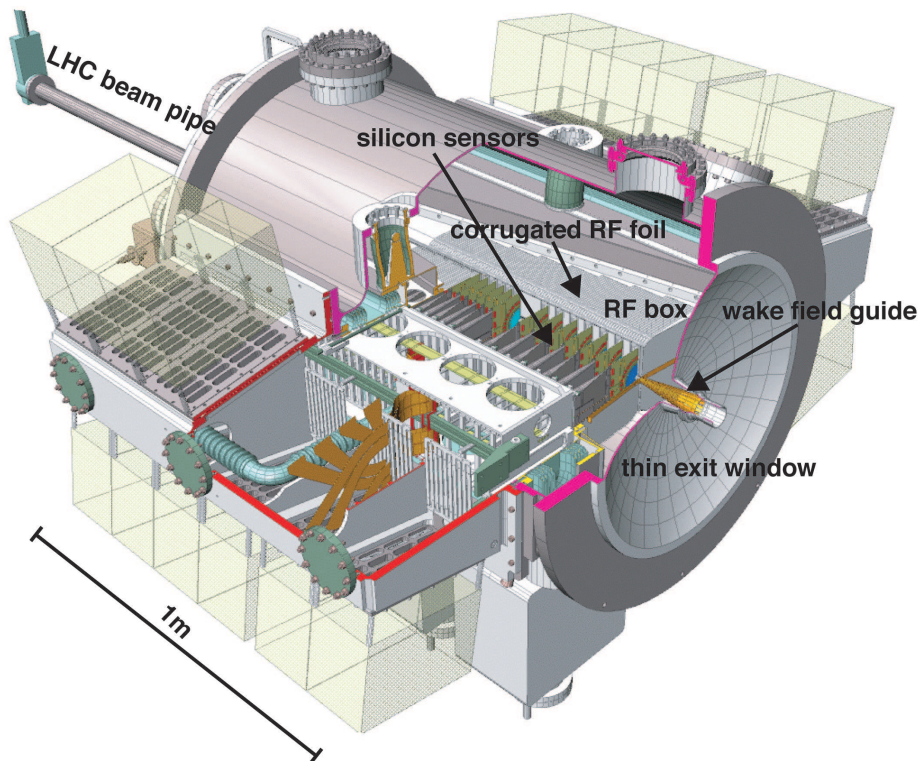


Figure 1.6: The VeLo detector assembled. Picture taken from [5].

1.2.2 RICH

The hadron identification in LHCb is provided by two Ring Imaging Cherenkov detectors (RICH) [12]. The Cherenkov light is produced when charged particles traverse a

transparent medium with a speed larger than the speed of light in that medium (it is analogous to the supersonic shock wave of an object travelling faster than Mach 1). High momentum particles (up to ~ 100 GeV/c) are identified by the RICH2 detectors, situated downstream of the spectrometer magnet and tracking stations. Lower momentum particles, up to about 60 GeV/c, are identified by the RICH1, located upstream of the magnet.

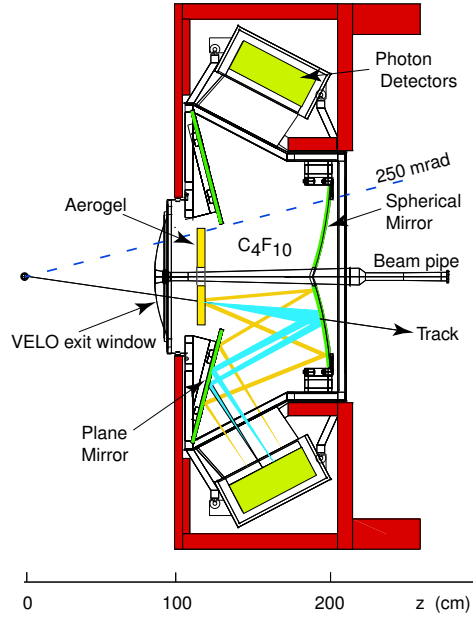


Figure 1.7: *Layout of the vertical RICH1 detector. Picture taken from [5].*

1.2.3 The magnet

A dipole magnet with normal aluminum conductor provides a magnetic field with a maximal strength of 1.1 T [13]. To reduce the power dissipation to about 4.2 MW, the pole faces are shaped to follow the acceptance angles of the experiment. The magnet is made of 50 tons of aluminum conductors and a 1450 tons heavy steel yoke. To compensate for left-right asymmetries, the polarity of the magnetic field can be inverted. The optimized detector has removed the shield plate which was foreseen to protect from the magnetic field the RICH1. This was done to let the field to penetrate the upstream part of the detector in such a way to allow a rough momentum analysis at the early stage of the triggering process.

1.2.4 The tracker

The tracking system provides the momentum measurement of charged particles and the information to link the hits detected in the calorimeters and the muon chambers [14, 15, 16]. It also provides the seeding information for the RICH. The tracker has been optimized during the re-optimization phase of the detector and consists now of 4 stations.

The TT station is located downstream of the RICH1 and in front of the Magnet. It fulfills a two-fold purpose. Firstly, it is used in the Level-1 trigger to assign transverse-momentum information to large-impact parameter tracks, using the fringe field of the magnet. Secondly, it is used in the offline analysis to reconstruct the trajectories of long-lived neutral particles (like Kaons) that decay outside of the volume of the Vertex Locator and of low-momentum particles that are bent out of the acceptance of the experiment before reaching tracking stations T1-T3. The TT station is made of four plates of silicon strip detectors. The first and fourth layer have vertical strip while the second and third have a stereo angle of $\pm 5^\circ$. The T1-T3 are split in Inner Tracker and Outer Tracker. In the inner part, where a high track density is expected, silicon strip detectors are used in a similar way as for the TT station. The Outer Tracker detector consists of layers of straw drift-tubes where each drift cell has an inner diameter of 5 mm.

1.2.5 The calorimeters

The calorimeters are used to identify photons, electrons and hadrons and measure their energy [17]. The primary function of the calorimeter is to provide input to the first level of trigger. The system is split in four parts:

SPD Scintillator Pad Detector plane for reduction of the background of the high E_T π^0 tail in the electron trigger.

PS Pre Shower allowing to separate photons and electrons by the topology of the electromagnetic showers subsequently measured in the ECAL.

ECAL Electromagnetic CALorimeter measuring the energy of photons and electrons.

HCAL Hadronic CALorimeter measuring the energy of the hadrons.

The SPD and the PS are put on either sides of a 12 mm thick lead wall and are made of 15 mm thick scintillator pads. The ECAL uses the "shashlik" technology with lead as absorber material. Finally, the HCAL uses 16 mm thick iron and 4 mm thick scintillating tiles.

1.2.6 The muon detector

Muon triggering and offline muon identification are fundamental requirements of the LHCb experiment [18, 19]. Muons are present in the final states of many CP-sensitive B decays. The muons are the only charged particles traversing the calorimeter system (besides neutrinos). The detector consists of four stations M2-M5 embedded in an iron filter and a special station M1 in front of the calorimeter. The first muon station is used to measure the transverse momentum which is used for the L0 trigger. The stations are made of multiple wire proportional chambers.

1.2.7 Read-out, Data Acquisition and Triggering at LHCb

The signals from the sensors are treated in a different manner for different sub-detectors. In general the most complex part of the signal processing is done in a radiation-safe area.

The detector signals are preamplified close to the detector and can undergo different kinds of preprocessing. The data are transported along 50 to 100 m long transmission lines to the counting room by copper or optical fibers. At this point the “Level 1” electronics (see next section) is in charge to perform a new stage of processing of the signals before to transfer the data to the Level 1 (L1) and High level triggers (HLT).

LHCb features a multi-level trigger system, based on calorimeter and muon systems at the first level (L0), on displaced vertices at the second level (L1) and a complete event reconstruction at the last level (HLT). L1 and HLT are software triggers, based on a 2000 CPUs farm.

The different parts of the experiment need to be synchronized and controlled. Clocks and fast signals use the TTC [20] concept based on optical transmission system. The Experiment Control System [21] (ECS) is a bi-directional interface (Ethernet) used to define the running conditions and communicate the parameters to the different subsystems, like running configurations, pedestal tables, temperature values,... It will also be used to transmit debugging and error signals, statistics, etc..

1.2.8 The Level 1 electronics and the TELL1 board

The signals from the front-end chips, after the L0-trigger decision are transferred to the “Level-1 electronics” (or Off Detector Electronics) for further processing. The Level-1 readout board which was developed for the VeLo is now adopted by many other LHCb sub-detectors: VeLo, IT, TT, OT, Muon and Calorimeters. The name of this central element of LHCb DAQ now “Trigger ELelectronics and Level-1 board” or TELL1. Other special systems which will use the TELL1 board in a special configuration are the Muon trigger, the Calorimeter trigger and the L0-Decision-Unit.

Sub-system	N boards	Input-interface	L1 trigger
VELO	88	Analogue	Yes
IT+TT	89	Optical	Yes
OT	24	Optical	Possibly
Calorimeters	22	Optical	No
Muons	10	Optical	No
Cal. Trigger	1	Special	No
Muon Trigger	8	Special	No
L0DU	1	Special	No

Table 1.1: *TELL1 boards for the LHCb sub-systems, with an indication of the number of boards needed and the kind of interface to the front-end. The last column indicates if the information will be sent to the L1 trigger processor.*

The R&D for this device started in 1998 [22]. The first prototype of L1 electronics (RB1), a 10 MHz board built in 1999 was followed by two 40 MHz boards of increased complexity. RB2 [23] is still used in test-beams and RB3 [24] is used in lab tests of readout chips, etc..

The decision to adopt of TELL1 by other subsystems in 2003 caused a considerable delay in the project due to the formulation of a precise requirement list for all users and

the need to setup a coordination strategy. The work on the final schematics of the board started in March 2003 and the prototype boards constructed at the end of 2003.

We foresee the construction of about 300 boards in total, as shown in table 1.1.

We summarize hereafter the general features of TELL1 which will be described in detail in the rest of this document. The TELL1 is implemented on a 12 layers PCB board of the 9U format. In order to get a maximal flexibility, TELL1 has been designed to accept input “mezzanine” cards for all the possible configurations requested by each sub-systems. The TELL1 board makes extensive use of large FPGAs, allowing to implement a variety of algorithms. For all sub-systems, TELL1 will perform the L1 buffering during the L1 trigger decision latency, the interfacing to the HLT, and also the interface to the L1 topological trigger, when required by the sub-system (see table 1.1). The communication with HLT and L1 trigger machines uses an unidirectional Gigabit ethernet connection. TELL1 will be controlled by ECS via a mezzanine “credit-card PC” running a Linux kernel, connected to the LHCb ethernet LAN. Clock and L0 and L1 trigger signals are received via the TTC system by optical fiber. For all the sub-detectors except the VeLo, the data are digitized close to the detector and transported by optical fiber to the TELL1, equipped by optical receivers. The special subsystems (the Level 0–Decision–Unit LODU and some triggers) use the TELL1 in an “ad hoc” configuration with their own input cards.

Chapter 2

The trigger and the data acquisition systems

This chapter explains in more details the concept and organization of the trigger and data acquisition systems in LHCb.

2.1 Why do we need a trigger system?

At the LHC every 25 ns a proton-proton collisions is generated. Taking into account that the size of one LHCb events corresponds to about 0.1 Mbytes, recording all the LHCb detector information at the LHC rate would require to store 4 Terabyte per second! To reduce this rate, the trigger has the task to select, out of these millions of events, the most interesting 200 per second. Only these events will be stored for further analysis.

The principle of data selection by a trigger system is not only employed in high energy physics but also for example in common electronics laboratory equipment as oscilloscopes. In the analog oscilloscope with a conventional cathode ray tube (CRT), the vertical axis is used to record the voltage which is sensed by a probe, amplified and then applied to the vertical deflection voltage plates. The horizontal deflection is started by the trigger system of the instrument. In the most common case, the trigger starts the time voltage ramp at the instant at which the measured signal passes a certain voltage level. As a result, the signal on the screen appears as a stable trace illustrated in figure 2.1. On the other hand, the trigger decision takes some time (nanoseconds, in the case of an oscilloscope) and the signals to the plates must be delayed accordingly, if one do not want to loose the first part of the waveform. Passing to the more recent digital oscilloscope a revolution in the

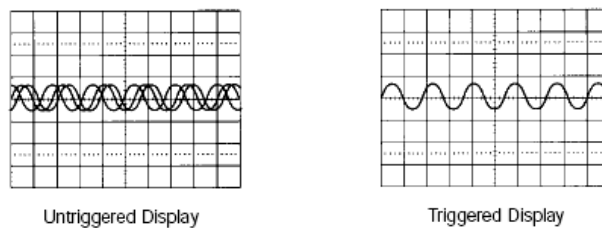


Figure 2.1: *Triggering stabilizes a repeating waveform.*

trigger system can be observed. The voltage sampled with the probe is digitized, stored and made available for the trigger processor. Defined by the instrument settings, the waveform undergoes digital signal processing to enhance for example the display quality, to select the display range in time and amplitude after signal capture, to select special (rare) occurrences as “glitches”¹ or other special events which are identified by some logic conditions calculated on the sampled data. The necessity of an efficient trigger system in such an instrument can be seen considering the sampling rate exceeding 20 G samples per second.

Trigger and data acquisition systems in high energy physics have grown to very powerful processing and high bandwidth data networks. In figure 2.2 an overview of the trends in trigger systems in high energy physics experiments is given. As in the oscilloscope,

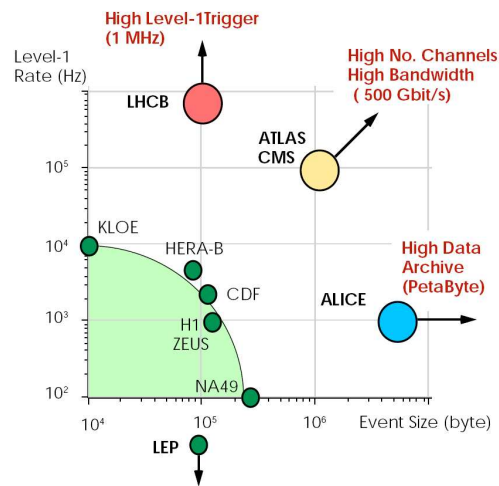


Figure 2.2: Trigger rate of the first level trigger and the average event size are shown on the axis. The multiplication of the two parameters is a measure for the acquisition bandwidth needed in the experiment. Experiments with large event size (like ALICE) must reduce their trigger rate in order to keep the amount of stored data within reasonable limits. Picture taken from [25]

trigger systems for high energy physics detectors select events according to a trigger logic based on physics data. Rapid rejection is essential since the information from all detector channels need to be delayed (buffered) like in the oscilloscope pending the trigger decision. High overall rejection is achieved by progressively reducing the rate after several stage of selection which allows to use more and more processing intensive algorithm (see figure 2.3 as an example). An overview of the first-level trigger systems at LHC is given in [27].

¹A glitch is a fast undesired transition of digital signals within one clock cycle usually caused by capacitive coupling between circuit traces, power supply ripples, high instantaneous current demand by several devices.

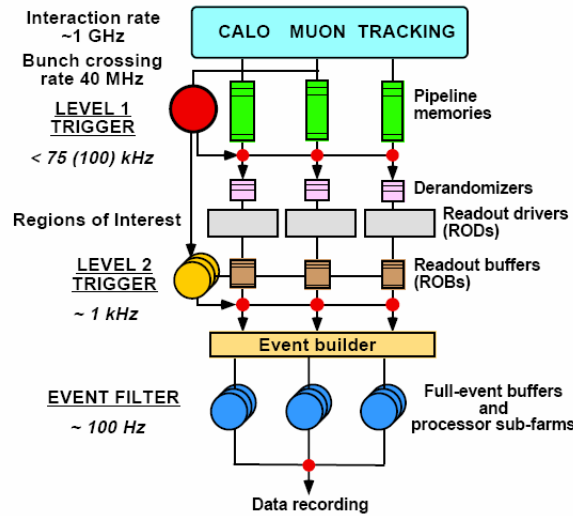


Figure 2.3: *The ATLAS multi-level trigger. Three levels of triggers are present, with pipelines and buffers to hold the data during the trigger decision latency. Picture taken from [26].*

2.1.1 The LHCb trigger from the physics point of view

With the given LHC bunch structure and the reduced luminosity at LHCb the rate of visible pp-interactions² is expected to be 10 MHz. This rate has to be reduced to 200 Hz, at which rate the events are written to storage for further offline analysis. From the 10 MHz visible pp-interactions a rate of $b\bar{b}$ -pairs of 100 kHz is expected. Furthermore, only 15% of the events will have one B-meson with its decay products contained in the acceptance of the detector. From the remaining events, the branching ratio for B-mesons interesting for CP violation studies is typically 10^{-3} leading to usable event rates of the order of “a few per second”. In LHCb the trigger selection is achieved by a three level trigger scheme: Level-0 (L0), Level-1 (L1) and High Level Trigger (HLT). We give now a short description of the selection criteria used at each level.

L0 trigger

The first level trigger reduces the event rate from the 40 MHz of LHC bunch crossing down to 1 MHz. This reduction has been chosen such that in principle all sub-detectors can be read out and therefore can contribute to the next trigger level (except for RICH). The L0 trigger is based on the fact that b-hadron decays result in leptons, hadron or photon with large transverse energy E_T or momentum p_T . Therefore the trigger reconstructs:

- the highest E_T hadron, electron and photon clusters in the Calorimeters,
- the two highest p_T muons in the Muon Chambers.

²An interaction is defined to be visible if it produces at least two charged particles with sufficient hits in the VeLo and T1-T3 to allow them to be reconstructible.

This information is collected by the Level-0 Decision Unit to select events. In addition global event variables such as charged track multiplicities and the number of primary interactions (as reconstructed by the Pile-Up system, an upstream portion of the VeLo detector) are used to reject events.

The L0 trigger system is a fully custom implementation and is split up in a data extraction part located in the radiation area and the actual trigger processor located in the counting house.

L1 trigger

The Level-1 algorithm uses the information from Level-0, the VeLo and the TT. The algorithm reconstructs tracks in the VeLo and matches these to Level-0 muons or Calorimeter clusters to identify them and measure their momenta. The fringe field of the magnet between the VeLo and TT is used to determine the momenta of particles. Events are selected based on tracks with a large p_T and significant impact parameter with respect to the primary vertex position, which has to be calculated independently. The event rate is reduced from 1 MHz by a factor 25 to a maximum of 40 kHz. The L1 trigger algorithm is implemented in a cluster of general purpose CPUs.

HLT trigger

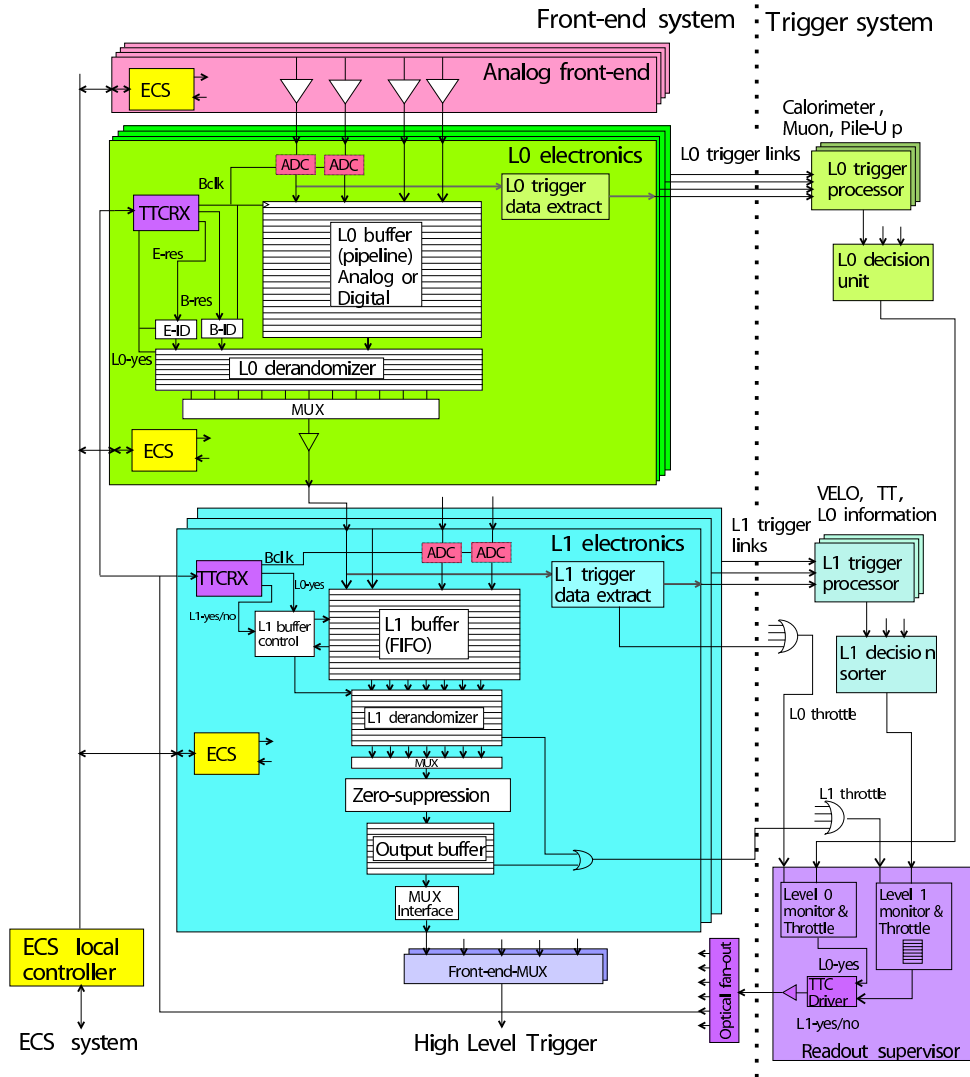
The HLT will have access to all the sub-detectors data. It recomputes the VeLo tracks with better precision and the primary vertex. A fast pattern recognition links the VeLo tracks to the tracking stations T1-T3. The final selection of interesting events is a combination of confirming the Level-1 decision with better resolution and selection cuts dedicated to specific final states. The processing is implemented using a part of the combined L1T and HLT CPU farm. The HLT selected events are stored for the off-line analysis.

2.2 Overview of the data acquisition

The experiments at LHC require an acquisition system very demanding in terms of resistance to radiation, event rate and data bandwidth. The high radiation level in the proximity of the detector requires full custom electronics. Therefore all the analog front-end circuits are implemented as ASICs (Application Specific Integrated Circuit) using special design rules and silicon processes developed for the radiation environment.

To minimize the amount of electronics prone to the radiation effects, the data is sent over long cables to the counting house where further processing take place. This complicate the process of synchronization. For this purpose a global clock and fast signalization system is used. Moreover a general experiment control system is needed to start and stop the acquisition, download parameters, monitor the vital parameters of the apparatus (temperature, gas pressures, power supply currents,...).

LHCb calls Timing and Fast Control (TFC) [28] the system used to communicate the “pace” to the whole experiment. The need for an appropriate timing and trigger control (TTC) [20] system for LHC has driven the development for a radiation hard receiver chip called TTCrx. It is used in all front end boards to receive the system clock, the trigger

Figure 2.4: *The FE architecture of LHCb.*

decisions, resets and special commands transmitted via dedicated TFC network driven by the Readout Supervisor (RS) [29]. The RS is responsible for scheduling the trigger decision distribution in order to avoid any buffer overflows. A detailed specification of the front end requirements concerning trigger delays, rates and sequence has been developed.

Before the decision of a trigger level, the data must be stored in a buffer for the whole trigger latency. We call “L0 electronics” the part of the DAQ which comes before the L0 decision. L1 electronics is where the data is kept after L0 but before L1 decision. The general architecture for each trigger level electronics is simple: it consists of a buffer defined by the trigger latency, an interface to receive the trigger decision and an output stage with a buffer to “derandomize” the data transmission to the next trigger level. The derandomization is needed to adjust to the input rate capability of the next level of electronics.

In the Level 0 electronics of LHCb, the data can be extracted and sent to the trigger

processor. The electronic implementation is in general specific to each sub-detector. The detailed concept and parameters are defined in [30]. As mentioned before, several ASICs, the so called “readout” or “front-end” chips, have been developed to cope with the high density of detector channels and the fast event rate. A fixed latency between the pp-interaction and the arrival of the trigger decision of $4 \mu s$ has been defined for the L0 trigger. This latency includes time-of-flight in the detector, transmission delay in the cable and all delays in the front-end, leaving about $2 \mu s$ for the actual trigger logic to derive the decision. The front-end is required to read out events in 900 ns which results in a maximal L0 accept trigger rate of 1.11 MHz. To avoid buffer overflows, the RS emulates the L0 derandomizer and “throttles” the L0 accept rate when the rate values becomes dangerously large.

In the following chapters, the Level 1 electronics will be discussed in detail. The concept and parameters are defined in [31]. A short description is given at this point to complete the overview of the system. The electronics for L1 trigger is entirely located in the counting house and receives its data over long analog or digital links from the L0 electronics located in the LHCb cavern. The L1 trigger is a variable latency trigger. We have foreseen a buffer size sufficient to store 58254 events. With the minimal event spacing of 900 ns this results in a minimal delay of 52.4 ms ³ which is needed to assemble the event fragments on a commercial network and process the L1 trigger algorithm in a cluster of general purpose CPUs. The buffer overflow prevention is implemented with the L0 and L1 throttle network supervised by the RS. In addition to the storage and derandomization of the output data stream, the L1 electronics also performs some data processing. In particular noise filtering and zero suppression.

³The buffer size of $64 \text{ kWords} \times 32$ is assumed. Since each event contains 4 header words plus 32 data words the number of events that can be stored is $64 \text{ kWords} \times 32 / 36 = 58254$. The latency therefore is $58254 \times 900 \text{ ns} = 52.4 \text{ ms}$)

Chapter 3

Data processing for the VeLo: the TELL1

As seen before, the data collected by the sensors is transported from the cavern to the radiation safe counting room.

With the exception of the VeLo, all the sub-systems use optical links for digitized data transmission (“digital-optical” links). In the case of the VeLo it was impossible to place ADCs and optical drivers close to the detector (because of space and high radiation level). The only possibility was to transmit the analogue signal to about 20 m and then to do the conversion to digital-optical. In the end it was decided to avoid this intermediate level and to transmit the analogue data directly to the counting room and digitize the analogue signals inside the TELL1.

This chapter is dedicated to the processing implemented for the VeLo. The requirements for the VeLo are in many aspects the most stringent and it is used as a reference design for all other detectors. The description of the adaptation to the digital-optical version is done in chapter 4. As an attempt to guide the reader through the processing steps, the figure 3.1 is used every time a new processing block is introduced, to show its location. The block diagram in figure 3.2 shows the main interfaces and the partitioning on the TELL1 board. In figure 3.1 the light shaded elements are implemented in the PP-FPGA, the dark shaded in the SyncLink-FPGA.

Most of the designs introduced in this chapter have been implemented in VHDL and have been tested on the prototypes. The complete design files can be found on the CERN EDMS [32] system.

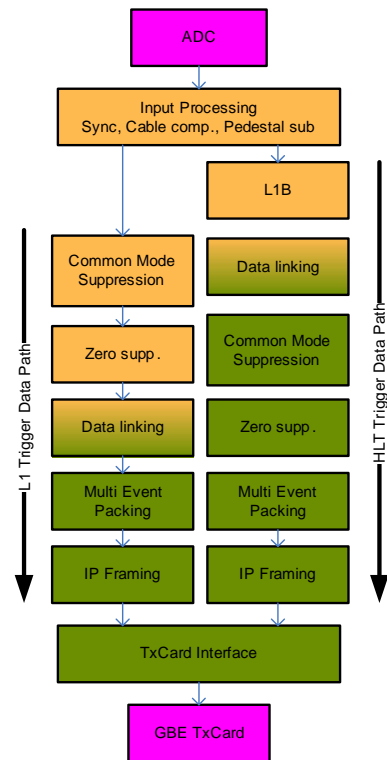


Figure 3.1: *Data processing overview.*

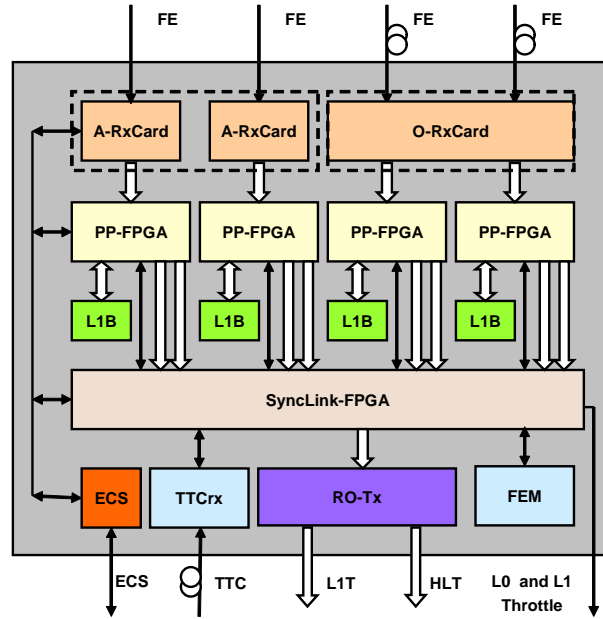


Figure 3.2: Block diagram of TELL1. The two options for optical and analog receiver cards are indicated. There are 5 FPGAs. 4 are named PP-FPGA and one is the SyncLink-FPGA

3.1 The VeLo front end. The Beetles.

The front end chip used in the VeLo (but also in the ST) is called “Beetle” [33]. It amplifies the signals from the silicon sensors, stores them in a pipeline and when a L0 accept signal is issued by the L0 trigger, the samples are derandomized and subsequently transmitted to the output. Each front-end chip can handle 128 detector channels and has 4 analogue outputs: each output multiplexes 32 channels at 40 MHz. A ~ 50 m long copper line will transport the analogue information from the hybrids to the TELL1 boards. In rack mounted close to the detector, radiation tolerant line drivers are used to load the twisted pair lines. We have chosen standard CAT6 cables with individually shielded twisted pairs. A line compensation network is used to correct for the reduced bandwidth of the line (figure 3.3).

The analogue interface on the TELL1 must be capable to digitize at 40 MHz and to do some signal handling. The contribution of the cable to the channel cross-talk can be kept under 5%. Further correction can be done by a digital Finite Impulse Response (FIR) filter in the TELL1. Moreover we have observed a quite large time skew between different lines (of the order of 4 ns for 50 m of CAT6 cable). This cannot be accepted because we would like the ADCs to sample the analogue signal with a 1 ns precision. This problem has been solved by the use of programmable delay for each channel. The FIR and the delay implementation will be described in the following sections.

Before to conclude, we should give some details on the digital aspect of the front end. As we have said, each Beetle processes 128 detector channels. 16 chips are located on one hybrid performing the readout of one sensor of 2048 strips. In the case of the readout scheme chosen by the VeLo, the Beetle transmits the L0 accepted data on the 4

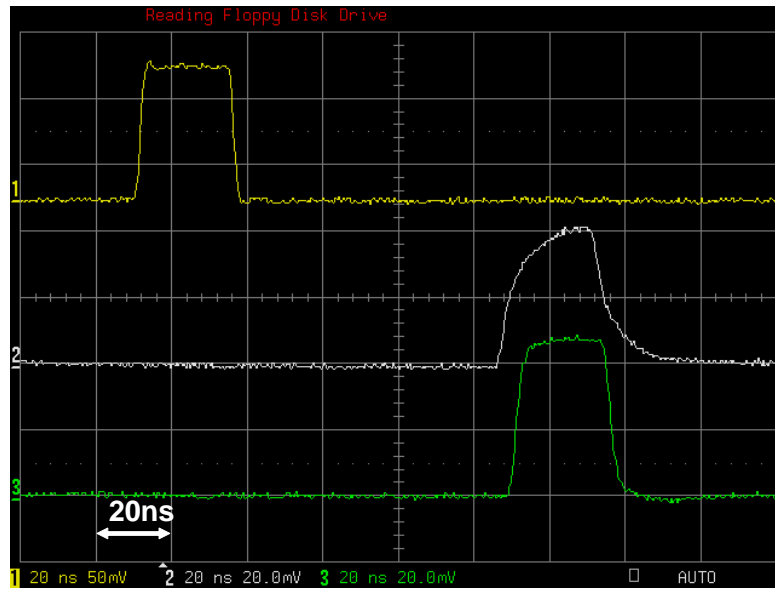
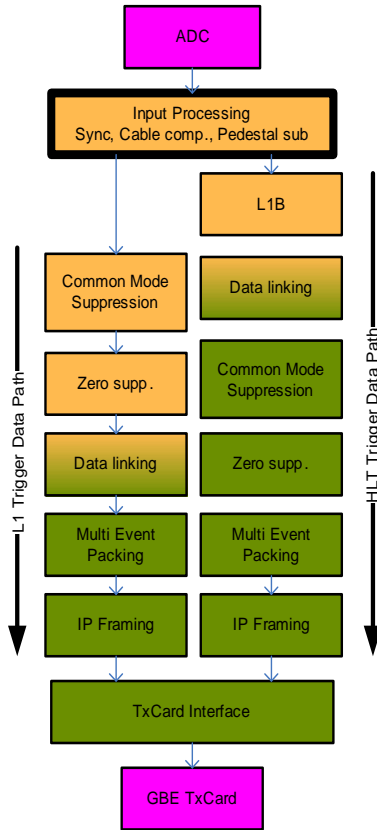


Figure 3.3: *Oscilloscope wave-forms of the injected pulse on the top, of the signal after 60 m of cable in the middle and after cable compensation at the bottom of the figure. LHCb will use a shorter cables about 50 m long. These measurement were done by Raymond Frei.*

analog links at 40 MHz. In addition to the 32 analog signal samples per line, 4 samples of “pseudo-digital” information are added in front of the data representing the “header” (no “trailer” is provided). The header contains the information at which pipeline location “Pipeline Column Number” (PCN) the analog data has been stored during the L0 latency. In addition a few chip specific status and error bits are transmitted. The consequence is that the data transmitted to the TELL1 readout board carry only a sparse event identification, distributed over the 4 links. We should also say that a “data valid” signal is in principle available from the Beetle, but the transmission of this information on an additional link would have increased the link cost by 25%. A different solution has been found as will be explained later.

3.2 Input data processing for the VeLo



An overview of the processing done on the input stage is given in figure 3.4. Each TELL1 board will handle 64 analog channels or 2048 physical channels. On the point of view of implementation, two readout channels are processed time multiplexed in order to reduce the resources needed.

First of all, the digitization process requires a synchronization task. Then, due to the analog data transmission and the specific readout connection scheme for the ϕ -sensors, the first processing stages for VeLo data include a FIR filter to correct for the residual crosstalk (introduced by the finite bandwidth of the long analog transmission cable) and also a sample reordering stage will follow the pedestal subtraction. Other “options” including gain correction and bad channel masking are discussed at the end of this section.

3.2.1 VeLo link synchronization

Three steps of synchronization need to be considered for the VeLo data transmission and will be discussed hereby: at the level of the ADC, at the level of “clock domain”, and at the event identification level.

1) ADC clock synchronization: on the physical layer, the Analog to Digital Converter (ADC) clock has to be synchronized for correct sampling. The typical pulse shape after the transmission is such that the quality of the result depends on an adjustable sampling phase for each channel. The delay skew of the 50 m CAT6 cables is of the order of ns, which must be compensated at the level of the ADC clock phase. In the following we give a description of the technique used on the PP-FPGA for generating these clocks.

Each PP-FPGA provides the sampling clock of 16 ADCs sitting on the analog receiver card. An individual phase adjustable clock is provided to each ADC. To generate delayed clocks by steps of 1-2 ns or so, various possibilities exist:

PLL Modern FPGAs as the Altera Stratix provide on chip PLL circuits where the phase shift can not only be configured at the initialization of the FPGA but can also be updated during operation. This is the perfect clock generator solution as long as the number of clocks to generate does not exceed the available PLLs.

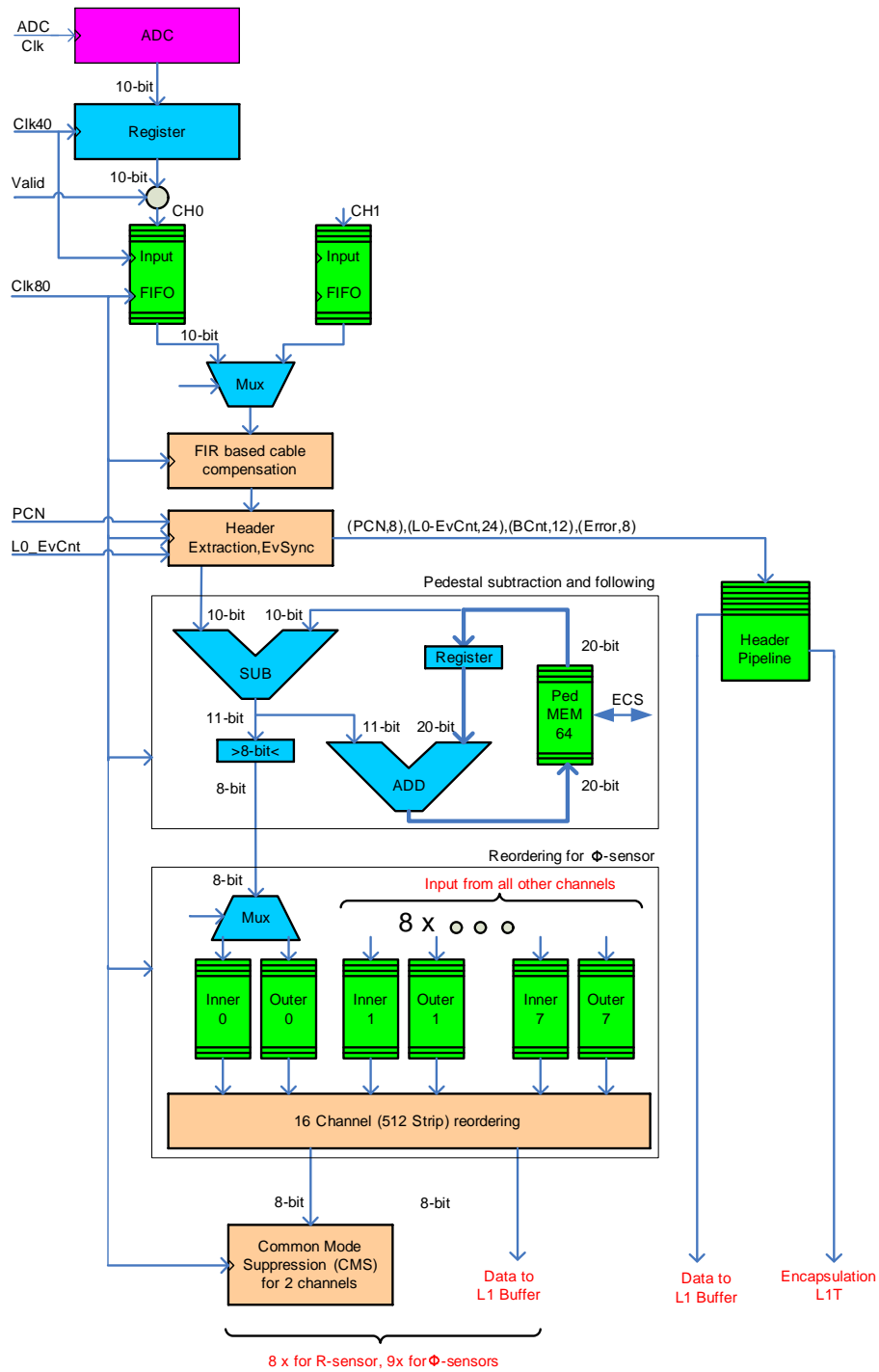


Figure 3.4: Block diagram of the first processing stages on one the PP-FPGA. The data flow from the ADC to the common mode suppression is shown in detail.

Gate delays To delay a signal, the delay accumulated after traversing consecutive gates on the chip can also be used. The drawback on this method is the temperature dependency of the delay.

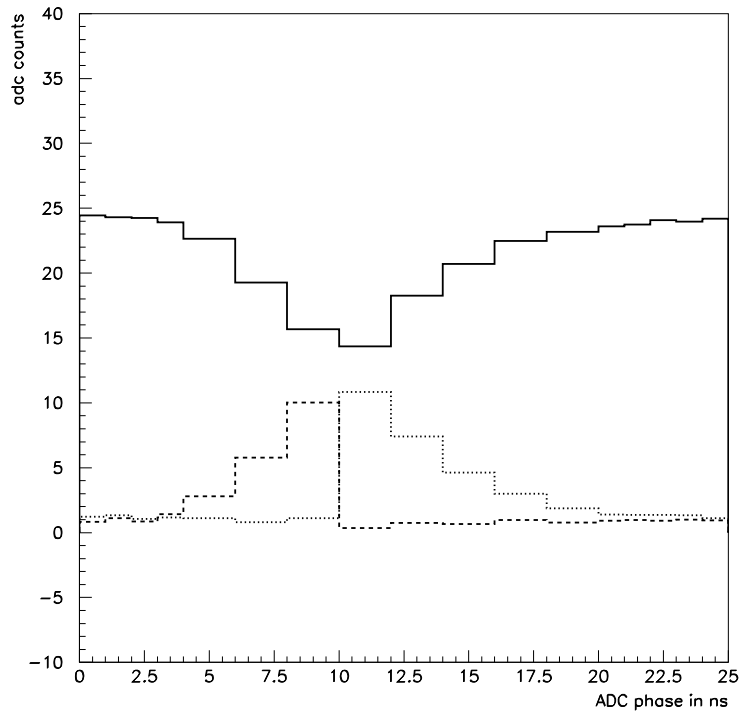


Figure 3.5: *The measurement shows the ADC value of one channel for different sampling phases (0 to 25 ns). In the lower part of the diagram the left and right neighboring channel are plotted. For this measurement we have used 60 m of cable and the values were averaged over 1000 samples. Measurement done by Laurent Locatelli on the RB3 prototype.*

Shift register clocked with a multiple of the base clock This is the method implemented in TELL1. In a first step the PLL circuit can be used to generate 4 clocks at 160 MHz (the base clock frequency multiplied by 4) each phase shifted by 90 degrees (see figure 3.6). Each 160 MHz clock contributes 4 rising edges within one base clock period resulting in a total of 16 equally distributed rising edges for one period of 25 ns. The resulting delay resolution is $25/16 = 1.56$ ns. To extract 40 MHz clocks from these rising edges, the 160 MHz clock is employed to shift the bit sequence "1100" in a looped back shift register. Each bit of the shift register toggles with one quarter of the clock frequency and every bit is one clock cycle (6.25 ns) phase shifted. Finally, each of the 16 generated clocks can be selected by a multiplexer circuit to assign it to the ADC clock pin. To optimize for the resources and the variation due to routing delays, it is advantageous to generate only the first 8 clocks in the way described above. The second 8 clocks, covering the delays 8 to 15, can be generated by inverting the delayed clocks 0 to 7. The most delicate part of this design is the resetting phase. Each shift register needs to have a synchronous reset to ensure its desired initial state.

In figure 3.7 the measurement of one clock output has been recorded in persistence mode where 5 different phase settings were consecutively selected.

2) Clock domain synchronization: once the data has been transmitted to the PP-

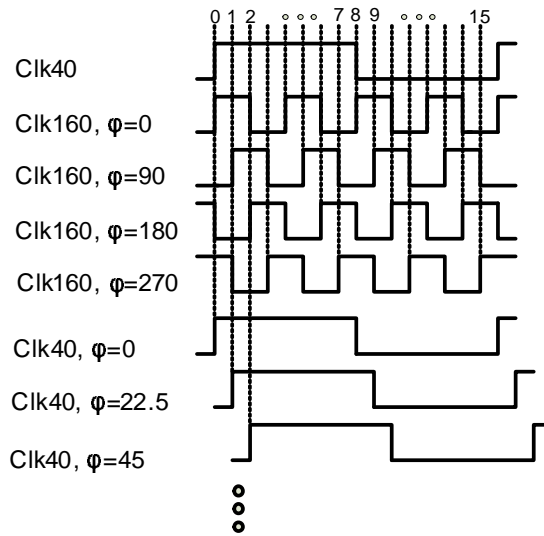


Figure 3.6: Illustration of the clock delay generation to be sent to the ADCs. The delay is based on four 160 MHz fast clocks each 90 degrees phase shifted. Each rising edge of the fast clocks (enumerated 0 to 15) is used to generate one phase shifted 40 MHz clock. The phase shift steps are 1/16 of a period equals 1.56 ns. (seen next figure)

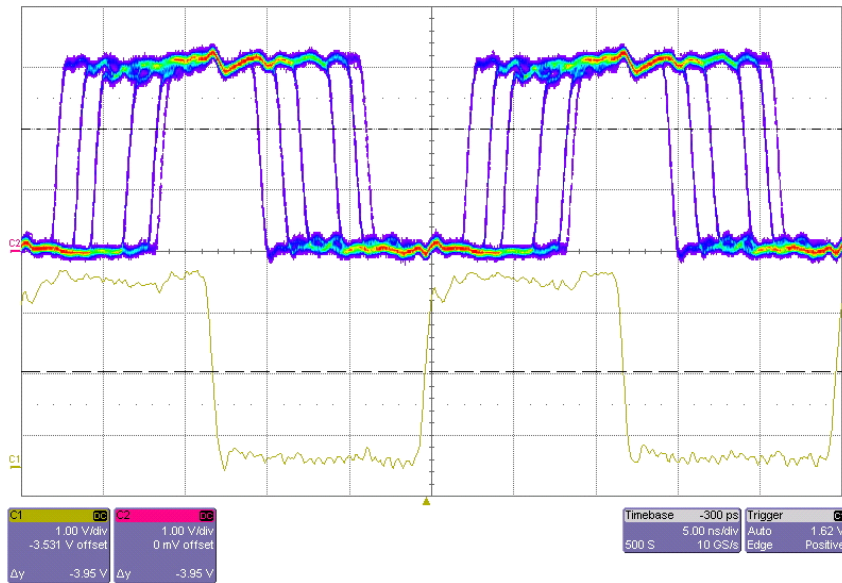


Figure 3.7: The phase shift steps should be 1/16 of a period or 1.56 ns as demonstrated in the previous figure. The difference observed are due to different routing delays on the output multiplexer. The 40 MHz clock on the bottom was used for the trigger where the upper curve is the ADC clock for 6 different settings.

FPGA, the next task is to synchronize all channels to a common 40 MHz clock domain. The synchronization in this case is not done with a FIFO since the phase relationship between the clock domains is known. The correct sampling edge for the input data can

be assigned knowing the clock phase shift at the clock generator circuit also implemented on the PP-FPGA. With two registers in series the data can be synchronized to one single clock domain which is illustrated in figure 3.8.

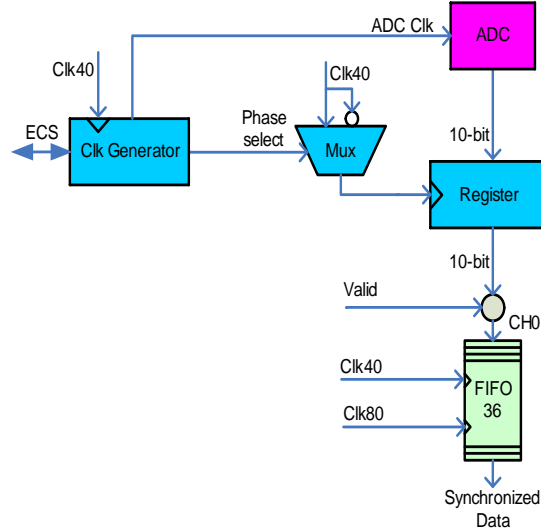


Figure 3.8: The input synchronization uses the clock generator information for selecting the right phase on the input register. Only valid data is written to the subsequent FIFO buffer used to change the clock domain for the following processing stages.

3) Event synchronization: The last step is to identify the event data out of the continuous data stream. A data valid signal is generated with the help of a local running reference front-end chip (the so called “Front end emulator” is a Beetle chip employed to reproduce the same state machines as used in the chips sitting on the detectors). The logic part of the readout chip is more complex then could be assumed looking at the system behavioral view. Because a clock cycle accurate implementation is mandatory the safest option was seen by using the “Beetle” itself. This reference chip provides the “data valid” information. The data valid is then delayed by fixed number of clock cycles to compensate for the data transmission time. Only valid data is written to the input data FIFO where the clock domain is changed before the subsequent time multiplexed processing. To ensure correct event synchronization, the PCN in the event data and the one of the reference Beetle are compared. In case of error, an “exception status flag” is raised and written into the data header. The data processing is continued to ensure that at least event headers can be assembled completely (this is important for the consequent event and multi-event linking stages). For monitoring, error counters can be implemented and made accessible to ECS.

3.2.2 The FIR filter

The frequency compensation on the analog circuit cannot exactly correct for the long cable effects because of component tolerances and the chosen simplified compensation scheme.

To fine tune the compensation a low order Finite Impulse Response (FIR) filter will be employed after digitization. The mathematical definition of the filter is very simple:

$$Q_n = S_n * C_n + S_{n-1} * C_{n-1} + \dots + S_{n-k+1} * C_{n-k+1} \quad (3.1)$$

Where S are the samples before and Q after the filter. The equation describes a filter of order k using a set of k constant coefficients C which parameterize the behavior of the filter. To obtain a filter with DC gain 1, the sum of the constants is required to be equal 1. To implement this FIR filter, k multiplications and k-1 additions per sample are required. With k=3, this sums up to 512×3 Multiplications-ACcumulation (MAC) operations per 900 ns equivalent to 1.7 GMAC/s which is the performance of several high end Digital Signal Processors (DSPs). The implementation of the multipliers in the FPGA needs to be carefully considered to minimize the amount of resources used. To choose the most suitable implementation some simplification to the general approach can be made as described hereafter.

In our case, only fixed coefficient multiplications are required to implement the filter, if the precision of the calculations can be kept low. This opens the possibility to use look-up tables instead of multipliers. Using for example a 4 kbit memory block allows to perform an 8-bit × 8-bit fixed coefficient multiplication.

It must be observed that the corrections made by the filter are typically small. Let's take an example with the following coefficients, in which the first term reflects the fact that 95% of the information is already present in the corresponding signal sample at the input:

$$C_n = +0.95, C_{n-1} = +0.1, C_{n-2} = -0.045, C_{n-3} = -0.005, \dots$$

The simplification that can be done is to scale the output data by the inverse of the first filter coefficient. The resulting filter equation reads then:

$$Q_n = S_n + S_{n-1} * K_{n-1} + \dots + S_{n-k+1} * K_{n-k+1} \quad (3.2)$$

The new coefficients K are essentially the same as before:

$$K_{n-1} = +0.105 \sim 0.1, K_{n-2} = -0.047 \sim -0.055, K_{n-3} = -0.0053 \sim -0.005, \dots$$

Using this method the first term of the filter is kept constant. In the case of an order 3 FIR, with the coefficients of the example we see that the following two multiplications covers a range of one order of magnitude, instead of 2 orders when the first term is included. In binary language, the multiplication of the 8 most significant bits (MSBs) of the samples times the 8-bit value of the coefficient is sufficient to calculate the correction to the n^{th} term which is a 10 bits number.

More quantitatively, let's consider our order 3 FIR example. Using only the 8 MSB of the 10-bit raw data leads to a maximal error (2 omitted bits) of 3 ADC counts on $(n-1)^{th}$ and $(n-2)^{th}$ terms. To calculate the error at the final correction only the difference due to this error is considered.

$$\Delta Q_n = \Delta S_{n-1} * K_{n-1} + \Delta S_{n-2} * K_{n-2} \quad (3.3)$$

For the calculation, the coefficients K are scaled and encoded in a 8-bit binary value. With $K_{n-1} = O(0.1)$ and $K_{n-2} = O(0.01)$ the scaling factor can be chosen for example as

$SH = 2^{10}$ (1-bit is reserved for the sign). The calculation of the error due to the omitted bits for the example is:

$$\Delta Q_n = (3 * 0.105 * SH + 3 * 0.047 * SH) / SH = 0.46 \quad (3.4)$$

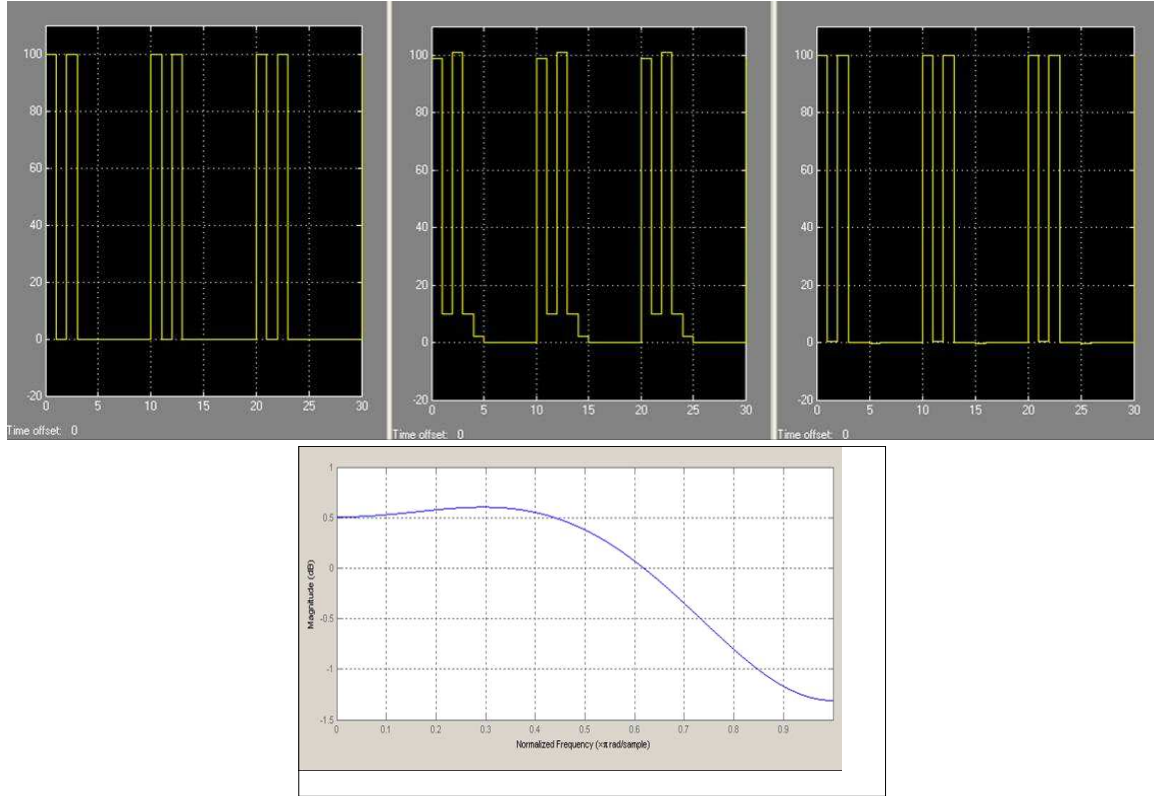


Figure 3.9: *The signal of an ideal driver drives two pulses (left), after passing a cable with low pass characteristic (middle) and the output of the FIR filter (left) are shown in the upper figure. The signal is well recovered. The bandwidth characteristic of the FIR filter is shown in the lower drawing.*

The calculation shows that even with quite large coefficients the resulting error is below 1 ADC count. On the other hand, this method leaves the final amplitude values slightly un-normalized (by 5%, in the example). This should not be a problem because calibration factors have to be introduced in any case in the off-line analysis. In figure 3.9 the filter response of the example is illustrated by a simulation.

A final remark is about the header words in front of the analog samples transmitted by the Beetles. They are encoded in pseudo-digital form with an amplitude which correspond to about twice the signal left by a particle. One ADC sample of the header represents one digital bit. The precise ADC value itself is not required and therefore does not need any filtering. Nevertheless the header affects the first two analog data samples in a considerable way and must be in any case included in the FIR correction of the first samples of the stream.

3.2.3 Pedestal calculation and subtraction

After filtering and before any other operation, in order to make the data set as uniform as possible, individual “pedestals” must be subtracted to each VeLo channel. The pedestals must be calculated periodically because they can change with time (due to temperature changes, for instance). We have identified two different approaches: calculation in the HLT and calculation “in loco”. In both cases the pedestal values must be written in some data-base and also they must be read and write accessible via ECS.

Pedestals calculated in the HLT

The pedestal values can be calculated in the HLT CPU cluster using raw data events. This supposes that periodically raw data are transmitted to the HLT, at the beginning of a “Run”, for instance. This ensures that the pedestal values are well known at any time of data taking and does not require any calculations to be done on the TELL1.

Pedestals are calculated locally

Assuming the pedestal values change too often during data taking, a local and continuous update can eliminate the need of frequent external recalculation and updating. The drawback of this method is that the pedestal values applied at every moment are not easily available for the offline analysis. To cope with this problem, one pedestal value per 64 strips can be added to the data transmitted to the HLT using a free space in the data header.

Several methods can be used for a continuous correction. We can consider the very simple possibility to just average a set of N (of the order of 1000) events and use the results from this set as pedestals for the the next, and so on.

Another possibility is to use a “running pedestal sum”, over a set of N events. The pedestal for a given channel at “time” n , $P(n)$, is then obtained by calculating the ratio

$$P(n) = \frac{P_{sum}(n)}{N} \quad (3.5)$$

The the running pedestal sum, $P_{sum}(n)$, can be defined by recursion: $P_{sum}(0) = 0$ or to any known value (settable via ECS) and

$$P_{sum}(n + 1) = P_{sum}(n) + S_{n+1} - P(n) \quad (3.6)$$

where S_{n+1} is the value of the data sample at $n+1$. We must notice that the pedestal corrected value $\overline{S_{n+1}}$ equal to the second minus third term in equation 3.6 has to be calculated in any case for subsequent use. This leaves only one extra addition to be performed for the sum calculation:

$$P_{sum}(n + 1) = P_{sum}(n) + \overline{S_{n+1}} \quad (3.7)$$

This procedure can be implemented using the scheme shown in the block diagram figure 3.10. To store the sum 10-bit pedestal values over 2^{10} samples (events), a total width of 20-bit is required. These 20-bit values, one per strip, need to be stored in

the pedestal RAM. One 4 kbit RAM block configured as 64×32 -bit (2 kbit) is therefore sufficient for the two multiplexed 32 strip packets.

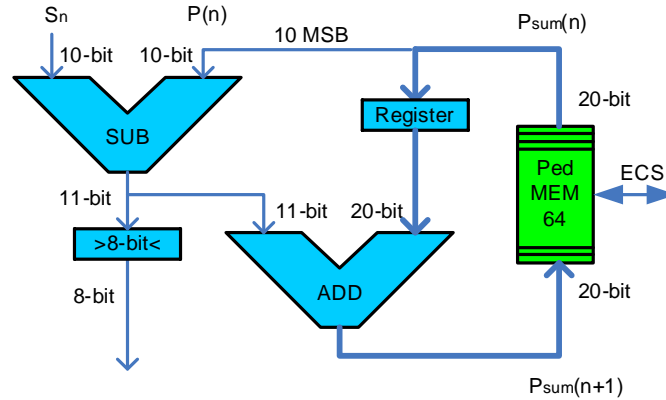


Figure 3.10: For one processing channel, one subtraction of 10-bit and one accumulator of 20-bit together with a 4 kbit memory block are used for the pedestal subtraction, following and storage.

The last step at the pedestal subtraction is to limit the data samples to the 8-bit signed value. The signal range is therefore limited to $+127 \dots -128$.

3.2.4 Channel reordering

This operation is needed to correct for the particular connection topology in the VeLo sensors. For the R-sensors (see figure 3.11) a simple connection scheme has been adopted. Each Beetle readout channel carries the information of a block of 32 detector strips.

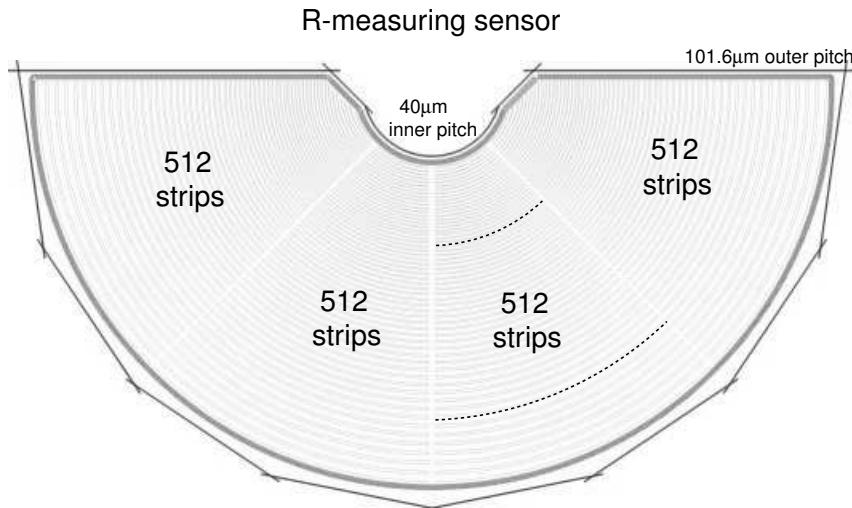


Figure 3.11: *VeLo R-sensor from July 2003.*

The connection of the 16 Beetle chips to the different regions of the detector is illustrated in figure 3.12. Remark that the inner region of the detector has a pitch of $40 \mu\text{m}$

and the outer region $101.6 \mu\text{m}$ and therefore the region that can be readout by one Beetle is much smaller.

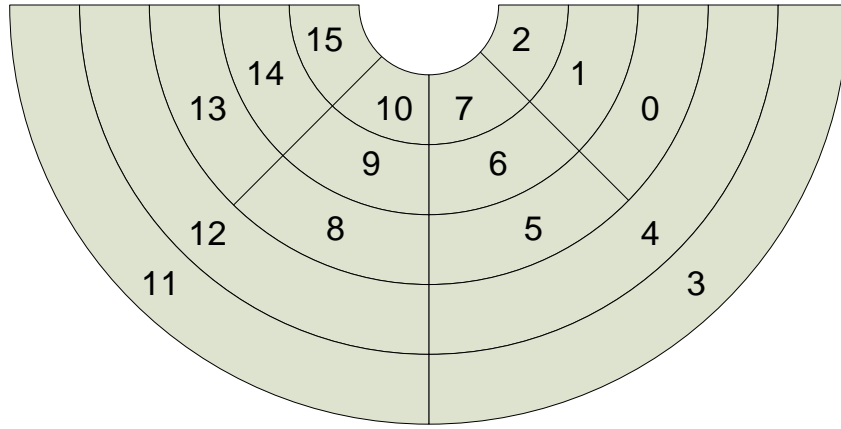


Figure 3.12: *VeLo R-sensor distribution of the readout region per Beetle chip.*

The ϕ -sensor is divided in 683 inner (one third) and 1365 outer strips (two thirds) (3.13). The connection of the strips to the bonding pads is illustrated in figure 3.14. To

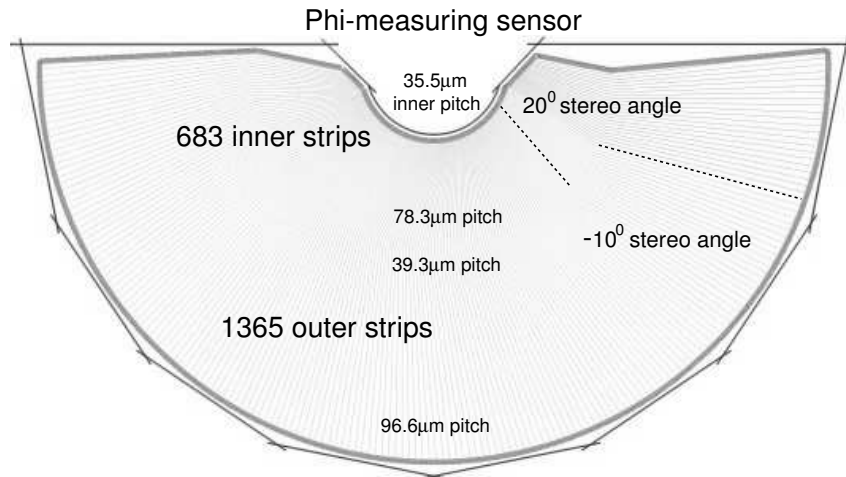
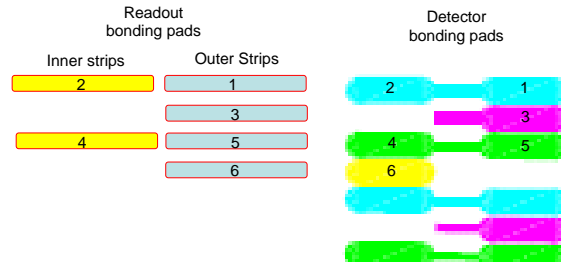


Figure 3.13: *VeLo ϕ -sensor from July 2003.*

simplify the detector layout, the second and third strip within a group of four are always read out in inverse order. To rearrange the strips to the physical layout on the detector, the sequence of strips "12345678" must be transformed into "13245768". The result of this bonding scheme is that inner and outer strips are mixed in the readout chain. The reordering of the channels read out by the Beetle to the spatial location on the detector is limited among 512 strips processed by one PP-FPGA. No interconnection between PP-FPGAs is provided. With the assumption that the Common Mode Suppression (CMS, see next section) is done among 32 detector channels and the fraction of inner and outer strips is one third to two thirds, the processing groups can not be perfect groups of 32 adjacent

Figure 3.14: *VeLo ϕ -sensor bonding order.*

strips. A proposal for the reordering is given in figure 3.15. The further processing for the CMS is done in time multiplexed mode. Two sets of 32 strips are processed in sequence in a pipeline with doubled speed. At the reordering this has to be taken into account. The processing channels are indicated in the last column in the table shown in figure 3.15. Due to the fact that the number of inner and outer strips can not all be merged into packets of 32 strips, an additional processing channel (9 in total, instead of 8) is needed. The dummy strips inserted in the last two packets of 32 are set to 0 which is the expected average value of a strip after pedestal subtraction. This ensures that a minimal effect to the CMS is obtained. Reordering therefore can not be performed previous to pedestal subtraction.

Analog Channel Beetle	Mux Pedestal	Inner Strips	Outer Strips	Reorder	32 Packet	Inner Strips	Outer Strips	Mux Channel
0A	0	10	22	X	0	32	0	0
0B		11	21		1	0	32	
0C	1	11	21	X	2	0	32	1
0D		10	22		3	32	0	
1A	2	11	21	X	4	0	32	2
1B		11	21		5	0	32	
1C	3	10	22	X	6	32	0	3
1D		11	21		7	0	32	
2A	4	11	21	X	8	0	32	4
2B		10	22		9	32	0	
2C	5	11	21	X	10	0	32	5
2D		11	21		11	0	32	
3A	6	10	22	X	12	32	0	6
3B		11	21		13	0	32	
3C	7	11	21	X	14	0	32	7
3D		10	22		15	10+Di	0	
				X	16	0	22+Do	8
					17	0	0	

Di: Dummy channels to insert for inner strips (22)
Do: Dummy channels to insert for outer strips (10)

Figure 3.15: *VeLo ϕ -sensor reordering.* In the first column the analog channels are listed (Beetle 0...3), two of them are processed time multiplexed for the cable compensation and pedestal subtraction.

This procedure needs some more detailed considerations. Starting from the output data stream of the pedestal subtraction the multiplexed data stream delivers data in

sequential order, first from the 0A, 0C analog Beetle readout channel followed by the 0B and 0D (see first column in figure 3.15). To understand the design problem, we look at the collection of the inner strip data. Looking at the processing channels (0A,0B) and (0C,0D) we can see that both provide inner strip data at the same time making a write operation to one memory impossible unless we enlarge the data bus width or increase the working frequency. This problem does not result from the fact that the processing in the previous steps has been time multiplexed. It just comes from the fact that 3 simultaneous data streams need to be merged into one. The same problem occurs for the outer strip collection with 2 data sources. One solution proposed to solve this problem is shown in the data processing diagram in figure 3.16.

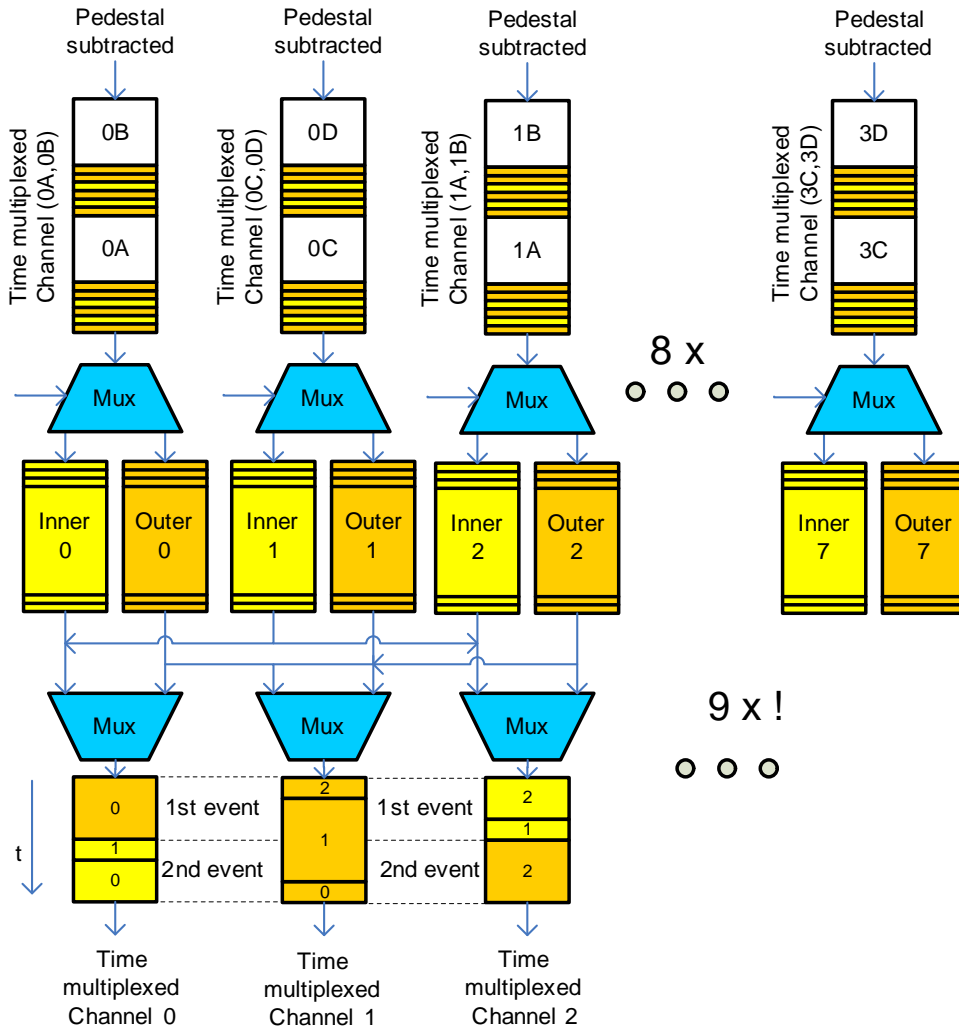


Figure 3.16: Procedure to reorder the ϕ -sensor data.

The reordering is executed in two steps, taking advantage of the possibility to rearrange the data at the write and the read process. At the write, the inner and outer strips per subsequent readout channel are separated and written into individual RAM blocks. This separation requires 8 memory blocks for inner and 8 for outer strips. The subsequent readout of these memories requires an additional multiplexing stage where the data

fragmented in different RAM blocks is assembled. The requirement that only one access to a memory at the time is needed is fulfilled. The access time to each memory can be seen from the block diagram where the length of access is indicated in the following data processing blocks (figure 3.16 last blocks.). The horizontal dotted lines in the output data stream indicate the first and the second 32 strip packets.

The reordering procedure proposed here takes advantage of the small memory blocks available on the FPGA. The reordering is “hard coded” in the FPGA and can not be changed by register or RAM configuration. The problem of the simultaneous arriving of data that needs to be processed or written to a memory is an every returning conceptual problem. It can be solved by using a multiplexer running at an increased frequency (for 3 input data sources the frequency of this multiplexer has to be tripled) or the data path is extended to a larger size. Both solutions result in extending the data bandwidth.

3.2.5 Other processing options

PCN inhomogeneity correction

The inhomogeneity of the pipeline of the Beetle front-end chip may require to use a “second order” pedestal correction taking the pipeline location inhomogeneity into account. With 187 pipeline locations the pipeline correction RAM requires $256(187) \times 512 = 128$ kbit values per PP-FPGA. This table can be accommodated by one out of two M-Memory block (512 kbit) available on the chip. The memory can be configured as 128-bit wide memory providing a 4-bit offset value leaving a correction range from -8 to +7 ADC counts. The calculation of the correction table cannot be done on the PP-FPGA due to the lack of memory.

Gain Correction

To compensate for gain inhomogeneities among detector channels a gain correction processing can be inserted. The operation required is one multiplication. The fixed coefficient 8-bit \times 8-bit multiplication can be done on a single 4 kbit memory block. The “Gain Correction Table” has to be downloaded via ECS and requires one 8-bit value per detector strip. The total amount of memory required by the gain correction on one PP-FPGA is 16 times 4 kbit, a considerable amount of memory (20% of the 4 kbit memory blocks on the chip!).

Channel Masking

Masking particularly noisy or dead detector channels can be applied in combination with another processing stage. If for example a gain correction is required, setting the gain to 0 masks the particular channel.

3.3 Data processing for the L1 trigger

3.3.1 L1 Trigger Common Mode Suppression (L1T-CMS)

After the data synchronization, cable compensation and pedestal subtraction the data is processed in two data streams. For the HLT the the next stage is to store this data in the L1 buffer. The 10-bit data sampled by the ADC is reduced to 8-bit width. This allows to avoid saturation of the signal keeping a good resolution even in the presence of a large Common Mode (CM) noise.

The source of the CM noise is for example RF pick-up from the sensor (for instance from the beam itself which is very close to the silicon sensors in case of the VeLo). Other sources can be power supply variation in the front-end chip due to high occupancy or similar effects on the analog readout chain. “Ground loops” can induce such effect over the transmission lines. The CMS goal is to filter CM noise that is variable from event to event.

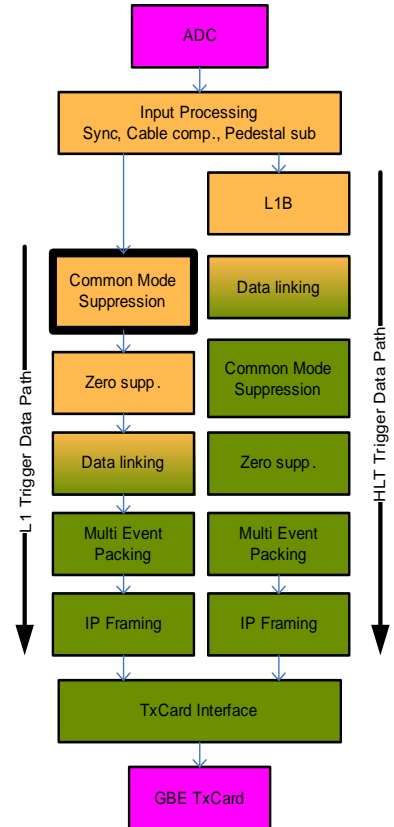
In the following, it is important to distinguish between CM induced at the sensor level and therefore dependent on the spatial location of the silicon strip and the CM that comes from the readout chain. In the second case, the CM suppression should be calculated among the strips that are readout together. In the past, the largest portion of CM was seen to come from the sensors. This has lead to the assumption that it is appropriate to reorder the strips in spatial order previous to the CMS. For the VeLo R-sensors, the complete readout is made in spatial order which allows to correct for the two sources of CM.

Several algorithms for the CMS have been tested with Monte Carlo and test beam data. The implementation of the Linear CMS (LCMS) algorithm was prototyped on a FPGA to estimate the required resources [34]. This implementation has lead to the choice of a specific kind of FPGA with enough resources. In the following the LCMS algorithm is discussed in detail.

Linear Common Mode Suppression (LCMS) algorithm

The CM correction procedure presented here assumes a noise varying in a time domain much larger than the 25 ns width of a VeLo sample. In this case a linear approximation can be done to correct for CM: the Linear Common Mode Subtraction method (LCMS). The LCMS is applied on a set of 32 detector channels ¹. It consists of two identical

¹The choice to group 32 detector channels for the LCMS is an adaption to readout with the Beetle. The algorithm described can be adapted to other (power of 2) sample counts.



iterations, in which a linear approximation of the CM present in one event is calculated. The algorithm subsequently identifies the hits and passes the information to the cluster encoding stage. The different steps of the algorithm are illustrated in figure 3.17.

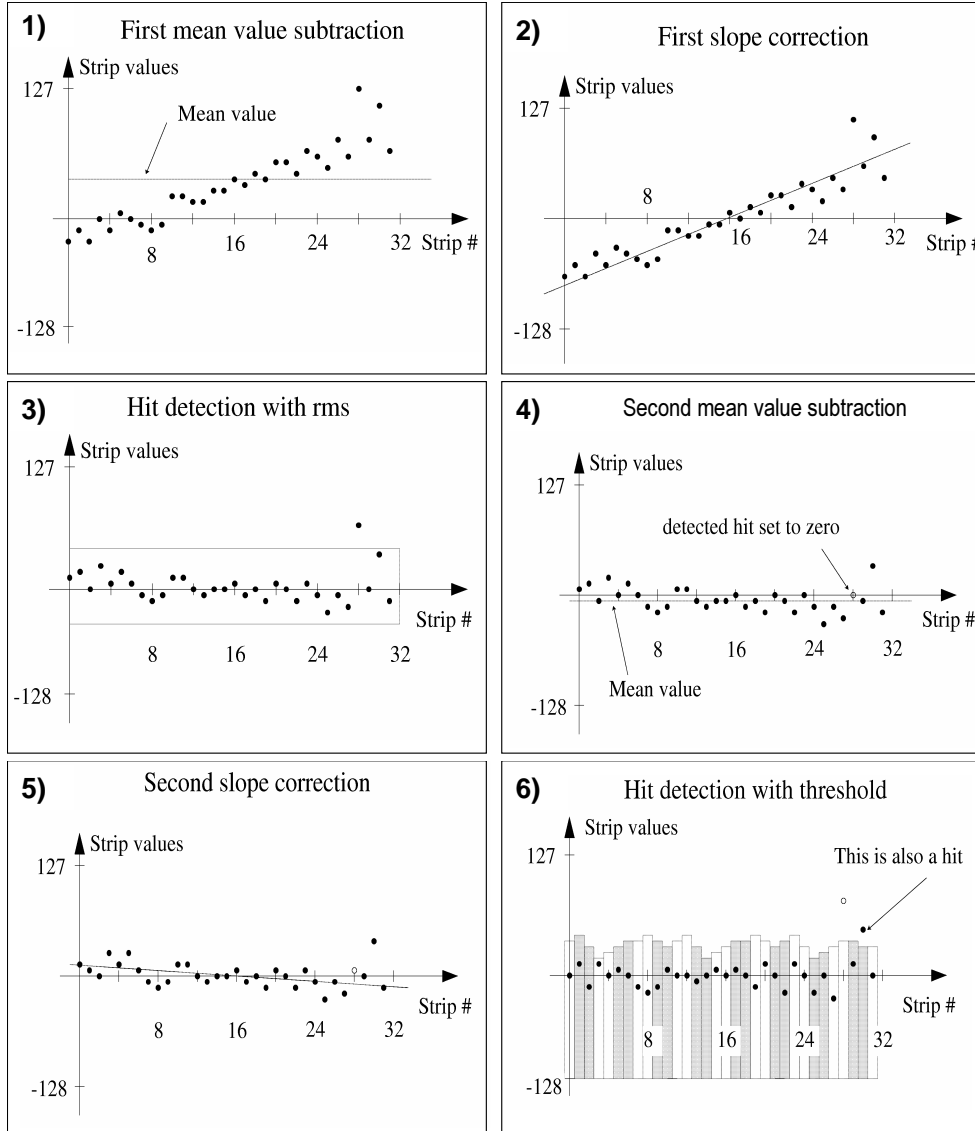


Figure 3.17: *The LCMS algorithm consists of two iterations. In a first iteration, the signal above a threshold is masked to allow for a refined second iteration.*

We give now a **formal description of the LCMS algorithm**. The linear approximation over a set of 32 samples is done by linear regression. In order to obtain an implementation with a reasonable cost in terms of resources, some approximations are

introduced. Starting from the function for a linear regression,

$$\tilde{y}(x) = \bar{y} + \frac{\sum_{i=0}^{N-1} x_i y_i - \bar{y} \sum_{i=0}^{N-1} x_i}{\sum_{i=0}^{N-1} x_i^2 - \bar{x} \sum_{i=0}^{N-1} x_i} (x - \bar{x}), \quad (3.8)$$

where N is the number of samples, x_i and y_i are the Cartesian coordinates of the sample points and \bar{x} and \bar{y} are the mean values over all samples in x and y direction. The formula can be simplified by re-centering both x and y values in such a way that the average values \bar{x} and \bar{y} are close to 0.

The samples are about equally spaced in x and by taking $x_i = i - 16$, we can approximate:

$$\bar{x} = \frac{1}{32} \sum_{i=0}^{31} x_i = -\frac{1}{2} \approx 0. \quad (3.9)$$

In order to center the ordinate at 0, we shift the y values by subtracting the average

$$a_i = y_i - \bar{y}_i. \quad (3.10)$$

This leads to the following approximation for equation (3.8):

$$\tilde{a}(x) = \frac{\sum_{i=0}^{31} (i - 16) a_i}{\sum_{i=0}^{31} x_i^2} x \quad (3.11)$$

with

$$\sum_{i=0}^{31} x_i^2 = 2736. \quad (3.12)$$

The division by the number 2736 is a difficult task to perform in an FPGA and we rely on the approximation

$$2736 \approx \frac{2^{13}}{3} = 2730.\bar{6} \quad (3.13)$$

The division can be replaced by a multiplication by 3 and a shift operation. The error introduced by this approximation is negligible for 8 bit precision.

The **first iteration** can be split up into the following steps, starting with the original pedestal corrected data y_i :

- Mean value calculation

$$\bar{y}_1 = \frac{1}{32} \sum_{i=0}^{31} y_i, \quad (3.14)$$

- Mean value subtraction

$$a_i = y_i - \bar{y}_1. \quad (3.15)$$

- Slope calculation

$$s_1 = \frac{3}{2^{13}} \sum_{i=0}^{31} (i - 16)a_i \quad (3.16)$$

- Linear CM subtraction

$$b_i = a_i - s_1(i - 16) \quad (3.17)$$

First hit detection After the linear CM subtraction the hits are identified by comparing the absolute value of the sample amplitude to an adaptive limit built from the rms of the sample distribution. The two steps used to identify are the following:

- Calculation of the variance. Since the mean value is now 0, the variance is simply

$$V = \frac{1}{31} \sum_{i=0}^{31} b_i^2 \approx \frac{1}{32} \sum_{i=0}^{31} b_i^2. \quad (3.18)$$

- The signal of each channel i is compared to the variance V :

$$b_i^2 F_1 > V. \quad (3.19)$$

F_1 is an integer constant which has to be optimized. For instance, $F_1 = 3$ is equivalent to a 3.3σ cut ($= \sqrt{\frac{32}{F_1}}$).

Second iteration The channels tagged by the first iteration are removed from the original pedestal corrected data by setting them to zero. A second mean value \bar{y}_2 and slope s_2 are calculated in the same way as in the first iteration. The CM found in this way is finally subtracted from the output data of the first iteration.

- CM subtraction

$$c_i = b_i - (s_2)(i - 16) - (\bar{y}_2). \quad (3.20)$$

This is the final common-mode corrected data.

Final hit detection The hits are found by comparison with a channel individual hit threshold value (T_i). Channel i is a hit if

$$c_i > T_i. \quad (3.21)$$

The optimization of T_i is described in [34].

Implementation of the CMS on a FPGA

The main constraint for the LCMS processing is the data throughput. Each FPGA receives the data of 512 strips or 16 blocks of 32 strips within 900 ns. To show that central processing is not possible one can calculate the average time to process one strip ≈ 2 ns and the number of mathematical operation per strip (≈ 15) resulting in 7 operations/ns. Modern large FPGAs provide the possibility to implement a multitude of small distributed arithmetic units which are all working simultaneously. Parallel or sometimes called distributed processing can be applied for certain applications but this is strongly dependent

on the algorithm to be processed. For the CMS data processing it can be assumed that a certain region of the detector can be processed independently from the others. This allows to divide the processing of the 512 strips in for example 16 blocks of 32 strips which can be processed fully in parallel. Distributed processing does not have an instruction code as in CPUs. Scheduling the processing is indeed more difficult. In a CPU processor the order in which an algorithm is processed is very easy to define - the first instruction in a code leads to a result and then the next instruction is executed ². In a distributed processor all instructions run simultaneously. It is imperative that the data flowing through each step of processing does not encounter any “bottle-neck” to pass to the subsequent step. The instructions for all steps need therefore to be synchronized. The scheduling is done with registers holding single intermediate results for one or several clock cycles or with memories used to store complete event data.

The pipelined processing will now be illustrated using the mean value calculation and subtraction. The following operations need to be considered:

- Read the data from the RAM blocks where it was stored after the reordering. The reordering for the ϕ -sensor required to split the data stream in several memories. The readout scheme for the inner strip requires to read data from two memories time multiplexed.
- The values of 32 strips in one packet are accumulated and the sum is stored at the end in a register.
- Intermediate storage of the 32 sample values in a RAM.
- Read samples from the RAM and subtract the average from the values.
- Pass the corrected values to the next processing step.

In figure 3.18 the necessary steps are graphically illustrated. The main difficulty to implement the design is not the arrangement of the processing steps but to organize a time schedule that fetches and processes the right data at the right time. Inserting registers in the data processing path is required to ensure proper timing of the circuit. Each logical operation leads to a small propagation delay which accumulates to several ns. The register allows to “re-synchronize” the data to the system clock for the next operation.

To schedule the processing a **pace maker** circuit provides a set of counters and control signals that can be used for all processing steps. The set of signals contains for example a counter (called CNT) that starts counting at time 0 from 0 up to 63, incremented each clock cycle and a control signal (called EN) that can be used to indicate valid data starting as well at time 0 during 32 clock cycles. The counter signal is for example used as the write address on a RAM and the enable signal as its write enable. The requirement to the electronics at level 1 specifies, that the processing needs to cope with a minimal event spacing of 900 ns. This leads to the scheduling scheme indicated in figure 3.19 using 900 ns to process two events in time multiplexed mode.

²This simple principle is only true in a very abstract view of the CPU, in reality every fast CPU works by pipelining the instructions to achieve the scheduling.

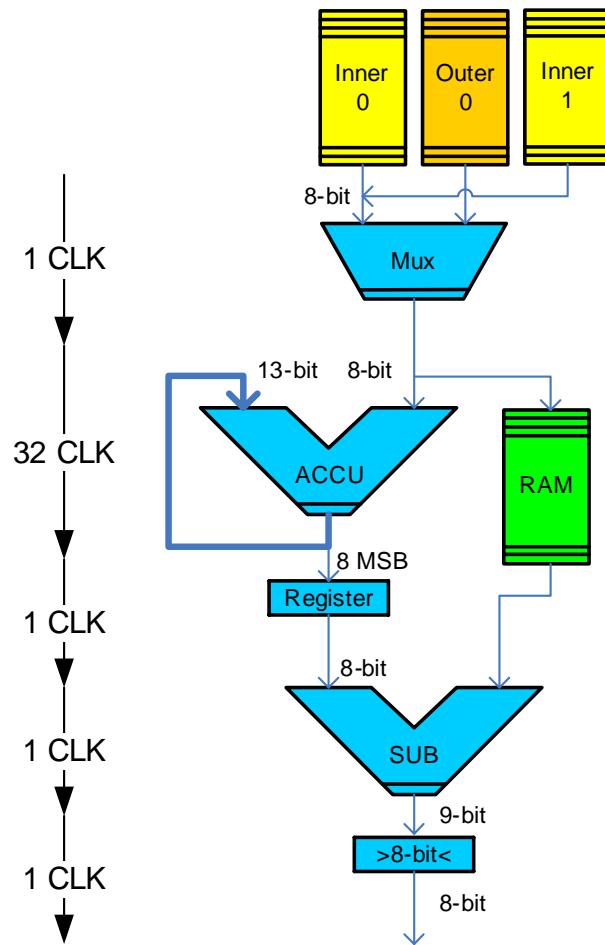


Figure 3.18: The elements necessary for the average processing are shown. Remark that in each arithmetic element a register keeps the contents during 1 clock cycle (indicated with a double line at the bottom of each). The memory address controller, the multiplexer control logic, and the logic to control the register keeping the average during 32 clock cycles are not indicated.

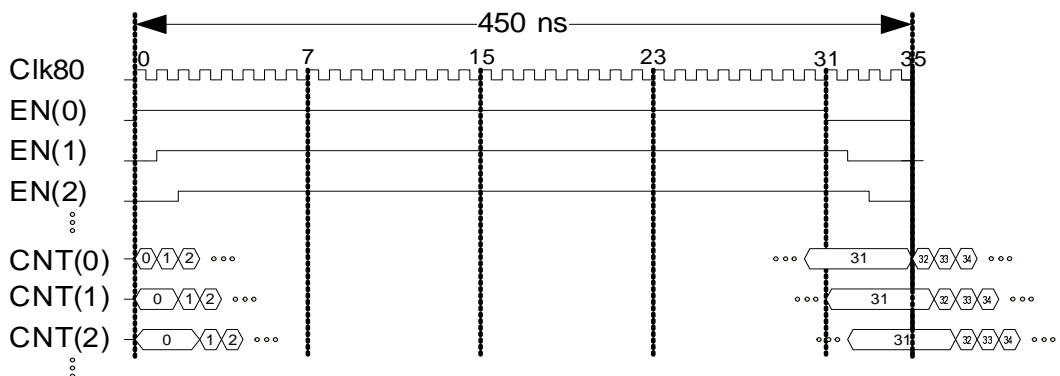


Figure 3.19: Scheduling the processing with a dedicated generator circuit “pace maker”.

The time of 450 ns is divided in 36 clock cycles (80 MHz) from which 32 cycles are used to increase the sample counters (CNT) followed by 4 idle cycles. An illustration of the processing assignments to subtract the average is given in figure 3.20. Remember that the memory data access takes two clock cycles and therefore the read address assigned on results valid data only two clock cycles later. In figure 3.20 the appropriate counters (CNT) or data enable (EN) signals are indicated with bold written number. A zero stands for a counter starting with 0 at time zero.

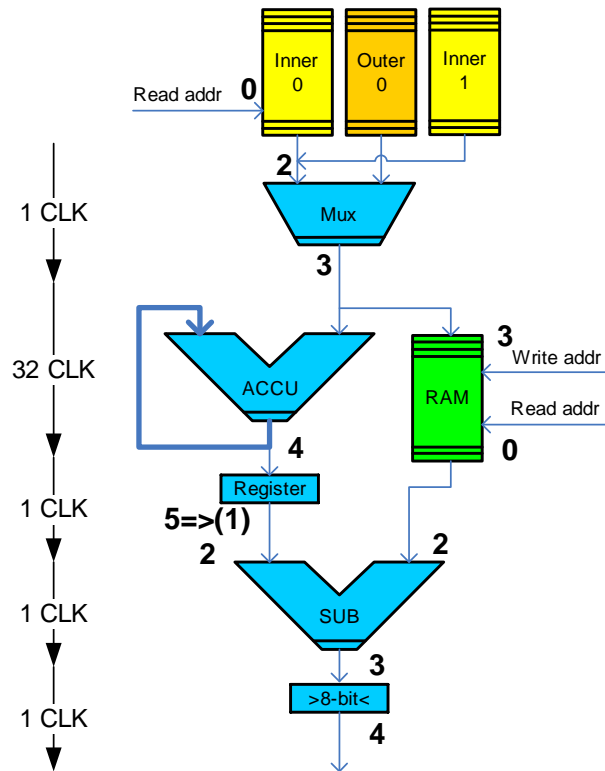


Figure 3.20: Using the pace maker signals allows to schedule the processing. The numbers in bold correspond to the timing signal that has to be used to select the necessary control signals.

The processing starts by assigning an address to the input buffers containing the data. The address used is in this case the CNT(0) signal. Two clock cycles later the data is accessible for the multiplexer where depending on the sample number the data from RAM Inner 0, RAM Inner 1 or RAM Outer 0 is taken.

The first data sample arrives at the accumulator circuit at time 3 (3 clock cycles after the address on the memory has been issued). The accumulator therefore needs to be initialized at least one clock cycle earlier (to start a new accumulation). To allow symmetric division by a simple shift operation, the reset value for a 5-bit shift operation (division by 32) should be 16. This introduces an offset that allows to have a rounding rule similar to the one implemented in standard computing systems $\frac{16}{32} = 1$ and $\frac{15}{32} = 0$. The accumulation process continues over 32 clock cycles and all samples are stored in a RAM buffer with the address generated with CNT(3) in parallel. After 32 clock cycles the sum

stored in the accumulator can be transferred to a register at the time 4. This register will contain the 8 MSBs of the sum during the next 36 clock cycles. The 5-bit shift operation is done by simple connection of the 8 MSBs to the consequent registers (shifting the bits to divide like in a CPU is not necessary). The data width of the accumulator is extended to 13-bits which allows to avoid overflows in the calculation (8-bit data summed over 32 samples can not be bigger then 13-bit). The next operation can be started immediately (indicated with $(5 \Rightarrow 1)$). Since the samples have been stored in the RAM and the earliest time it can be read is with the counter $CNT(0)$, the subsequent processing is done at time 2. The subtraction is performed between the average value (8-bit) and the 8-bit sample data with the result being a 9-bit value. In a last step the data range is limited to 8-bit where a saturation operation is used. The formal description for 8-bit limitation is:

$$Limit8(x) = \begin{cases} 127 & x > +127 \\ x & -128 \leq x \leq +127 \\ -128 & x < -128 \end{cases} \quad (3.22)$$

In conclusion, the introduction of a set of scheduling control signals and an appropriate notation has been seen very useful for implementing the LCMS algorithm.

VHDL coding

The example of the average subtraction is further used to show how the circuit is defined in the hardware description language. To simplify the VHDL code no interface, components or signals are declared in the example. Architecture and Entity declaration are also left out to concentrate on the “important” assignments. The processes listed are simultaneously executed (concurrent processing) and are clocked with the processing clock clk_80 . A chip wide reset signal called “reset” defines the startup condition. With the first lines of code “Assignments to Input RAM”, the read address, write address, write enable and write data signals are assigned to the RAM blocks not shown in the code.

```
-----
-- Assignments to Input RAM
-----
ram_inner_0_raddress <= CNT(0);           -- Assuming the data
ram_outer_0_raddress <= CNT(0);         -- in stored appropriate
ram_inner_1_raddress <= CNT(0);         -- order

ram_1_wraddress      <= CNT(3);         -- Memory to store data
ram_1_wren           <= EN(3);         -- while average
processing ram_1_raddress <= CNT(0); ram_1_data <=
data_mux_out;
```

In the following part of the code we show the assignments to infer the processing logic. Note that the VHDL code has been simplified for the input multiplexer to the situation where the addressing of the memories can be done in consecutive order. Unfortunately this is not true as it was shown in figure 3.16.


```

-----
-- Multiplexer assignments
-----
set_mux: PROCESS (clk_80,reset) begin
  if reset='1' then
    data_mux_out <= X"00"; --hex
  elsif rising_edge(clk_80) then
    if CNT(2) < 21 then
      data_mux_out <= ram_inner_0_q; -- the first 21 samples from inner_0
    elsif CNT(2) < 32 then
      data_mux_out <= ram_inner_1_q; -- the next 11 samples from inner_1
    else
      data_mux_out <= ram_outer_0_q; -- the second 32 samples from outer_0
    end if;
  end if;
end process; -- data_out_mux available at (3)
-----
-- Accumulator assignments
-----
set_accumulator: PROCESS (clk_80,reset) begin
  if reset='1' then
    sum_1 <= "0000000010000"; --bin
  elsif rising_edge(clk_80) then
    if EN(3)='1' then
      sum_1 <= sum_1 + data_mux_out; -- accumulate
    else
      sum_1 <= "0000000010000"; -- reset sum
    end if;
  end if;
end process; -- sum_1 available at (4)
-----
-- Average assignments
-----
set_average: PROCESS (clk_80,reset) begin
  if reset='1' then
    average <= (others => '0');
  elsif rising_edge(clk_80) then
    if CNT(4)(4 downto 0)= 31 then
      average <= sum_1(12 downto 5); -- shift 5-bit to get average
    end if;
  end if;
end process; -- average available at (5) => (1) but 2
-----
-- Average subtraction assignments
-----

```

```

set_average_subtraction: PROCESS (clk_80,reset) begin
  if reset='1' then
    data_1 <= (others => '0');
  elsif rising_edge(clk_80) then
    data_1 <= SXT(ram_1_q,9) - SXT(average,9); -- sign extended (SXT)
                                                -- values subtracted
  end if;
end process;    -- data_1 available at 3
-----
-- Limit data to 8-bit width assignments
-----
set_data_2: PROCESS (clk_80,reset) begin
  if reset='1' then
    data_2 <= (others => '0');
  elsif rising_edge(clk_80) then
    if data_1 < -128 then
      data_2      <= "10000000"; -- Limit8 operation
    elsif data_1 < 127 then
      data_2      <= data_1(7 downto 0);
    else
      data_2      <= "01111111";
    end if;
  end if;
end process;    -- data_2 available at 4

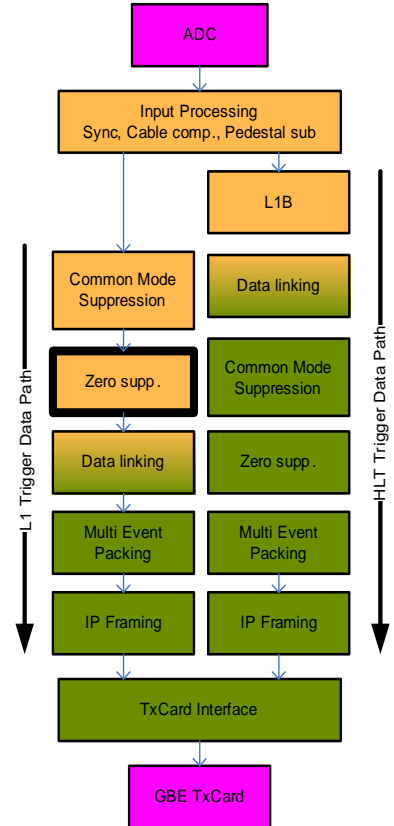
```

Resource usage

During the prototyping phase for the L1PPI [34] and RB3 [24], an estimation for the required resources was given based on Altera APEX [38] FPGA data. By using the Stratix FPGA family, many changes to the architecture have taken place (see section B). To have a better estimate for the current design, 8 channels of L1-CMS have been implemented together with the complete HLT data (9 channels are required for the ϕ -sensor). The missing part were the L1T zero suppression and linking only. The chip resources used was 65% of the logic elements and memory blocks. The balance between logic elements and memory usage can be adjusted by implementing arithmetic operations as fixed coefficient multiplications in RAM blocks as a lookup table or with logic gates (logic elements). All DSP multiplier blocks were used. The two large M-memory blocks cannot be used because they are reserved for the “PCN correction”, already seen in section 3.2.5, and by the ‘L1T data derandomizer’ which will be discussed in section 3.3.4.

3.3.2 L1T Data zero suppression (sparsification)

The zero suppression processing block receives the common mode corrected data in form of 8-bit sample values. The low occupancy of the vertex detector makes zero suppression an obvious way to reduce the data. Moreover the L1 trigger algorithm requires anyway clusterized data. Several different cluster encoding schemes were studied to minimize the data to be transferred over the event building network to the L1 trigger computer farm [35]. The format of cluster encoding is strongly dependent on the number of bits that are available per cluster. In the case of the VeLo sensors, the 2048 strips can be encoded in a 11-bit number. Since the clusters have to be transferred over a 32-bit word oriented network system, a cluster size of 16-bit is a reasonable choice. The simulation of the distribution of the expected cluster size shows that only a very small fraction of the clusters is bigger than 2 strips. The distribution given in figure 3.21 shows that only 3.5% of R and 1% of ϕ clusters have size bigger than 2 strips. The choice of encoding clusters with size up to 4 strips makes use of 2 bits for the cluster size encoding. Clusters are only formed within a packet of 32 strips processed together in the CMS algorithm, clusters located on a boundary are not linked together. An additional bit is used to indicate if one of the strips in the cluster has exceeded a second threshold. Two bits are free, leading to the cluster format:



Bit	Description
< 1 : 0 >	Cluster size Is 0 for clusters of one hit and 3 for four hits
< 12 : 2 >	Strip number Unique strip number per board (2048)
13	Second threshold If one of the hits in the cluster exceeded the second threshold level this bit is set.
< 15 : 14 >	Unused
16-bit	Total

Table 3.1: Velo cluster format for the L1 trigger.

The implementation of a basic cluster algorithm is done in three steps:

1. Search for hits exceeding the first or the second level threshold. Two hit masks of 32 bits each are created. The original strip values are not required anymore.
2. Clusters of maximal 4 strips are formed and the position of the first hit with the first threshold (lower threshold) is encoded in 5-bit (the position is encoded relative

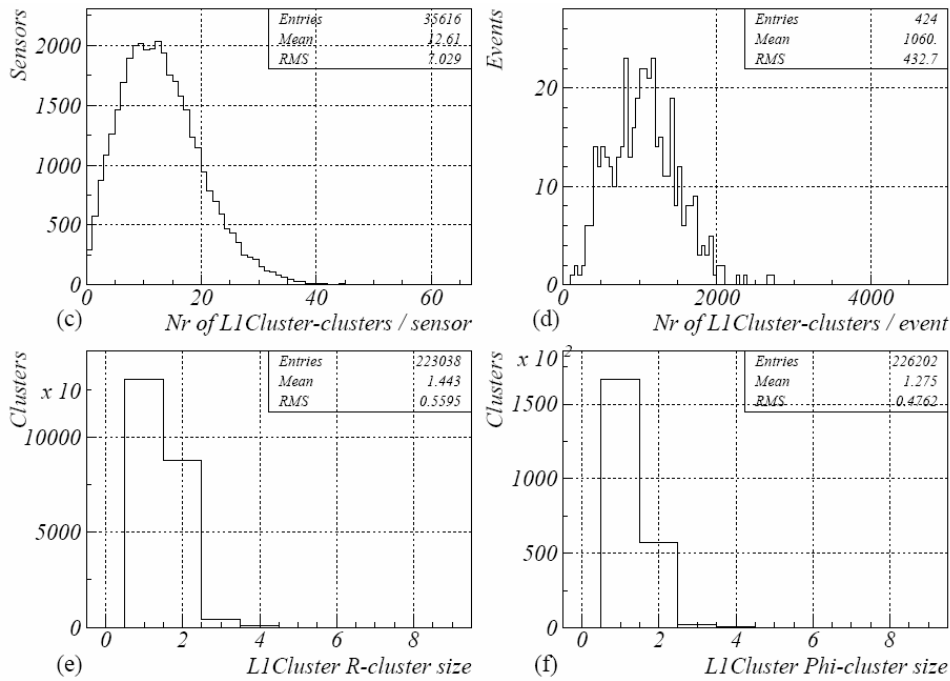


Figure 3.21: *Top: number of L1 cluster per sensor and event. Bottom: cluster size for R and ϕ sensors.*

to the 32 strip segment). The size of the cluster is added as well as the second threshold information.

3. The clusters are written into a FIFO.

A possible procedure to improve the clustering is to search for hits in two iterations. In a first search, a quite high threshold is applied to identify the hits that have a high “good” signal to noise ratio. This allows to keep the noise clusters low. In a second step, the neighboring strips to the found hits are searched for signal with a less stringent cut to find the complete cluster signal. The total charge deposit in a cluster can be calculated by summing signal of all strips in a cluster. The cut whether a cluster is accepted or not can then be done on the basis of the total signal of the cluster compared to a threshold for the seed strip. This can be implemented generating a third hit mask as already done in step 1. The clustering (step 2) always need a seed hit (first hit mask) and then adds possible neighboring hits from the third mask.

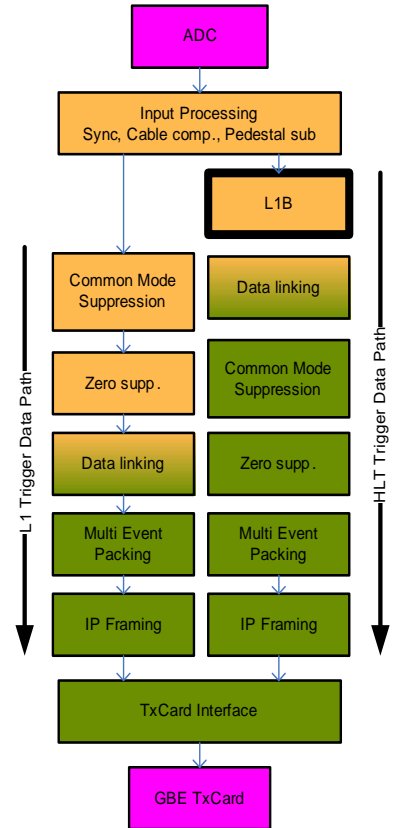
The implementation of the clustering algorithm does not look critical regarding design and resource usage. Additional input from the detector physics is needed to optimize the procedure. The clustering for the VeLo has been studied during test beam data analysis [36].

3.3.3 L1 buffering

(See also the dedicated section 5.5) With the final choice of DDR SDRAM memory for the L1 buffer, various aspects of the implementation need to be considered for the design. In

fact it occurs that apart from the CMS algorithm, critical in terms of resources usage, the L1 buffer controller is the most challenging object on the TELL1 board. The difficulties come from the physical interface between the FPGA and the memory chip, the tight timing constraints on the interface of the user logic to the memory controller (DDR SDRAM IP-Core), and last but not least the arbitration of the read and write operation due to the fact that only one data access port exists.

The access performance to the buffer needed is specified in the L1 front-end electronics requirements [31]. The implementation has to cope with a write rate defined by the minimal L0 event spacing of 900 ns and the read data rate given by the minimal L1 accept spacing of $20 \mu\text{s}$. The average read/write ratio of $1/25$ suggests that the bandwidth of the memory is not simply divided in two equally long time slots one for read and one for write, but to schedule (arbitrate) the memory in a more intelligent way. Reducing the required memory bandwidth leads to less IO usage and therefore a smaller and less expensive implementation. The difficulty with scheduling the read and write operations is that the data stream needs to be sufficiently buffered. To define the size of the buffers, simulations have been performed. The usage of a random access memory instead of a FIFO buffer has opened the possibility that the L1 buffer content is made accessible for read and write operations by ECS. This is particularly helpful for system debugging and low speed data acquisition. During the system test phase, simulated data can be inserted in the buffer to test the proper functionality of the whole downstream processing and event building network, not requiring any detector data. The ECS access nevertheless does also complicate the implementation. In addition to L0 and L1 data the buffer access has to cope also with the ECS read and write requests. The timing constraints on the chip, the L1 requirements and the ECS access, together with the optimization for resources has lead to the implementation shown in figure 3.22.



With a first FIFO buffer stage the pedestal corrected data is grouped in 3×32 -bit wide data streams. Remark that the architecture of the subsequent steps has to be designed for a TELL1 equipped with optical receiver cards. The optical readout allows for the acquisition of 6 data streams (32-bit at 40 MHz) in contrast to the analog which is restricted to 4. For the VeLo the 8 data streams after the pedestal subtraction (8×8 -bit at 80 MHz) occupy only 2 out of the 3 available FIFO buffers³. The task of the input FIFO stage controlled by the “L1B Data Sender” is to change the clock domain from 80 to

³The additional 9th processing channel formed for the ϕ sensors after reordering requires storage in the 3rd FIFO buffer.

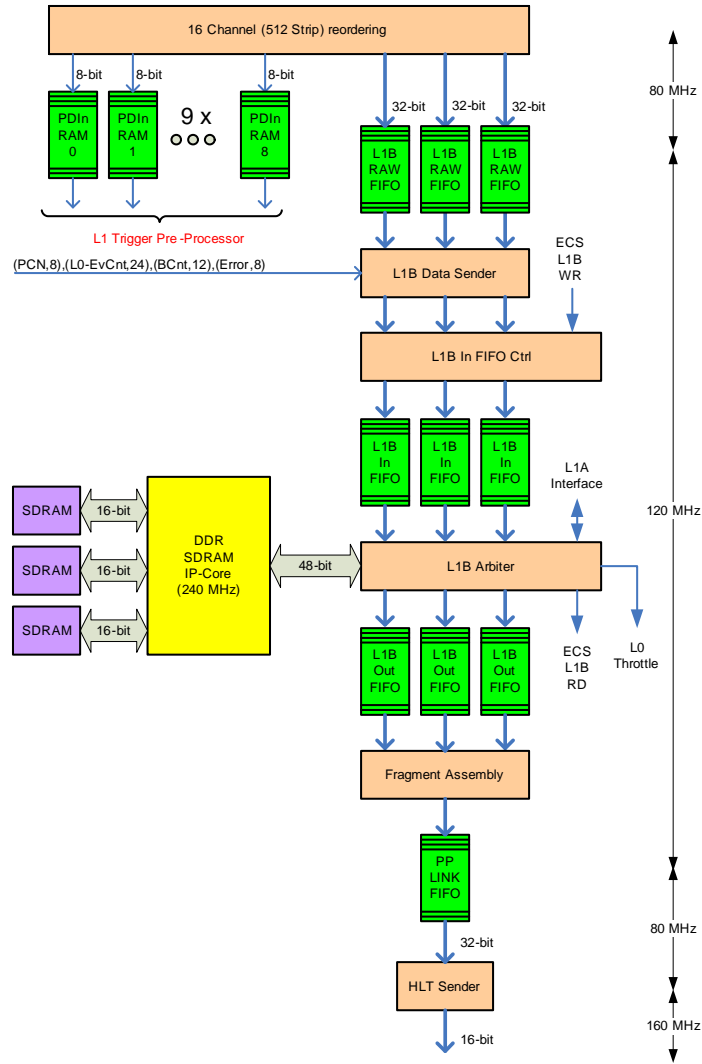


Figure 3.22: Block diagram of the L1 buffer implementation.

120 MHz⁴ and to alter from a push to a pull data transfer protocol. The later is required for the header information extraction, read from the “Header Pipeline” (see figure 3.4) and the ECS write access arbitration. A data format common to all sub-detectors is used after this processing stage [37]. The FIFOs at this stage are not required to hold much data. The read access at 120 MHz allows to keep the FIFOs as small as 64 words.

The arbitration between read and write accesses is implemented in two stages. The timing requirements for the user interface to the DDR SDRAM IP-Core does not allow a multiplexed write data path which is required if the L0 data and the ECS data needs to be written to the buffer. The multiplexing stage is therefore pushed on the input data path to the derandomizer buffer “L1B InFIFO” required for the read to write arbitration. The write data access to the buffer is organized in the following way: the monitor of the fill state of the “Ped FIFO” indicates to the “Data Sender” when sufficient event data is

⁴In 3.4 a detailed clock cycles count for each memory access including memory refresh and all specific timing parameters of the SDRAM have lead to the necessary clock frequency.

available. As soon as the last access to the “L1B InFIFO” is terminated (the “L1B InFIFO Ctrl” indicates to the “Data Sender” that a new event can be assembled and written to the “L1B InFIFO”. The “L1B InFIFO Ctrl” arbitrates the ECS write and L0 event write accesses. With the first word, written to the “L1B InFIFO”, the “L1B Arbiter” receives the write start address and the number of words to be written to the L1B. It is the “L1B Arbiter” that schedules between the “L1B InFIFO”, the L1 accept event data read and the ECS read. The state-machine also generates the user interface signals for the DDR SDRAM controller including for example the physical address and the data mask for masking valid data on the large (96-bit wide) data bus.

Over a dedicated link between the SyncLink-FPGA and the PP-FPGA the L0-EvCnt for a L1 accepted event is transmitted to the “L1B Arbiter”. The readout of the event is scheduled with low priority such to avoid the risk of “L1B InFIFO” buffer overflows. The total bandwidth on the buffer exceeds by 50% the write bandwidth needed, leaving a sufficient margin for the L1 accept readout, memory refresh and ECS accesses. A detailed clock cycle Register Transfer Level (RTL) simulation shows that the chosen buffer size for the “L1B InFIFO” of 256 words is sufficient. Figure 3.23 shows the fill state of each

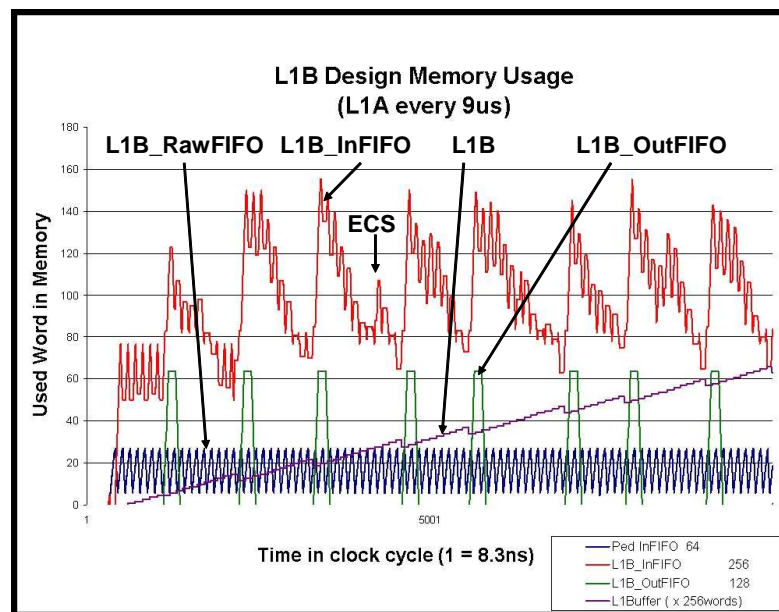


Figure 3.23: The graph shows the buffer occupancy for the L1B control logic. It was recorded stressing the system with a readout spacing for L1 accepts of $9 \mu\text{s}$.

FIFO buffer of figure 3.22. The simulation was done with 900 ns L0 event spacing and a L1 accept rate of $9 \mu\text{s}$ to observe the behavior of the buffers submitted to a much bigger load than the one needed in the experiment. The effect of infrequent ECS accesses can be seen at for example 4700 clock cycles where the “L1B InFIFO” increases by about 20 words. The Level 1 buffer fill state is calculated by the difference of the current write to read address for L0 and respectively L1 accepted events.

The L1 accepted data will be called “HLT data” from now on. The bandwidth required for the HLT data is about 25 times smaller than the L1 and a second data multiplexing

stage can be inserted. The data distributed over three 32-bit wide FIFOs are merged into one named “PP Link FIFO”. For this step the header data is merged and a new data format is used. At this stage no sub-detector specific processing or data format is needed anymore. The event has a fixed length.

L1 Buffer physical addressing scheme

To simplify the addressing scheme a memory space of 64 words is allocated to store the 36 data words of one fraction of an event. This scheme allows to use simple bit manipulation to calculate the start address for each event from the L0-EvCnt. The memory space with the current memory configuration provides therefore space for 64 k events which requires 16-bit of the L0-EvCnt as event base address. For a future implementation the memory space can be filled up with events spaced 36 words with the drawback that the base address of each event has to be calculated by a multiplication. The “expensive” multiplication luckily can be avoided, since the multiplication with 36 can be reduced to one simple addition. The binary representation of 36 has only two non zero binary digits leading to

$$\text{BaseAddr}(\text{EvCnt}) = \text{EvCnt} \times 2^5 + \text{EvCnt} \times 2^2 \quad (3.23)$$

where the multiplications with 2^5 and 2^2 are trivial.

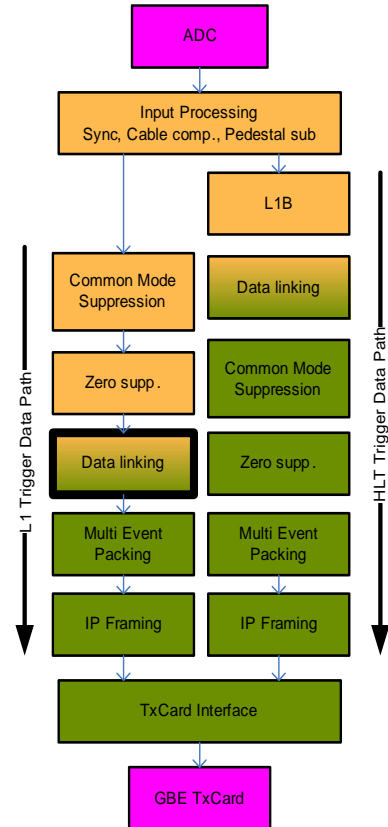
3.3.4 L1T Data Linking

The clusters written by the last stage of the zero suppression are available in 9 FIFOs on each of the 4 PP-FPGAs. In this section the design used to merge these fragments into one data stream is described. The linking stage needs to cope with the variable number of clusters contributed by each processing channel as well as with the restricted bandwidth on the links connecting the PP-FPGAs to the SyncLink-FPGA. The outline of the design is as follows: the concept of a fixed latency processing as in the CMS and the Zero Suppression is continued for the first linking stage situated on the PP-FPGA. With an average of 12.6 expected clusters per sensor per event (3 per PP-FPGA) a design for the theoretical maximum ⁵ is not appropriate. An ECS configurable maximum number of clusters per PP-FPGA being accepted is implemented. Any additional clusters are discarded and the incident is flagged in the data header. In order to simplify the timing constraints on the linking on the PP-FPGA the maximum is set to 64, leaving more than one clock cycle at 80 MHz per cluster. The “L1T Cluster Link” state-machine links the clusters from the 9 input FIFOs, extends the cluster format to the 16-bit format described in table 3.1 and discards clusters in case of necessity. To deal with the variable number of clusters per event the first word in all cluster FIFOs contains the number of clusters in that event. The data transfer between PP and SyncLink-FPGA is implemented with a 16-bit wide point-to-point link running at 160 MHz data transfer rate (2.5 Gbit/s bandwidth). The bandwidth of one of the 4 links is sufficient to fully load the 3 dedicated Gigabit Ethernet channels at the interface to the event building network. Note that the average load on the Gigabit Ethernet links should not exceed ~ 0.8 Gbit/s. This reveals that

⁵The worst case is 256 one strip clusters in one PP-FPGA. There is little chance that an event with this local occupancy of 50% can be analyzed by the L1 trigger in a reasonable time.

the FPGA interconnection does not add any limitation. The subsequent data merging is implemented on the SyncLink-FPGA. If no other preventive measure would be taken, the linking had to be done at a frequency of 4×2.5 Gbit/s to cope with the arriving data streams. To avoid this, the links are equipped with a large de-randomizing buffer on the link source side. The buffer is sized large enough to sustain at least 64 worst case events (events with 64 clusters). This ensures that this buffer can be protected from overflows by the L0 throttle mechanism which is retarded by a maximum of 32 events ⁶. The total size required is $(64 \times 64 \times 16\text{-bit})$ 64-kbit which is available in one of two large M-memory block (512-kbit) on the Stratix FPGA. Having the data streams passed through the de-randomizers simplifies the next step of processing. The linking on the SyncLink-FPGA can be adapted to the available bandwidth on the readout network and requires a maximum of 2.5 Gbit/s. This can be implemented with a state machine pulling the data from the 4 PP-FPGA links and multiplexing them into one data bus 32-bit wide at 80 MHz. The structure of the linking design is shown in figure 3.24.

The input FIFO buffers on the SyncLink-FPGA are inserted to allow for the HLT and the L1T data transmission design to be identical. The possibility to change the clock domain and to do detector specific processing is also provided. The output of the L1T linker is split up in two separate FIFOs, one for the event data and one for the total size information which is needed for the assembly of a Multy Event Packing (MEP) (see section 3.5).



⁶16 events in the L0 derandomizer and about as many on the link and in the TELL1 CMS.

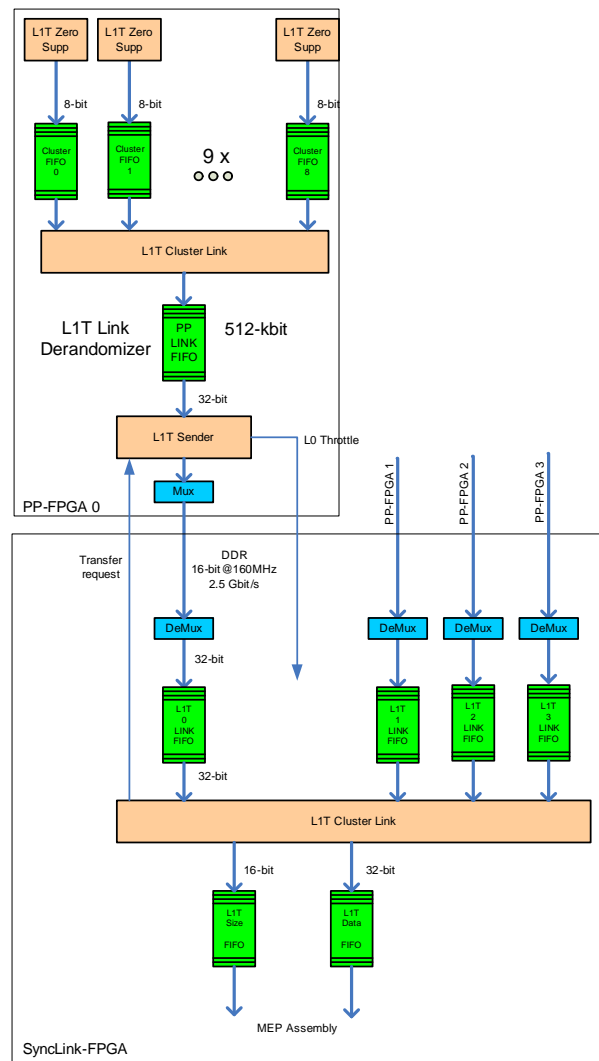
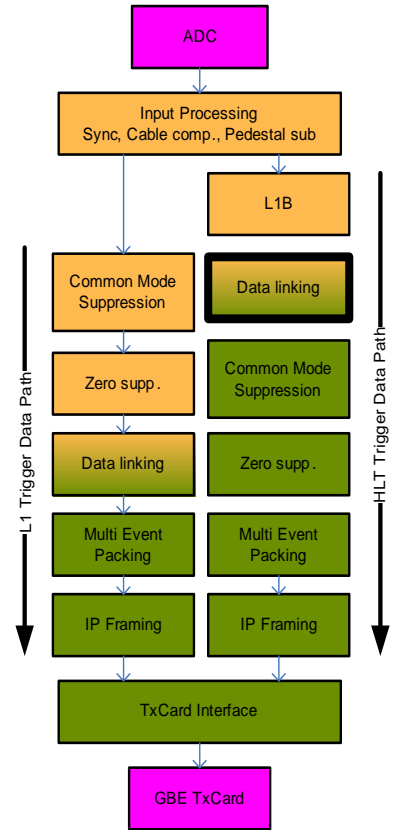


Figure 3.24: *L1T* data linking stage from the 4 *PP-FPGAs* to the *SyncLink-FPGA*.

3.4 Data processing for the HLT

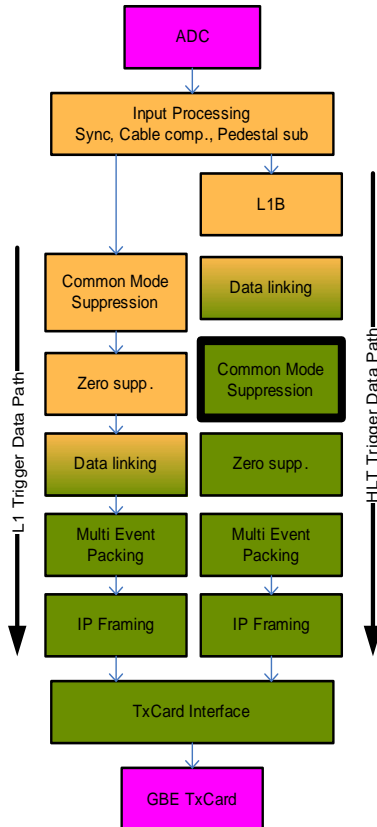
3.4.1 HLT Data Linking

Linking the HLT data is partially done already in the multiplexing stage after the L1B. The header data of the entire PP-FPGA event data is assembled. Since the data after the L1B is not zero suppressed a fixed length fragment format has been defined at this stage. The transmission to the SyncLink-FPGA is buffered with a FIFO sized for one single event only. The fixed minimal event spacing and the fixed size does not impose larger buffering. The transmission and linking on the SyncLink-FPGA are identical to the one for the L1T data. Because only one event can be stored in the output FIFO buffer a data transfer request from the SyncLink-FPGA is not necessary. The transfer is initiated automatically as soon as the event is assembled. The minimal event spacing of $20 \mu\text{s}$ ensures that the data transfer can be done event by event without buffering of multiple events.



3.4.2 HLT Common Mode Suppression

The HLT CMS processing can be performed with relaxed requirements compared to the L1T, because of the 25 times lower event rate.



From the physics point of view, the data transferred to the HLT must contain all necessary information to reconstruct the L1T decision. This requires that the data previously sent to the L1T needs to be either stored in the L1 buffer and read out after a L1 accept or it can be recalculated using a CMS algorithm after L1 buffer storage. The two options must be discussed in some more detail.

The information sent to the L1 Trigger contains 2-bit information per detector strip (one for each threshold) which need to be either stored or recalculated. To store this information in the L1 buffer, 25% more bandwidth on the buffer is required. The implementation with the additional data and its arbitration by the L1B controller have been simulated. It showed the necessity of an increased L1B operation frequency. The most convenient way to recalculate the L1 CMS is to reuse the same FPGA code again and process it on the SyncLink-FPGA. To take advantage of the 25 times larger event spacing the time multiplexing of 32 strip packets can be extended from the 2 used in the L1 CMS to 18⁷ in the HLT CMS. At the present understanding, increasing the precision on the LCMS algorithm does not seem to give extra benefit compared to the L1T case [34].

It is therefore most likely that the HLT CMS can be done identical to the L1T CMS, without any additional processing.

DSP based HLT CMS processor

The implementation of the processing for the HLT CMS has been studied concerning the feasibility to perform the processing on either FPGAs or dedicated Digital Signal Processors (DSPs). The major advantages of the DSP over the FPGA implementation are the availability of higher precision arithmetic (32-bit) and the reduced complexity for the algorithm implementation. The implementation of the LCMS algorithm has been prototyped and bench marked on a Texas Instruments TMS320C6211 fixed point DSP [39]. The results have been summarized in [40] and the final conclusions are:

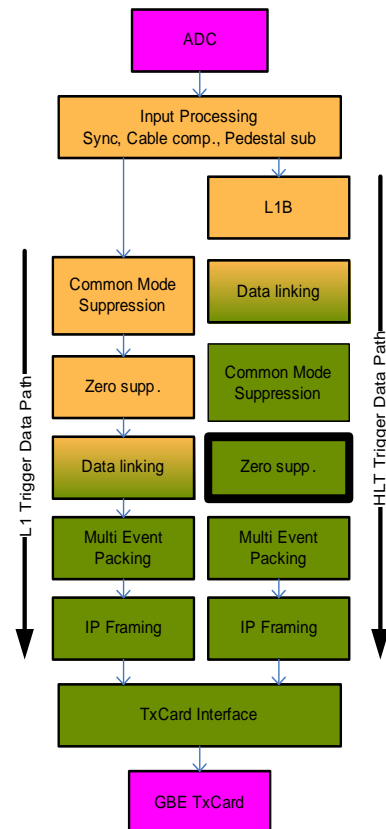
- The event rate of 40 kHz forces the programmer to include optimized assembler code for the data transfer from the L1B to the DSPs internal data memory. Note that the implementation of the interface from the FPGA to the DSP was chosen to be a shared memory access of the L1B directly. This has the disadvantage of tight buffer access constraints. A more relaxed solution could be to transfer the data via dedicated FIFO buffer on the PP-FPGA.

⁷The ϕ -sensor provides 18 packets of 32-strips (9 L1T CMS processing channels) per PP-FPGA.

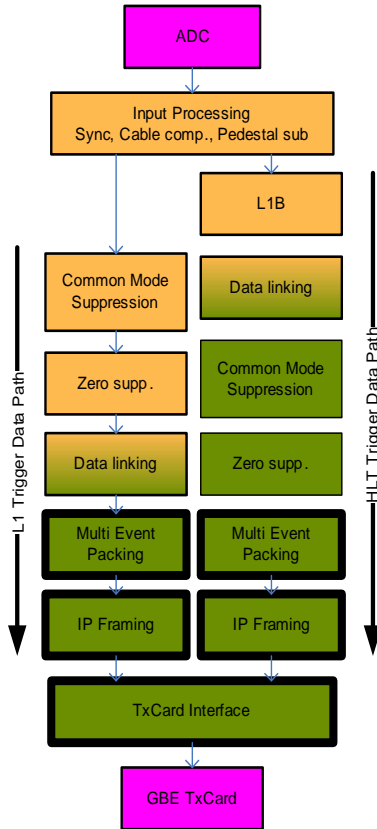
- The processing power of the Texas Instruments high end DSP family is sufficient for processing multiple channels but not for a complete TELL1. Several DSPs are needed on one board. With the more recent available DSPs, TMS320C64x this remains true.
- The DSP solution introduces tight space constraints for the PCB which has additional large (300-pin BGA) chips to accommodate.
- For the long term maintenance, a single technology (FPGA only) motherboard is easier to handle.

3.4.3 HLT Zero Suppression

The calculation of the precise position of a track in the detector requires the track angle and the charge distribution seen in a cluster. Therefore the HLT data must contain the 8-bit sample values of the strips in a cluster. Therefore the HLT cluster format carries the cluster position, the cluster size and the cluster strip values. In contrast to the L1T cluster format, this strongly suggest to have a variable length cluster encoding dependent on the cluster size. In addition to the HLT clusters, the recalculated L1T information is added with the same zero suppression scheme as for the L1T.



3.5 Multi Event Packing (MEP)



The performance of an Ethernet based network is strongly dependent on the data packet size and its transmission rate. To cope with the challenging first level trigger rate of more than 1 MHz, different solutions have been presented in the past. The first approach using the so called “Readout Unit” [41] was based on the principle to reduce the number of packets sent into the event builder by introducing a first multiplexing stage between the TELL1 board and the commercial event builder network. The multiplexing stage was designed to aggregate the event data from several boards but did not reduce the 1 MHz event rate. The transmission between TELL1 and the RU was defined to be an S-Link [43] based point to point connection. For the event builder an SCI [44] network with two dimensional torus interconnection and scheduling with tokens was developed [45]. In a second approach a Network Processor (NP) based readout unit using Gigabit Ethernet for input and output interfaces was outlined [46].

Finally, the idea to assemble multiple event fragments of consecutive events in the TELL1 into Multi Event Packet (MEP) has been adopted. In order to lower the packet rate on the network, the data from several events on the TELL1 are kept in a buffer. The accumulated data is then sent in one packet to the event builder which profits from the reduced packet rate and less protocol overhead. The packed events are formatted complying to the Internet Protocol (IP) standard and injected via Gigabit Ethernet [47] into a network built on commercial GigaBit Ethernet (GBE) equipment.

Implementation

Two almost identical MEP and Ethernet framer channels can be employed for the L1T and HLT data streams. The bigger event size and the resulting MEP packets for the HLT data, requires a larger MEP buffer. The necessary buffer size can only be achieved with an external memory. To simplify the buffer access and its control, an external QDR SRAM [48] with independent read and write ports is used. A fast double data rate interface makes this memory ideal for this application. For the L1T MEP buffer one of the large M-RAM blocks of 512 kbit size is used. An overview of the design blocks implemented for the MEP assembly is given in figure 3.25.

The implementation is described using the L1T data stream. The linking stage, “L1T Cluster Link” provides the event size and the cluster data via two FIFO buffers to the “L1T MEP Assembly”. The data flow through the consequent design separates header and data information. The interface to the MAC device on the Gigabit Ethernet TxCard

is based on the SATURN Development Group’s “POS-PHY Level 3” interface [50], which will be referred to hereon as the SPI-3 interface.

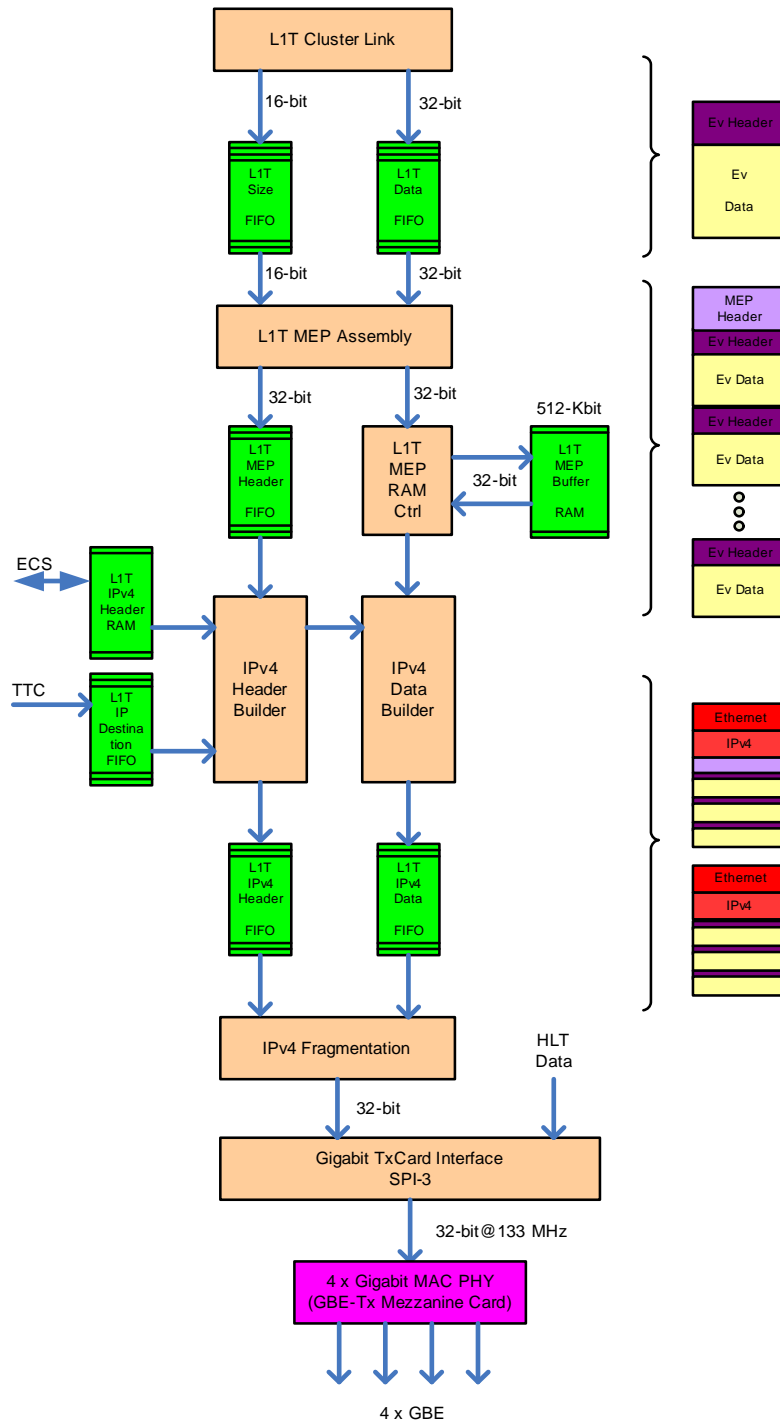


Figure 3.25: Block diagram of the MEP assembly and buffering followed by the Ethernet framing and fragmentation. The L1T and HLT data path to the common mezzanine card interface is arbitrated by the SPI-3 controller.

The MEP assembly can be described as follows: The MEP assembly starts as soon as

the first event has completed the zero suppression and the size of the event is available in the ‘L1T Size FIFO’. The start address of the MEP is written to the ‘MEP Header FIFO’ which serves as readout pointer for the next stage. The event data is written to an on chip buffer (512-Kbit) large enough to store two complete MEPs⁸. The number of events to pack in a MEP ‘Packing Factor’ can be configured via ECS. After the last event of a MEP has been written to the ‘MEP Buffer’ also the size of the complete MEP is written to the ‘MEP Header FIFO’.

In the next stage, the Ethernet and IPv4 protocol is wrapped around the data. The format of the header and the generation of the standardized packets is documented in [49]. The ‘IP Header Builder’ needs the information from three sources: first, all constant values as IP source address and header type set via ECS to the ‘IPv4 Header RAM’; second, the IP destination address assigned dynamically using the TTC system to the ‘IP Destination FIFO’; and, third, the ‘MEP Header FIFO’ providing the total MEP size.

The subsequent Ethernet fragmentation process fragments the total MEP data into Ethernet standard (1500 bytes) or ‘Jumbo’ (9000 bytes) frames. The framing process is launched only if data is in the MEP buffer and the SPI-3 is available. The storage of framed data is not foreseen!

The final stage ‘SPI-3 Ctrl’ arbitrates the L1T and HLT data streams. It assigns the physical destination port (4 Gigabit links are available on the mezzanine card) used by the Media Access Controller (MAC).

3.6 L0 and L1 throttling (buffer overflow prevention)

The prevention of any buffer overflow is essential for a large system. The Readout Supervisor (RS) [29] schedules the trigger decision which are distributed over the Timing and Fast Control (TFC) [28] system to the readout electronics. To prevent buffer overflows in the readout two so called ‘Throttle networks’ are implemented as a feedback path to the RS. A ‘Throttle OR’ module to collect L0 and L1 throttles is foreseen to be plugged in each TELL1 crate. The generation of the throttle signals on the board in the different processing locations is shown hereafter.

3.6.1 L0 throttle

Stopping the acceptance of events at the L0 trigger level will not stop the arrival of events at the TELL1 board front-end links immediately. The response time of the RS (4 events), the signal transmission time to the front-end (1 event), the events stored in the L0 de-randomizer (16 events max) and the data already in the links to the TELL1 board (1 event) cause a total delay which will not be bigger than 22×900 ns. The events stored in the pipeline of the processing need also to be counted as an additional delay. The total response time therefore is of about 32 events at the output of the L1T zero suppression.

Throttling is not needed on the Input processing, CMS, Linking and L1T zero suppression because the pipelined processing is paced with 900 ns event rate. Remember

⁸At least two MEPs need to be buffered since assembly and transmission are concurrent.

that the maximal number of clusters for the VeLo has been limited such that the clusterization can also be done within 900 ns. The subsequent processing depends on the data flow through the SyncLink-FPGA to the event builder. In case the readout network is overloaded events are accumulated in the “PP Link FIFO”. It is this buffer that needs overflow protection by the L0 throttle. The threshold needs to be set in such a way, that 32 worst case zero suppressed events can still be stored in the buffer without overflowing. Such a worst case event consists of 64 16-bit cluster words, equal to 128-Byte. The buffer space required until an asserted throttle takes effect is 4 kByte. The L1 buffer fill state is monitored by the “L1B Arbiter” which can also assert a L0 throttle. The supervision of the L1B fill state can be used to monitor the correct emulation by the RS.

3.6.2 L1 throttle

The L1 throttle is used to indicate a potential buffer overflow on the HLT data path to the RS. On the data path from the L1 buffer to the zero suppression on the SyncLink-FPGA the event rate of 40 kHz is guaranteed and therefore the data can not accumulate in any buffers. In case the readout network is overloaded, the L1 accepted data is written in the MEP buffer and a throttle limit has to be set to ensure the buffer never overflows. With a maximum L1 throttle latency of 2 μ s, enough space for one more L1 accepted event has to be available in the MEP buffer after asserting the throttle. This doesn't pose any problem with the 2-Mbyte external buffer.

Chapter 4

TELL1 for optical readout

The adaptation of the TELL1 to the optical readout is implemented using mezzanine cards for optical receiver and de-serializer stage. Two Optical Receiver Cards (O-RxCards) [51] per board are designed for the reception of one 12-way optical fiber ribbon each. The link operates at 1.6 GHz, transmitting 1.28 Gbit/s user data per optical fiber. The total receiver bandwidth is $24 \times 1.28 \text{ Gbit/s} = 30.7 \text{ Gbit/s}$. The chosen link technology makes use of the radiation hard CERN GOL [52] serializer chip on the transmitter and from the Texas Instruments TLK2501 SERDES [53] on the receiver side. Each of the optical links can be used for the acquisition of a data stream of 32-bit at 40 MHz. In this chapter, only the important differences from the VeLo data processing are discussed.

4.1 Input data processing for the optical readout

The processing for each sub-detector can differ at the input stage but the basic principle remains unchanged. Each de-serializer chip sends data synchronized to the clock. The clock signal is recovered from the data received over the optical link. The data bus is 16-bit wide and runs at 80 MHz (“TLK Interface”). The valid data is indicated with the “data valid” signal generated by the de-serializer. After being de-multiplexed to 32-bit it is written to the “Input Data FIFO”. Using the reference L0-EvCnt and BCnt (available from the local TTC receiver), the correct event synchronization can be verified and accordingly flagged in the event data header. To prepare for the subsequent processing, the header information is written in a dedicated memory called “Header Pipeline” from where the header data can be accessed later. The separation of the event data from the header data can be avoided if no information is sent to the L1 Trigger (see figure 4.1).

Pedestal subtraction and calibration

The pedestal subtraction can be done as a first step after the event synchronization. Two ways to calculate the pedestals were introduced for the VeLo processing. Time multiplexed processing is also considered to save resources at this stage. The “Pedestal Subtraction” block can be replaced for example by a detector specific calibration of the event data. For instance the calorimeter data will be calibrated at this stage. If no special processing is required, the interface to the L1 buffer may also be done directly after the

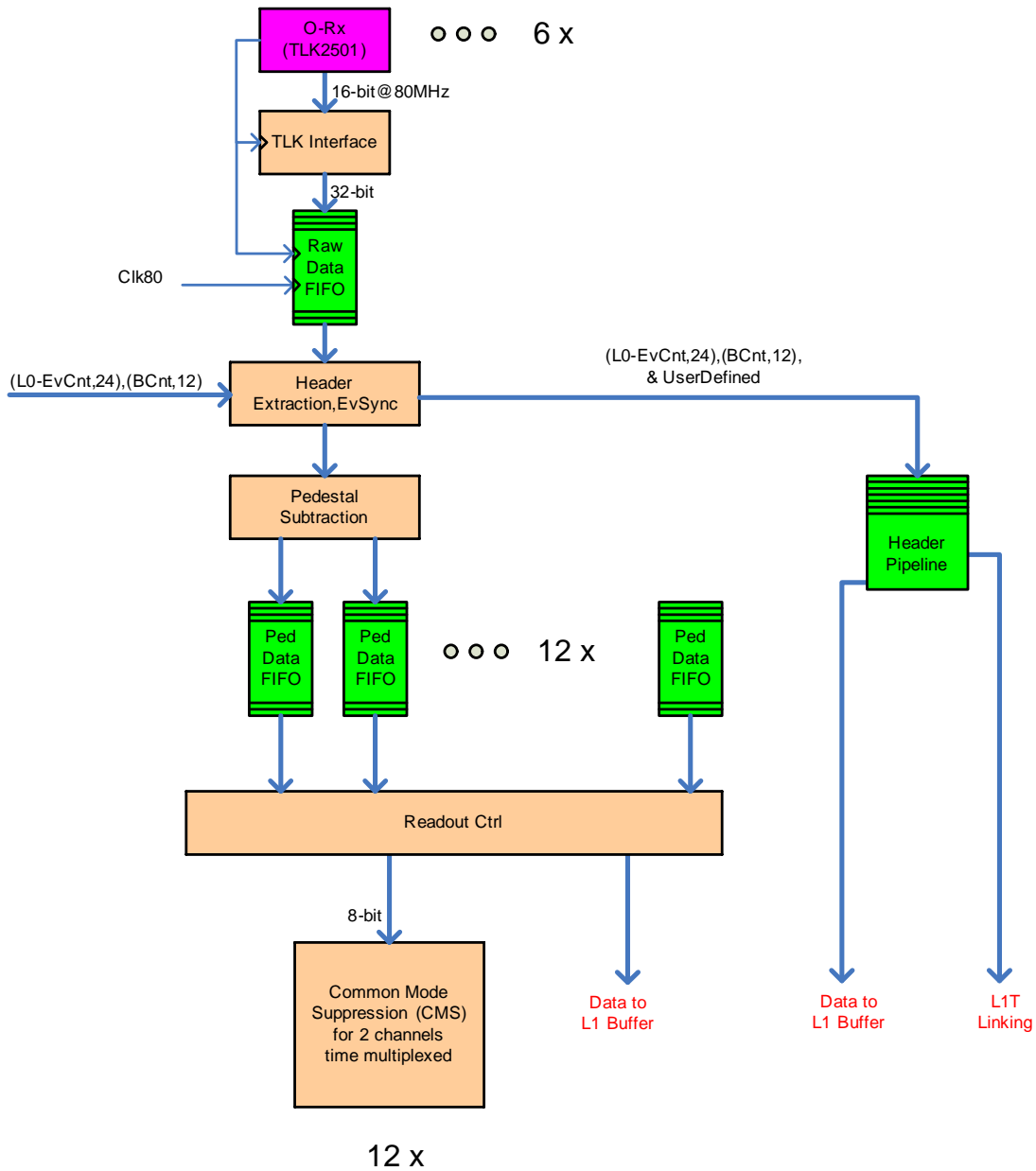


Figure 4.1: *Input processing overview in optical readout mode.*

event synchronization.

4.2 L1 Trigger Common Mode Suppression for IT and TT

The data read by the IT and the TT in principle suffers from the same potential problems as the VeLo signals. The HF pick-up is expected to be smaller than for the VeLo due to the increased distance from the beam and should lead to less common mode noise. On

the other hand the sensors strips are longer and the capacitances larger. As a baseline, the same CMS algorithm used for the VeLo should be used for IT and TT. The increased number of processing channel for the optical readout may force us to use a simplified CMS algorithm.

4.3 L1T Data Linking for OT

In a possible future upgrade of the current L1 trigger it is foreseen to use the OT information for the L1T decision. This requires to implement the L1T data path on the TELL1. The OT uses the OTIS [54] Time to Digital Converter (TDC) front-end chip for their read out. Digitization is applied on the OTIS. The most critical aspect for the OT L1T readout is the demand of high bandwidth even after zero suppression due to a very high occupancy in the detector [55]. An upper limit for the bandwidth needed is given by the case when non zero suppressed data is transmitted to the L1T which is of course an undesirable situation. To adapt the readout path to the OT detector geometry only 18 links are connected to one TELL1 instead of 24. Then the total number of channels is $18 \times 128 = 2304$ producing a non zero suppressed data rate of 2.6 Gbit/s (plus a small contribution from the header) which is about the maximal bandwidth for 3 Gigabit Ethernet links.

4.4 HLT readout

In order to adapt the TELL1 processing to the special needs for sub-detectors, a common data format for the HLT data transfer has been defined. The first common format is realized at the input to the L1B processing (see figure 3.22 the first FIFO stage “L1B RAW FIFO”). All the following stages are designed to use identical processing blocks. The CMS and zero suppression performed on the SyncLink-FPGA is specific to each sub-detector and can be fit in the design using the FIFO interfaces already introduced for the VeLo.

Chapter 5

The development of TELL1

The development of the readout electronics has gone through a long history and it is still ongoing... From the first board to the final production more than 8 years of studies and prototyping were needed. This period has been intensively used to gain the knowledge to handle large scale FPGA hardware design. This chapter is dedicated to summarize the development and show some particular design techniques used to build the complex data acquisition module that is made to process 30 Gbit/s of data in real time.

5.1 The ancestors of TELL1

5.1.1 RB1

The development for the first prototype readout board RB1 started in early 1998 by Y.Ermoline [22]. The prototype was wire-wrapped ¹ and used already a first TTCrx based mezzanine card for timing and fast control. The VME 6U module used a VME interface for the acquisition of two 8-bit ADC channels sampling at 5 MHz. The FPGA used on the board was an ALTERA Flex 10K.

5.1.2 RB2

The first PCB based prototype also equipped with a VME interface provided the readout capability of 4 8-bit ADC channels running at 40 MHz. The card was used in many test beam setups and can still be found in some labs serving as simple platform for detector acquisition for LHCb. The board provides the possibility to store 2 kWord of data to store 56 events, allowing to use a trigger based system. As already RB1 the FPGA used was an ALTERA FLEX 10K and the board was also designed by Yuri Ermoline [23].

L1PPI for RB2

The data processing for the L1T data containing CMS and sparsification was seen to be the most critical part of the L1 custom electronics development. A first prototype using a faster and larger FPGA ALTERA APEX 20K was built to study the implementation

¹This was a common prototyping technic at the time when signal speed was low.

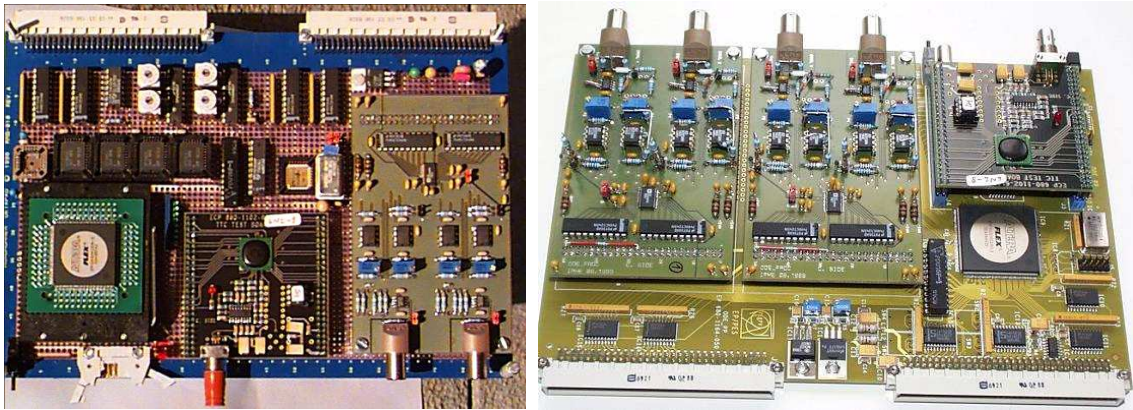


Figure 5.1: *The two first prototypes RB1 on the left and RB2 on the right.*

of the LCMS algorithm in an FPGA. The prototype was built as a mezzanine card for RB2. One of the ADC mezzanine card was replaced with the L1PPI card, the first card designed for LHCb by the author of this thesis. A lot of difficulties with the debugging of this card were related to the fact that not enough IO (connections between the RB2 and the daughter card) was available to control and readout the L1PPI card. A posteriori, we realized that a complete re-design with VME interface would have been more appropriate.

MXI for RB2

A second card using the RB2 VME interface and the two ADC card connectors, was built to connect the Texas Instruments DSP TMS320C6201 evaluation card. The card contains an FPGA driving data into the first prototype of RAM based L1 buffer, working at a L0 accept rate of 1 MHz. Bus switches were foreseen to arbitrate the L1B access between the FPGA and the DSP connected via Memory eXtension Interface (MXI). The implementation allowed us to benchmark the required transmissions between FPGA and DSP and the on chip DSP processing. This was the base for the design of a full custom DSP mezzanine card (see later).

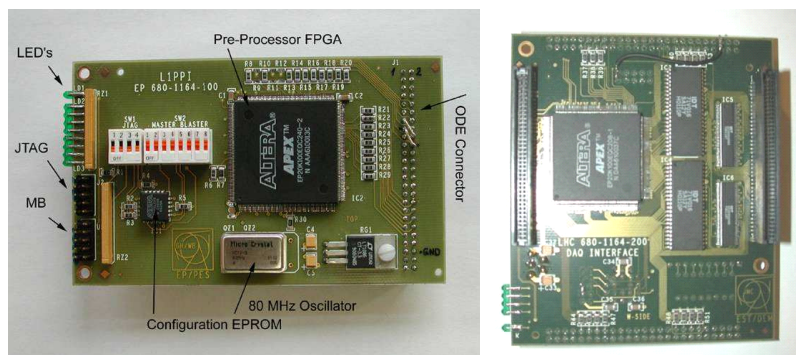


Figure 5.2: *L1PPI on the left and MXI on the right.*

5.1.3 RB3

The RB3 was the first step towards a readout board similar to the final board used in the experiment [24]. The card was designed by Yuri Ermoline and by the author. It allows to read 16 8-bit ADC channels and process the L1 data on the large ALTERA APEX 20K FPGAs implemented on the 9U sized motherboard. To have the flexibility for prototyping different implementations for the L1B and the HLT data processing, these functions were implemented on mezzanine cards. The interface to the L1T and the DAQ is provided with two separate S-Link connectors. For the ECS interface a custom micro-controller card was built that provides all necessary interfaces. RB3 is still in use and employed for readout tests in the Lausanne LPHE lab.

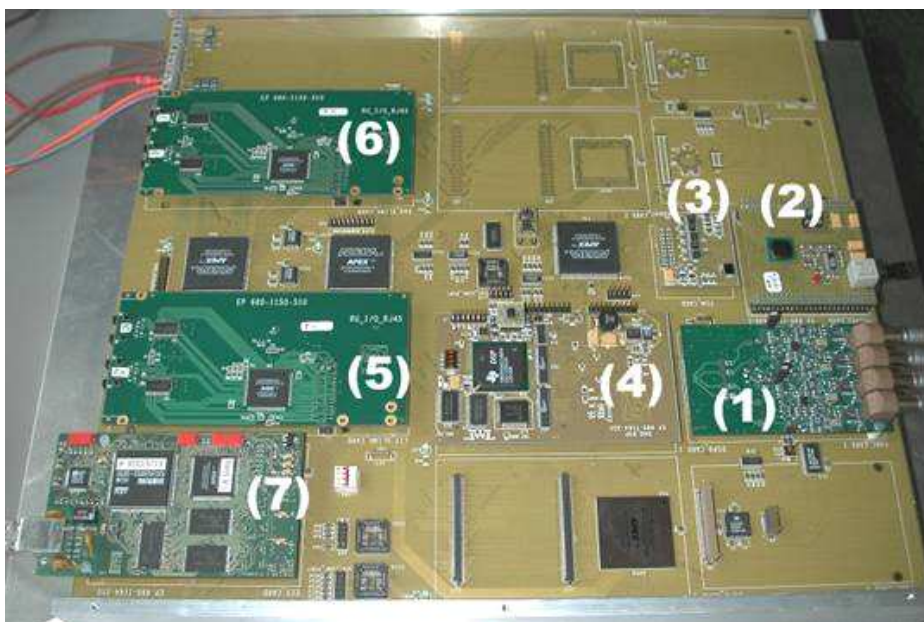


Figure 5.3: *RB3 partially equipped with ADC card (1), TTC receiver mezzanine (2), Front End Emulator (reference Beetle) (3), DAQ DSP card (4), L1T and DAQ S-Link source cards (5,6) and the ECS interface card (7).*

DAQ DSP for RB3

The HLT (former called DAQ) interface based on a Texas Instruments TMS320C6211 fixed point DSP running at an internal frequency of 167 MHz was designed to provide a L1 buffer and HLT interface for RB3. The complicated memory arbitration, insufficient processing power (16 DSP processors of this type would have been necessary on TELL1) and considerations for long time support have lead to the conclusion not to pursuit the DSP solution for the final implementation [40].

DAQ FPGA for RB3

To gain experience with the faster and more complex FPGAs and the QDR SRAM memory (see section A), we designed a mezzanine card with L1B and HLT processing function-

ality. The XILINX VIRTEX II FPGA used on the card was the first chip with embedded multipliers as they are now available on the ALTERA STRATIX used on TELL1. This prototype was also the first experience with the leading FPGA vendor XILINX and was intended to guide the final choice of FPGA. The layout of the the PCB was made at the workshop of EPFL.

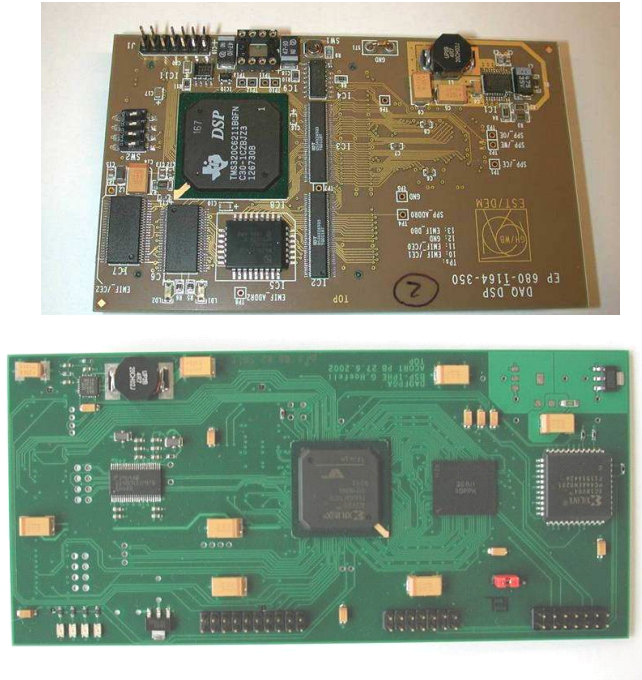


Figure 5.4: *The DSP (top) and FPGA (bottom) L1 buffer and DAQ interface cards are shown below.*

5.2 TELL1

The design of the final readout board was started at begin 2003. The previous prototype RB3 and its mezzanine cards was a good starting point to work on an improved architecture and the final choice of interfaces. At the time the interface to the event building network was not defined. Some lessons we learned from RB3:

- The separation of L1T data and HLT data linking into two separate FPGAs is inefficient from the IO usage point of view. With fast signal rates on the links between L1T-CMS processor and the FPGA responsible to link the data from the whole board, the separation can be avoided.
- The interface to synchronize with the TTC system can be made with the same chip which formed the name “SyncLink-FPGA”. This FPGA is the master of all functionality on the board and distributes clocks, triggers and resets to all devices on the board. On RB3 the functionality was distributed on 3 FPGAs.
- No DSPs will be placed on the board for the HLT data processing.

- The clock generation for the 16 ADC channels per PP-FPGA should be done directly on the FPGA to ease the synchronization and avoid the problems with the clock delay chips used on RB3.
- The S-Link specification allows transmission flow control only in a very restricted way. The possibility to have a fully bidirectional link is advantageous.

5.2.1 Analog receiver card (A-RxCard)

The analog receiver card is also developed in Lausanne. It will digitize 16 analog channels per card. 4 cards will be placed on a TELL1 for a total of 64 input channels. The digitization is synchronized via TTC system and a clock delay adjustable for each channel is foreseen 3.2.1. The prototype of the final card is in progress and will be tested as soon as possible. The analog signals are amplified and sampled by a 10-bit ADC fully differential in order to get the maximal possible noise rejection. Voltage offsets can be set via I2C and a digital to analog converter (DAC) is used to define the sampling window of the ADC for each channel.

5.2.2 Optical receiver card (O-RxCard)

The optical receiver card carrying one 12-way optical receiver module and 12 individual de-serializers allows for reception of one optical fiber ribbon with 12 x 1.28 Gbit/s data bandwidth. Two O-RxCards can be inserted on TELL1. The card has no intermediate FPGA for data reception but drives the data directly on the PP-FGPAs. This card has been developed by our Heidelberg colleagues.

5.2.3 Event builder Network interface (GBE RO-TxCard)

With the advance of a combined event builder for L1T and HLT a separate but unique interface based on Gigabit Ethernet was made possible. The board is developed by CERN LHCb colleagues. Two designs were considered, differing by the local host and data interface (PCI or POS PHY Level 3).

The direct interface of TELL1 to commercial network equipment was only made possible after reducing the event rate using multi event packets (see section 3.5) for L1T but also HLT. The subsequent “intelligent” multiplexing stage (Network Processors) was not needed anymore but required to be partially replaced by the source (TELL1). The common way to implement this tasks on network equipment is to use a so called “Framer” followed by a “Medium Access Controller (MAC)” and the “PHYSical layer device (PHY)”. The implementation for TELL1 was driven by the devices available on the market at the time. The most current used MAC devices on standard Network Interface Connection (NIC) cards interface via fast PCI bus (PCI 64-bit/66 MHz) due to the availability of PCI on standard PCs. The disadvantage with the PCI MAC device is the need of a PCI based fast host interface not available from the ECS. An additional PCI bridge from the ECS PCI (32-bit/33 MHz) would be required. To adopt a “Bus” interface (PCI) for a “Point-to-Point” problem seems to be somehow not ideal.



Figure 5.5: Quad copper Gigabit Ethernet mezzanine card (photo taken from [47]).

Searching for a “better” solution brought up the current chosen implementation. The Intel IFX1104 [56] is a Quad Ethernet MAC chip, that provides the possibility to transfer data between the SyncLink-FPGA on a FIFO like interface (SPI-3 [50]). The register setting of the chip is implemented on a separate microprocessor interface available on the ECS “Glue Card” (see next section) which is also used for the FPGAs on the motherboard. The two 32-bit SPI-3 interfaces can run at a data transfer rate of up to 133 MHz and therefore provide more than 4 Gbit/s for transmission and reception. Having a receiving path is foreseen for testing purpose at the moment but could also be used for network traffic scheduling. The simple interface of the MAC to the framer (SyncLink-FPGA) allows to use the MAC on the mezzanine card without any intermediate FPGA stage as it was considered. The implementation of the card was done by the CERN EP group and is documented in [47].

5.2.4 ECS interface (CCPC and Glue Card)

The ECS interface is used to download processing parameters to all programmable chips on the motherboard. No fast acquisition is needed for that task. The required interfaces are the microprocessor interface, I2C, JTAG and some GPIO pins given by the requirements of the chosen chips on the motherboard.

The most obvious interface in a crate mounted board is VME, as it was used successfully in other experiments.

The fact that the data acquisition is done on a “private” LAN has led to the conclusion that it would not be cost efficient to have a VME crate controller and a VME backplane for the ECS use only. The chosen implementation is based on a credit card sized PC (CCPC) running Linux connected to a 10/100 Ethernet LAN. The PC is interfacing to the so called “Glue Card” connected over PCI. On the Glue Card a PCI bridge is employed to convert PCI to a 32-bit parallel microprocessor bus. In addition 3 JTAG chains and 4 I2C buses are made available for the use on the motherboard. Two GPIOs are used for initializing the reconfiguration and reset the FPGAs.

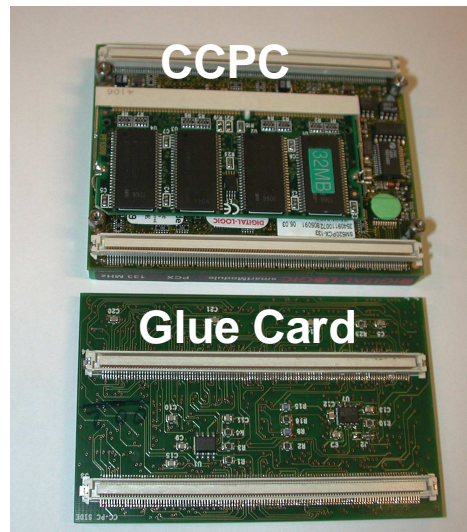


Figure 5.6: *CCPC and Glue Card.*

5.2.5 FEM

The Front end EMulator (FEM) card is used for synchronization of the VeLo readout. The card uses a Beetle chip to provide the data valid signal not transmitted along the data sampled on the ADC mezzanine cards. The chip bonded directly on the PCB will only be sealed with glue to avoid expensive packaging. It uses I2C to control the operation of the Beetle.

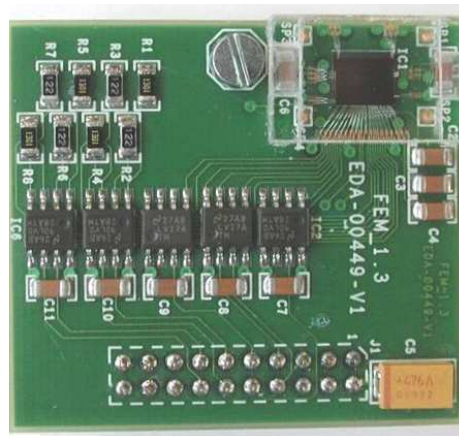


Figure 5.7: *FEM card with the reference Beetle. The chip is protected with a piece of plexiglas. The card size is 25mm x 35mm only.*

5.3 The Signal Integrity (SI) problems

With the transition to high speed signaling in the range of 100's of MHz it is mandatory to simulate the signal behavior of all critical parts on a board. The FPGAs used on TELL1 are optimized to solve SI problems providing termination and driving options for the IO cells. In this section I would like to show how the interconnect of critical signals on the board have been optimized by using simulation tools. For the design of critical interconnects three major design steps are taken. The pre-layout simulation is used to evaluate different options for a specific problem. Different termination schemes among optional driver and receiver settings can be tested assuming trace length and topology as expected by the pre-layout according to a floor plan of the board (see figure 5.8). The simulation models for driver and receivers are provided by the silicon vendors.

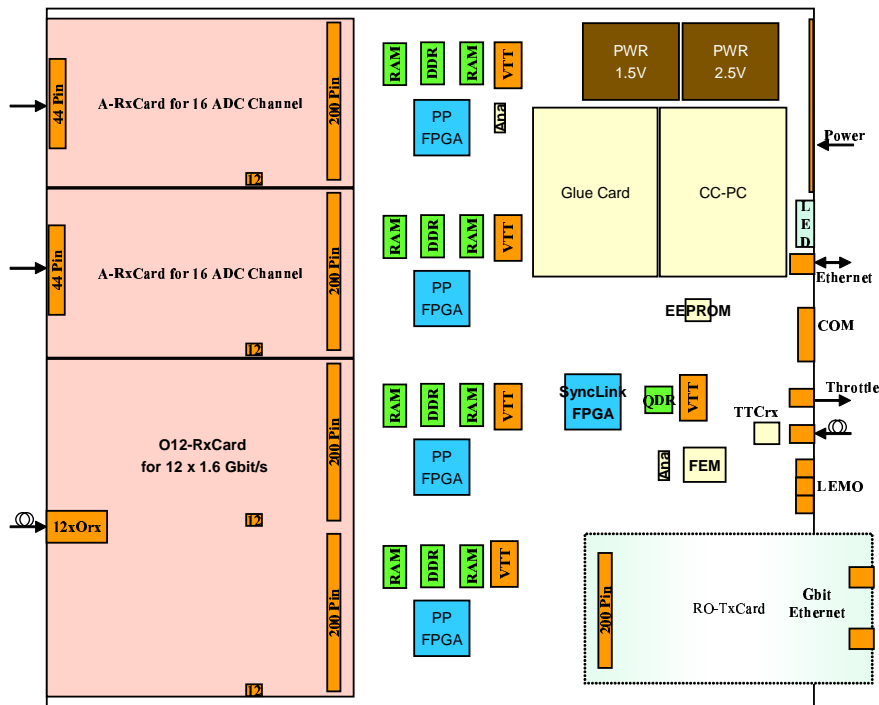


Figure 5.8: *The floor plan can give some estimate of the interconnection length on the board.*

Having found an appropriate implementation the schematics and the layout can be made regarding the placement of termination resistors, trace topology, and length. As the layout is made, the actual geometry of the PCB stackup and the trace routing including “vias”² can be extracted from the PCB data base to calculate accurate models of the interconnects. Gathering all this information allows an accurate post-layout simulation. The last step in the design is the verification of the simulated signals with the prototype. Nevertheless, measurements on high speed and high density interconnects are often

²A “via” is the interconnect between layers of the PCB. For high speed signals, the inductance of a via degrade the signal quality.

not possible since the signals are buried completely by the PCB and the chip packages. Probing points for signals can be implemented for verification. The signals recorded on the board for this work, were sampled with a high speed oscilloscope and active probes loading the test circuit with a minimal capacity (0.7 pF). The bandwidth is limited by the probes (1.5 GHz) and the Oscilloscope (2 GHz).

5.3.1 The PCB

At high frequencies, when the pulses rise or fall time becomes small compared to the propagation delay from the source to the sink, the signal will be affected by the transmission line characteristics. Reflections due to impedance mismatch between transmission line and receiver travel back to the driver and can be observed as spurious signals. The PCB, receiver and driver therefore need to be adapted to perform in the desired way. The signal edge of a 100 MHz digital transmission typically works with an edge of 1 ns. A line longer than 1/6 of the edge rate (≈ 3.5 cm for 1 ns) should be considered as a transmission line and requires an adequate termination [57, 58].

Two kind of transmission lines are very commonly employed on PCBs, microstrips and striplines. A microstrip is typically routed on an outside layer of the PCB and has only one reference plane. There are two types of microstrips, embedded and not embedded. An embedded microstrip is simply a transmission line that is buried into the dielectric but still has only one reference plane. A stripline is routed on an inside layer and has two reference planes (see figure 5.9).

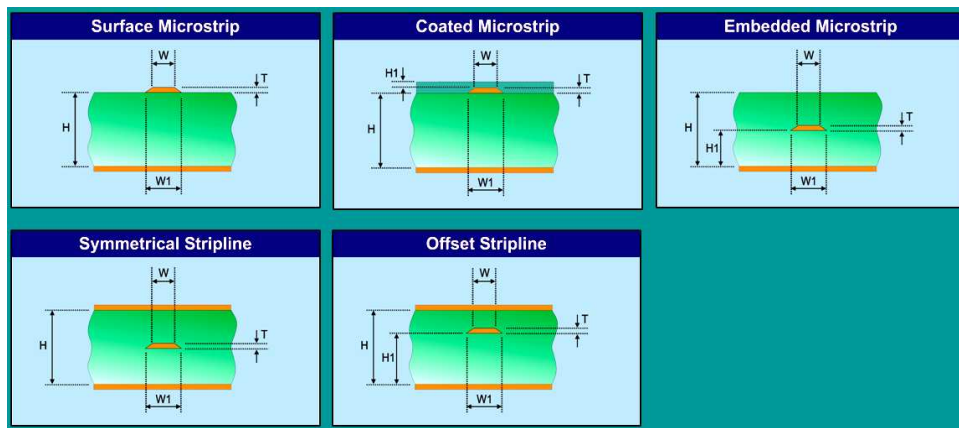


Figure 5.9: *Illustration of different transmission lines on a PCB. The coated case for microstrip is due to the solder stop layer usually implemented on the top and bottom layers. Illustration taken from [59].*

The impedance of the transmission lines are defined by the geometrical parameters indicated in figure 5.9 and the dielectric constant of the surrounding material. In case of a typical PCB, the dielectric material is FR4, which is a type of fiberglass. The calculation of the impedance can be done with impedance calculators made for PCB design, using precise geometrical models for the impedance calculation. The stack-up of the PCB can be defined by the electronics designer to obtain the desired behavior for each layer. The

TELL1 PCB stack-up has been made of 12 conducting copper layers and the adjacent FR4 material. The manufacturing of the PCB is split up in processes where FR4 planes with two copper layers (top and bottom) are produced before building the sandwich of the FR4 planes together. The FR4 planes are separated by the so called “Prepreg Layers” which are thin dielectric foils that act as a glue layer between the FR4 planes. The stack-up of TELL1 can be seen in figure 5.10.

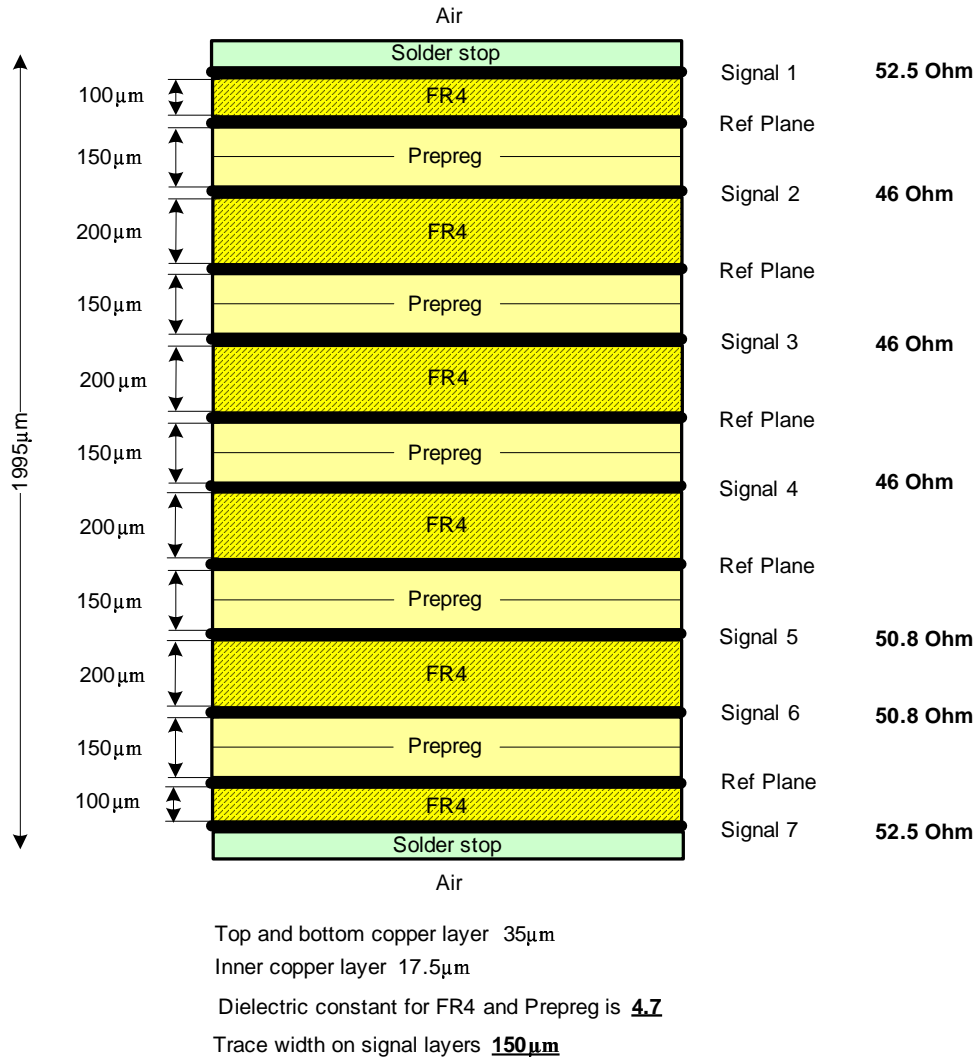


Figure 5.10: Stack-up of the TELL1 12 copper layer PCB. The thickness of the dielectric layers have been chosen such that the transmission line impedance becomes as close as possible 50 Ohms on every layer. Reference planes are inserted between each signal layer except for signal layer 5 and 6. Remark that a symmetrical stack-up is desired to avoid asymmetrical strength inside the PCB that could bend the PCB.

The stackup provides 7 signal layer and 5 reference planes. The reference planes on a PCB perform 3 functions. First they are used to provide stable reference voltages for driver and receiver, second they are used to distribute stable power to logic devices and last, to avoid crosstalk between signals. The signal return path taken by a high speed

signal lies directly under a signal conductor, following the path of least impedance dominated by the inductance for high speed signals. Providing a solid reference plane reduces therefore cross talk between nearby signal traces. Cuts in power planes known as “Split Power Planes” must be avoided or adequate precautions have to be taken. For example by avoiding signal traces running across the reference plane cuts or providing a low impedance path via distributed capacitors along the cuts. Plane cuts are introduced to change the DC voltage on a reference plane used for power supplies. On the TELL1 a multitude of power supplies are needed and the 5 reference planes are cut several times, one example is shown for the power supply for the PLLs on the FPGAs. The manufacturer recommends to supply a filtered power which requires to define small PLL power lands underneath the FPGAs (see figure 5.11).

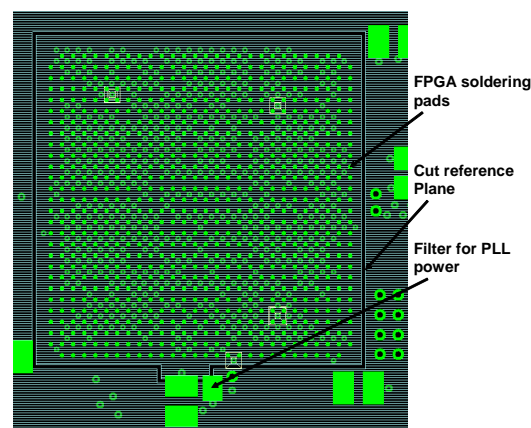


Figure 5.11: *The power plane is cut in the region of the FPGAs to provide filtered power to the PLLs.*

To economize the number of layers needed for the PCB, signal layer 5 and 6 are facing each other without an intermediate reference plane. This is tolerable if the layout follows an X Y routing scheme which means that traces on layer 5 are routed perpendicular to the ones on layer 6, this avoids crosstalk. In our case, the two layers are used for control signals only (ECS data bus for example) where the edges of the signals are slowed down by reducing the driving strength.

To control the accuracy of the PCB fabrication concerning the impedance, test circuits are implemented by the PCB manufacturer outside the customers circuit. For the TELL1, long test traces have been implemented directly on the PCB to measure the impedance on each layer independent of the manufactures test.

5.3.2 Termination

Having defined the impedances of the transmission lines on the PCB, a proper termination can be envisaged. The most popular termination scheme for signal transmission on PCB are the serial source and the parallel termination. The application of the two termination schemes with some variations are shown on specific examples used on the TELL1.

Serial clock termination

The preferred way to fan out (distribute the clocks from a single driver) is to use clock drivers that provide one driver per destination, so a point-to-point connection can be realized. The reason why special care is taken on clock signal distribution is to minimize jitter and glitches due to reflections on the clock lines. For DDR signalling, the edge rate for clocks is not higher anymore then for the data signal which has to be considered in the subsequent analysis. By inserting a serial resistor on the driver, undesired reflections for a point-to-point connection can be avoided. The value of the serial resistor on the driver is chosen such that the impedance of the driver plus the serial resistor is equal to the transmission line impedance of the trace. A perfect serial termination shows an incident wave amplitude of 50% of the steady signal. The voltage division between the driver plus serial resistor and the line impedance explains this circumstance. The schematics for the simulated topology is shown in figure 5.12.

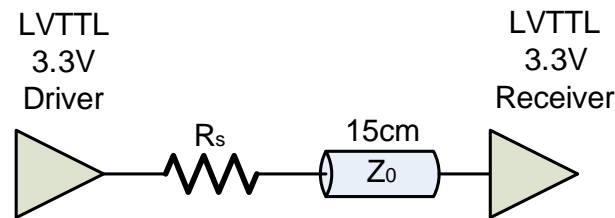


Figure 5.12: *The serial termination resistor R_s is chosen such that the reflected wave from the receiver completely is absorbed by the driver. Therefore driver impedance plus serial resistor equals the transmission line impedance.*

The simulations are performed with the Cadence SPECCTRAQuest Signal Explorer [60] using IBIS [61] driver and receiver models provided by the silicon vendors. A transmission line of 1 ns length (≈ 15 cm) is used. The two simulation results show the signal after the serial resistor and on the receiver. For illustration the simulation is done with and without serial resistor (see figure 5.13). Remark the large over and undershoot without serial resistor.

The length of the transmission line determines the time until the receiver receives the incoming signal transition. A big advantage of the serial termination is the low power consumption. Its disadvantage is the slowing effect by the additional serial resistor that acts like a driver with increased impedance. The rise time of the signal at the receiver is determined by the capacitive load and the driver strength. It is an RC circuit with R equal to the drivers impedance (driver plus serial resistor) and C the capacitive load. For $R = Z_0$ the 10-90% percent rise time equals:

$$T_{10-90} = 2.2Z_0C \quad (5.1)$$

Note: The typical capacitive load of a pin on a FPGA is 10 pF and the transmission line impedance 50 Ω . Therefore a rise time of 1.1 ns is a typical value. Note that a 10pF probe of an oscilloscope as commonly used for low speed signal probing is equals to the receiver lead.

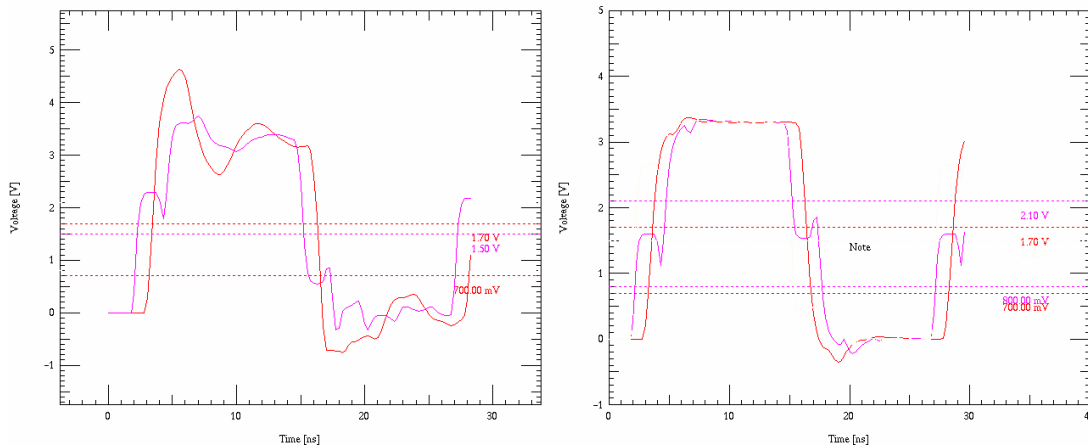


Figure 5.13: Results of the simulation without termination (left) and with serial termination (right). The signals shown are the voltage after the serial resistor and on the receiver.

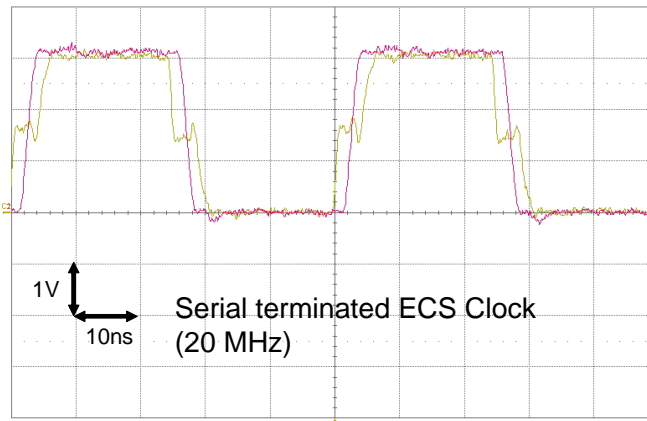


Figure 5.14: ECS clock distribution measured on TELL1.

5.3.3 Parallel termination for DDR SDRAM address bus

To use parallel termination for a transmission line, the driver has to be adapted to it ³. On the address bus of SDRAM modules several memory chips share a common address and data bus. In case of the TELL1 the address is shared among 3 memory chips (see figure 5.15) but the data is a point-to-point connection. The SSTL-2 [65] standard using parallel termination is employed to achieve a fast and low power “multidrop” interconnection ⁴. To reduce the power consumption the IO voltage is reduced to 2.5V and the termination resistor is connected to a termination voltage at half the IO voltage 1.25V. The current flowing in the driver has the same value for high and low logic state but opposite direction and requires the Voltage Termination Terminal (VTT) to be able to

³For example it is not possible for a 24mA, 3.3V LVTTTL output driver (typical for an FPGA) to supply sufficient current for a 50Ω terminated line (66 mA are required).

⁴“Multidrop” is a bus with several drivers and receivers.

source and sink fast currents. This is done with active devices specially designed for this purpose. The current defined by the SSTL standard is ± 7.6 mA for Class I and ± 15.2 mA for class II drivers.

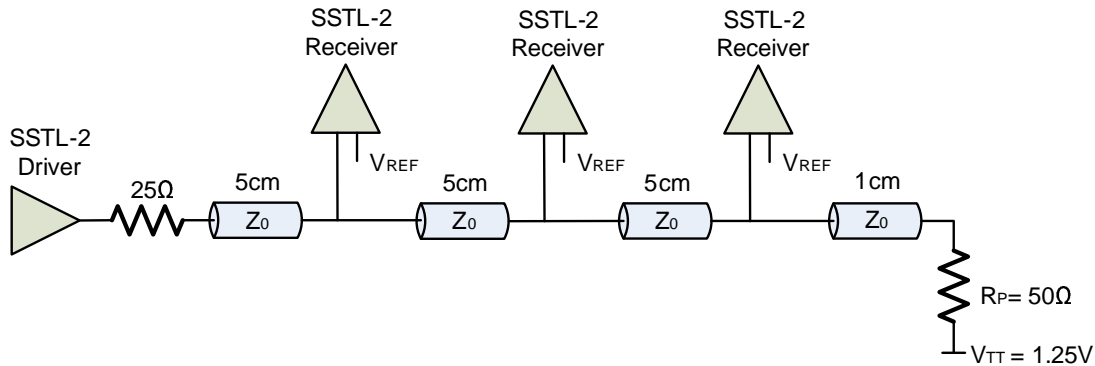


Figure 5.15: *SSTL-2 Class I transmission with one driver and 3 receivers. The 25 Ohm resistor defined by the SSTL standard can be omitted with the drawback of slightly increased noise.*

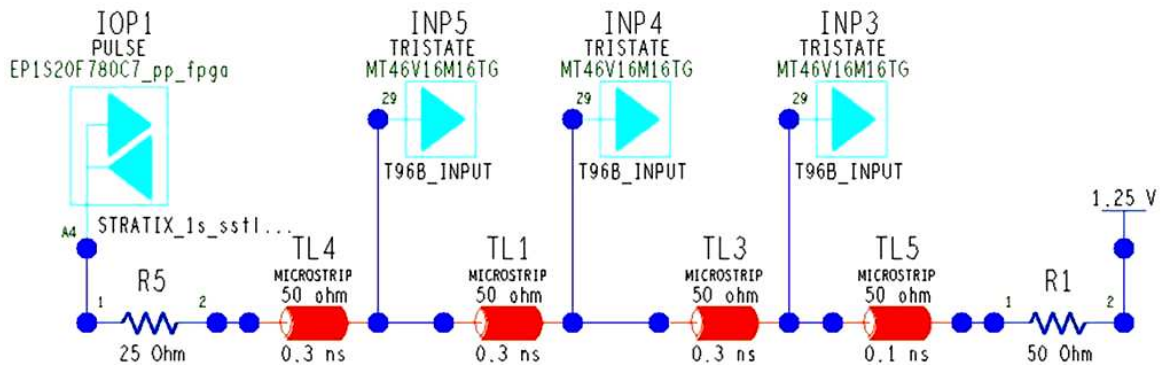


Figure 5.16: *SSTL-2 Class I transmission with one driver and 3 receivers. The topology as it is simulated with the IBIS models of the STRATIX SSTL-2 Class I driver and the Micron MT46V16M16TG receiver. The influence of the package of the chips are included in the IBIS description (pre layout simulation).*

The SSTL standard uses a reference voltage to define the switching point of the receiver which is normally set to half the IO voltage.

The reason why it is of interest to know if the serial resistor is required or not, is the following. The high density packaging of today's FPGA with pin counts of more than 1000 (SyncLink-FPGA counts 1020 pins) make the placement of resistors close to the driver impossible. This problem has been solved by providing on chip termination which in principle allow to insert serial and parallel resistors on the driver. Manufacturing problems of the silicon process has forced ALTERA to disable this option on the STRATIX device family used on the TELL1. For the TELL1 the "OCT Bug" has made some re-design necessary. The measured waveform of the data bus show that the signal quality on the data bus is very good and no hardware changes were necessary.

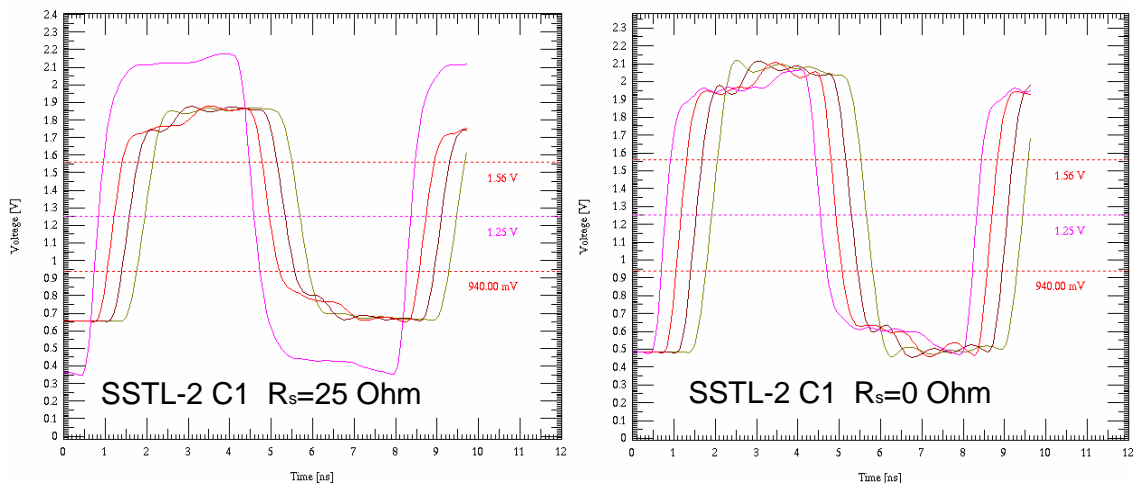


Figure 5.17: The address bus with (left) and without serial resistor (right) The noise margin for the second case is improved. This is the chosen implementation on TELL1.

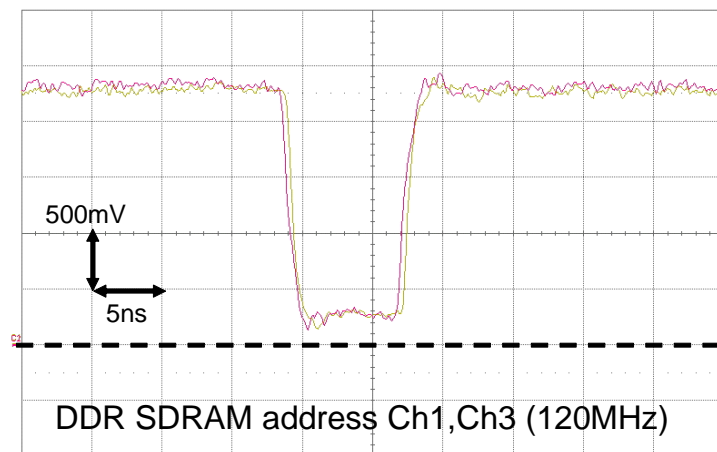


Figure 5.18: The address bus measured on chip 1 and 3. The memory access has been set up such that the address bit observed is required to go low for one clock cycle only. The driving strength of the driver of the first memory seems to be less strong than the one for the third. This can be caused by manufacturing differences of the silicon. The address on DDR SDRAM are operated in SDR mode. The cycle time for the address therefore is 8.333 ns at 120 MHz.

5.3.4 Point-to-point termination for DDR SDRAM data bus

The data bus of the L1 buffer implementation is a 48-bit wide bus running at 240 MHz data transfer rate. The power consumption of fast DDR memory has a major contribution from the memory interface and needs to be considered for the heat dissipation of the FPGAs. The “proper” termination for bidirectional multidrop buses requires parallel termination on both ends of the transmission line as shown in figure 5.19. Remark that two parallel and one serial resistors are required per data line.

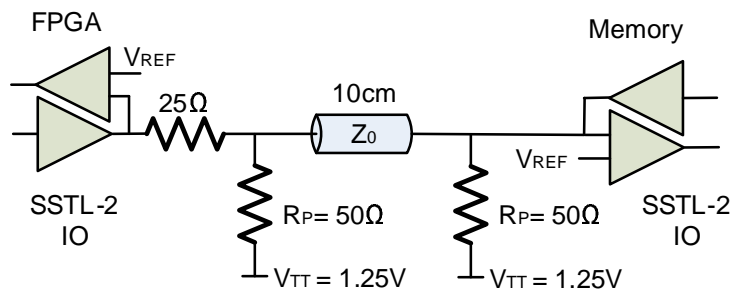


Figure 5.19: *Parallel termination on both side of the bus.*

The special situation where the memory data bus is a point-to-point connection, allows to apply a termination scheme described in [62]. A serial 68 Ohm resistor is inserted in the middle of the trace which is damping the reflections.

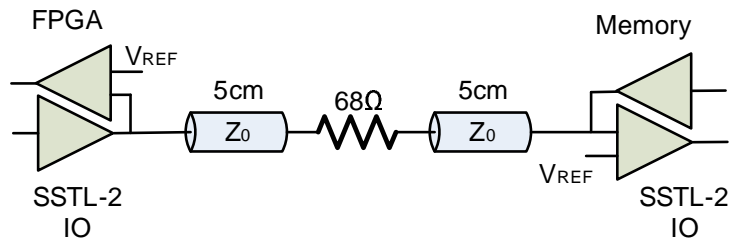


Figure 5.20: *Only one serial resistor is required with this termination scheme and the power consumption is very low.*

Once the layout is completed, the simulation of a specific connection can be done more accurately. The given example so far used only an approximation to the geometrical situation on on the PCB. Here is an example where the topology of a data trace has been extracted from the data base of the PCB. It includes now also the models for the vias, the models for the transmission lines calculated from the definition of the PCB stackup and the geometry of the traces and the driver models as already used in the pre layout studies (see figure 5.21).

5.3.5 QDR bus signals

The QDR memory used for the HLT MEP buffering is a fast dual port memory transferring two data words per clock cycle. Three interfaces of the chip have to be considered. The data read and write and the address port. The situation compared to the DDR SDRAM is more relaxed due to the fact that only unidirectional point-to-point interconnections are employed. The data write and the address port are identical signals for the termination point of view. The HSTL 1.5V Class I driver on the FPGA drives the memory receivers and a parallel termination resistance is required. The schematics is shown in figure 5.23.

Due to the very short distance of the memory from the FPGA, the fact of having point-to-point connection and perfect parallel termination a very good signal quality is not surprising (see figure 5.24).

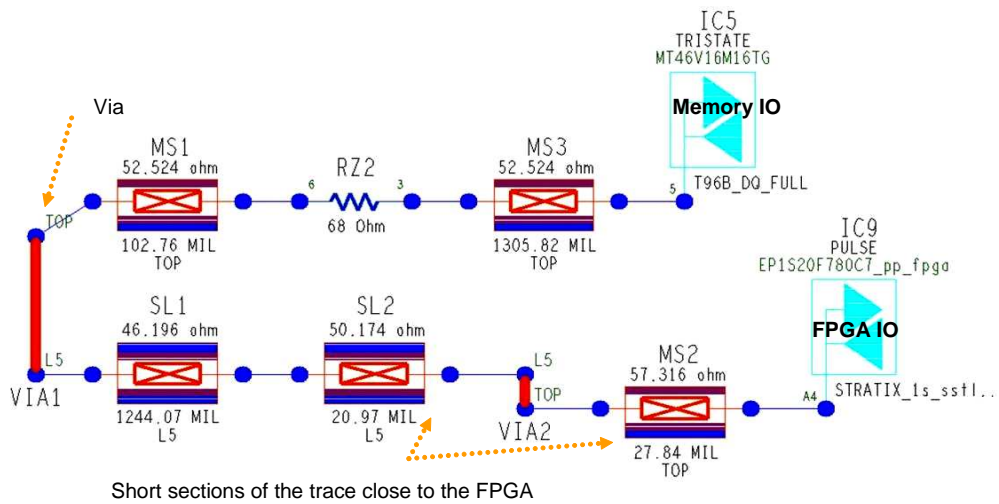


Figure 5.21: Post layout simulation of the data line of the DDR SDRAM.

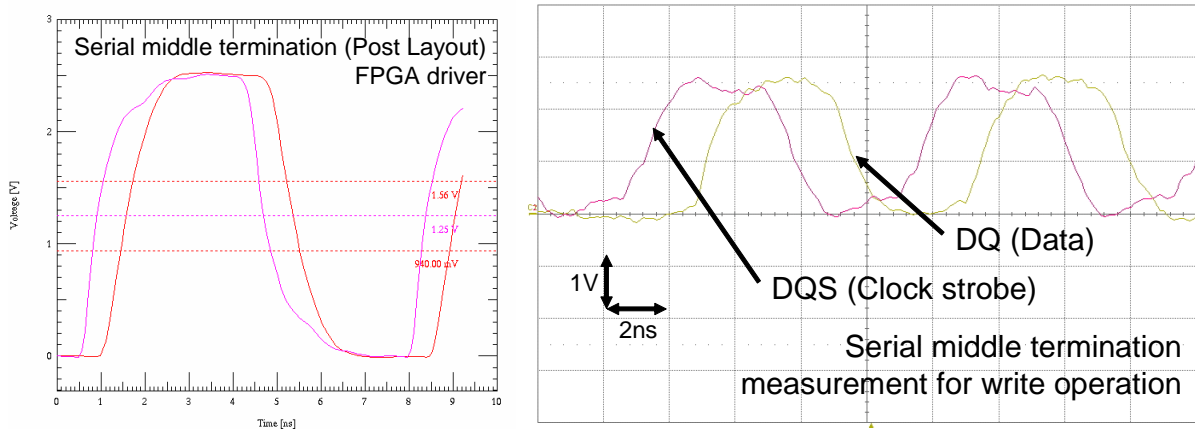


Figure 5.22: The simulation of the DDR SDRAM data bus signal in the post layout simulation on the left. The measurement corresponds very well with the simulated waveforms. The two measured signals are the DQS clock strobe and DQ data signal. Both signals used an identical termination scheme. The data is sampled on the rising and falling edge of the the DQS strobe signal!

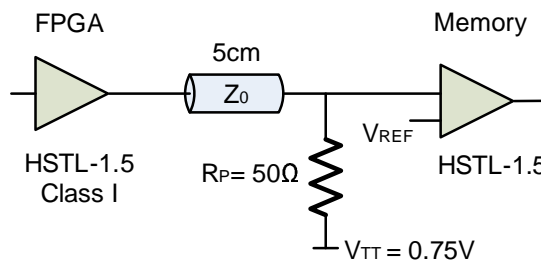


Figure 5.23: QDR RAM data write and address port implementation.

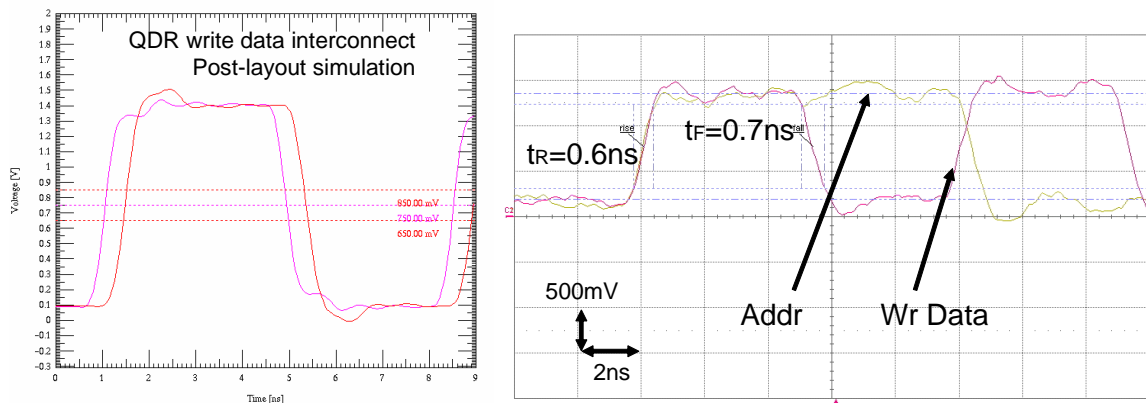


Figure 5.24: Simulation and measurements of the data write and address interconnects of the QDR RAM.

The read data path should be implemented with identical termination scheme but the termination resistors were planned to be implemented using on chip parallel termination resistors. The only solution found to this problem is not to terminate at all! The trace length in this particular case can be kept very short, the total length is less than 5 cm. The simulation and the measured result are shown in figure 5.25.

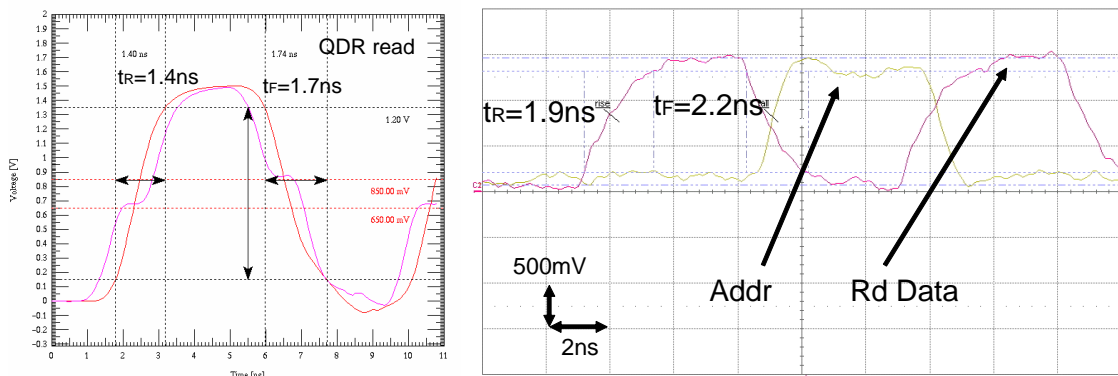


Figure 5.25: Simulation and measurements of the read data on the QDR RAM. The increased rise and fall time of the signal allows to run the interconnect without termination at 5cm connection length.

5.3.6 Inter FPGA connects for data linking

Much longer interconnects are required for the data links between the PP-FPGA and the SyncLink-FPGA. Two 16-bit wide double data rate buses for L1T and HLT data are required. The data transfer rate is 160 MHz. During the development phase these links were envisaged to be implemented with HSTL-1.5V Class I drivers with an on chip parallel termination resistor. This scheme is used for the QDR write data transmission illustrated in figure 5.23. Once more, the “OCT-bug” has forced the design to be changed. The

implementation now uses the on chip “Reduced Driving Strength” option which allows to achieve similar result as with serial source termination. With the driver strength of LVTTL-1.8V 8mA an acceptable signal quality is obtained 5.26. Almost half of the clock periode is used by the fall and rise time of the signal. The large difference between the link connections from the different PP-FPGA makes it necessary to sent a sampling clock on a separate line. In this way the flight time of the signal is compensated individually for each PP-FPGA.

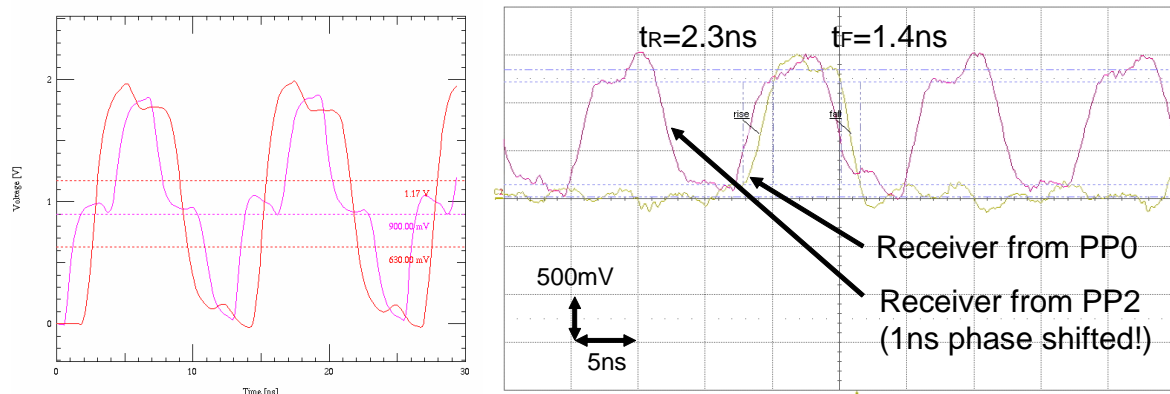


Figure 5.26: Simulation and measurements of the inter FPGA links. The connection measures about 20cm and is terminated using the reduced driving strength option. The data transfer is done with 160 MHz.

5.3.7 Driving strength control for SPI-3 interface to TxCard

Two 32-bit (Rx,Tx) point-to-point links running at 133 MHz single data rate are employed for the data transfer between the SyncLink-FPGA and the MAC PHY chip on the Gigabit Ethernet TxCard. Satisfying results for the up to 15 cm long links were obtained using the driving strength reduction option (8 mA) on the FPGA for the transmitter and serial resistors close to the MAC PHY for the receiver interconnects. The result is shown in figure 5.27.

5.3.8 ECS bus “Local bus”

The ECS 32-bit parallel bus is relatively slow (20 MHz) compared to the links we looked at so far. The SI challenges are given due to the long bus structure (trace length exceeds 50 cm) and the fact that 8 devices on the bus can act as driver and receiver. The correct termination scheme was found only thanks to pre-layout simulation. As a first point, it was seen that the clock signal can not be distributed in a bus scheme due to the poor signal quality obtained in the simulations. Our choice is to use a clock driver and implement point-to-point connections. The best result for the data bus was found by implementing the distribution by choosing a central bus crossing the board with short stubs connecting the devices to the bus (see figure 5.28). The bus is terminated on either end with a RC termination. Each stub is either working on reduced driving strength or with serial resistors.

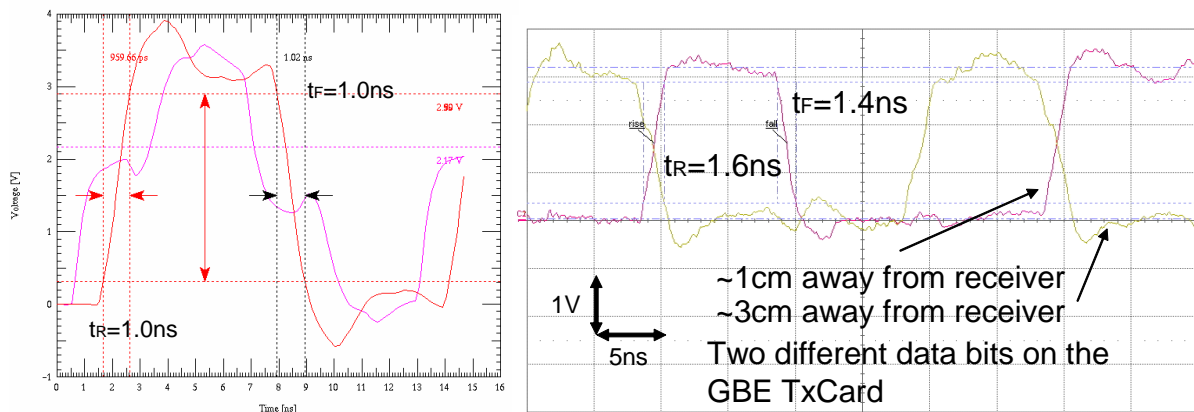


Figure 5.27: Simulation and measurement for SPI-3 interface data signals. The measurement has been done on a point 1cm and an other 3cm away from the receiver. Probing directly on the receiver is not possible because the signals run inside the PCB.

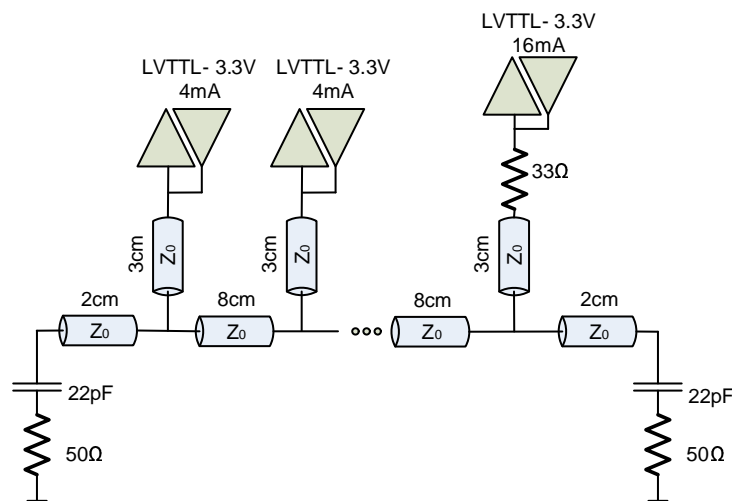


Figure 5.28: Schematics of the ECS bus implementation.

The signals do not look very nice on the simulation (figure 5.29) due to many reflections. To illustrate that the RC termination helps to reduce reflections and therefore the noise, the simulation is shown without RC termination as well (figure (b)). The reduction of the noise is important because of the cross-talk to neighboring systems.

The difference between the driver with serial resistor (33 Ohm) and the drivers with the driving strength reduced to 4 mA can be seen by comparing the signals in (a) and (c). The 16 mA driver has much faster edges. The measurements in figure 5.30 show the same signals as (a) and (c) with the clock superimposed. The signal level is sampled on the rising edge of the clock. The rising edge is driven by the PLX (16 mA with serial resistor) and the falling by the FPGA (4 mA drive).

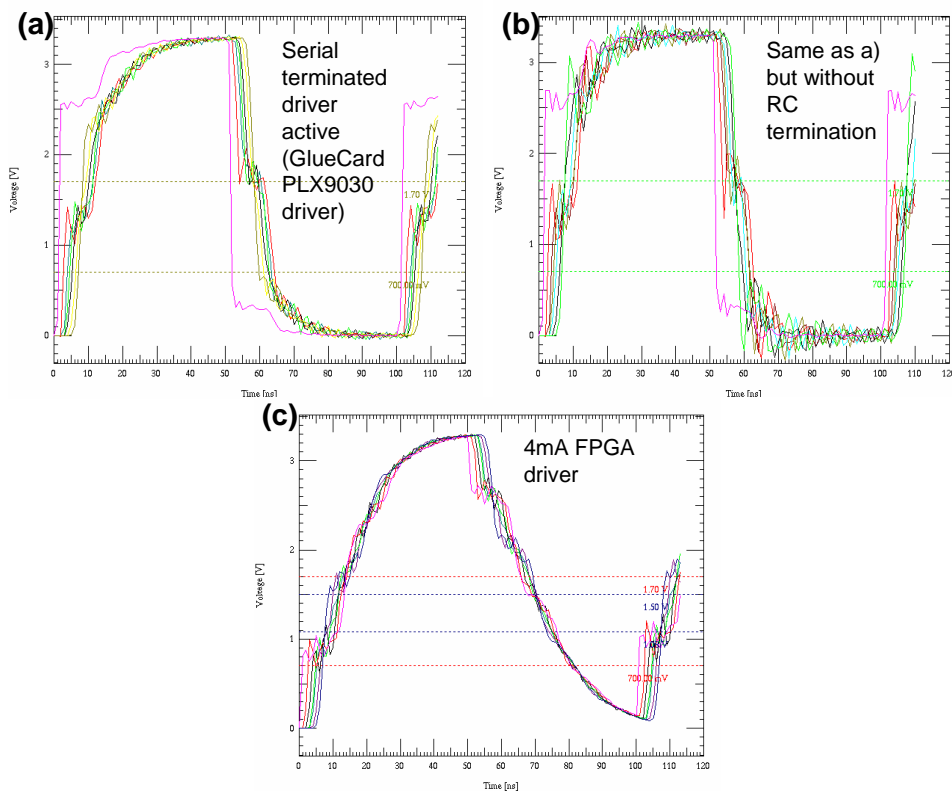


Figure 5.29: *Simulation for the ECS bus implementation.*

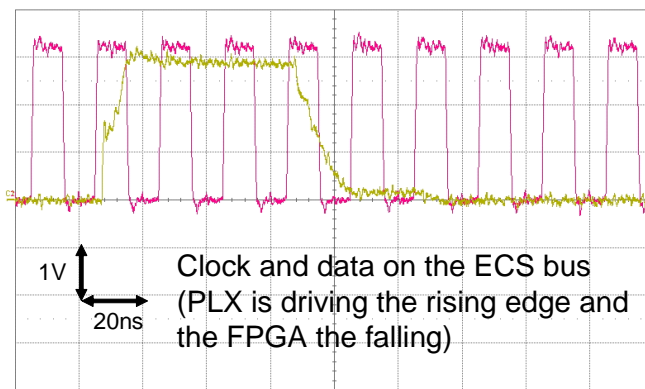


Figure 5.30: *Measurement for the ECS bus implementation.*

5.4 PCB Routing

Tight timing constraints for the high speed memory interconnects require to equalize the length of the signals on the data bus. On the layout this is implemented with meandered traces to fulfill the geometrical constraints. In case of the DDR SDRAM, the electrical trace length on the PCB have been matched to 50 ps accuracy. Remark that the signal velocity on the top and bottom layer of the PCB is higher then on the inner layers. The difference results from the fact that microstrips are half surrounded by air and half by

FR4 while striplines are fully buried in FR4. The calculated velocity on the microstrip layer 1 is $v_{\text{microstrip}} = 16.84$ cm/ns and on layer 3 $v_{\text{stripline}} = 13.83$ cm/ns. The electrical difference in length therefore is 13 ps/cm. The layout for the DDR SDRAM bank is shown in figure 5.31.

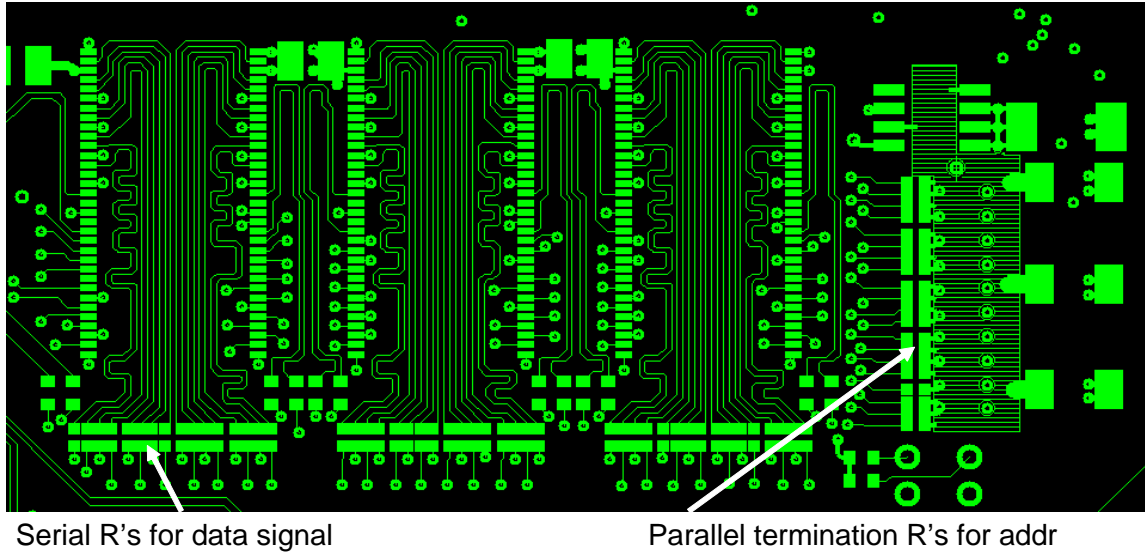


Figure 5.31: *DDR SDRAM to FPGA layout. The traces of the data bus have equal length.*

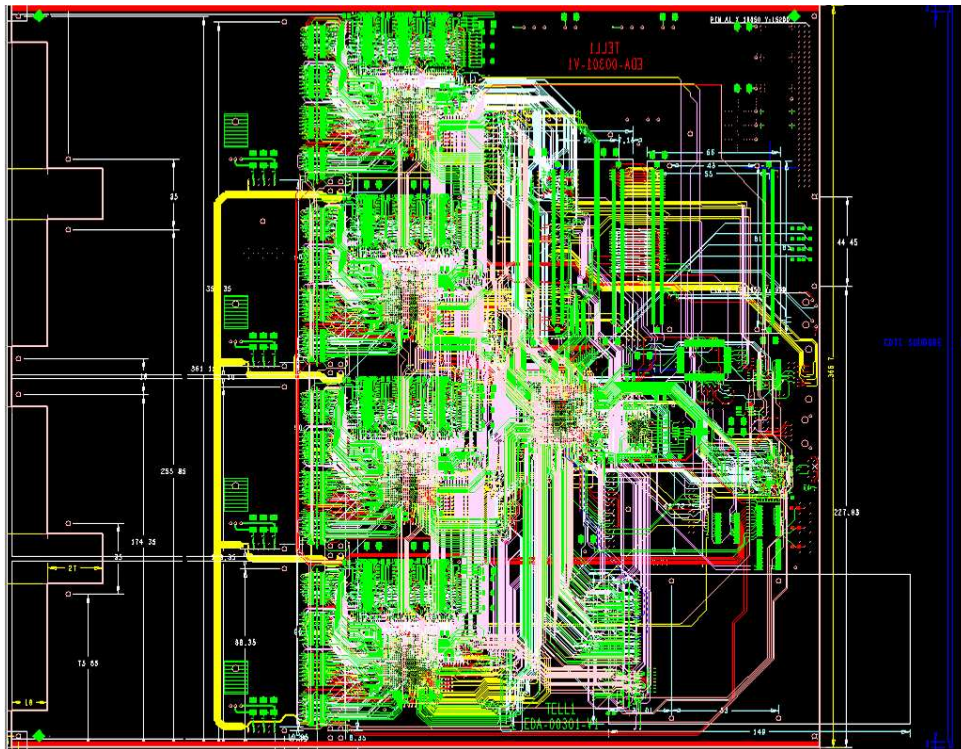


Figure 5.32: *TELL1* PCB routing. The layout of a complex board with routing constraints, large BGA packages and a lot of interconnects is an impressive amount of work.

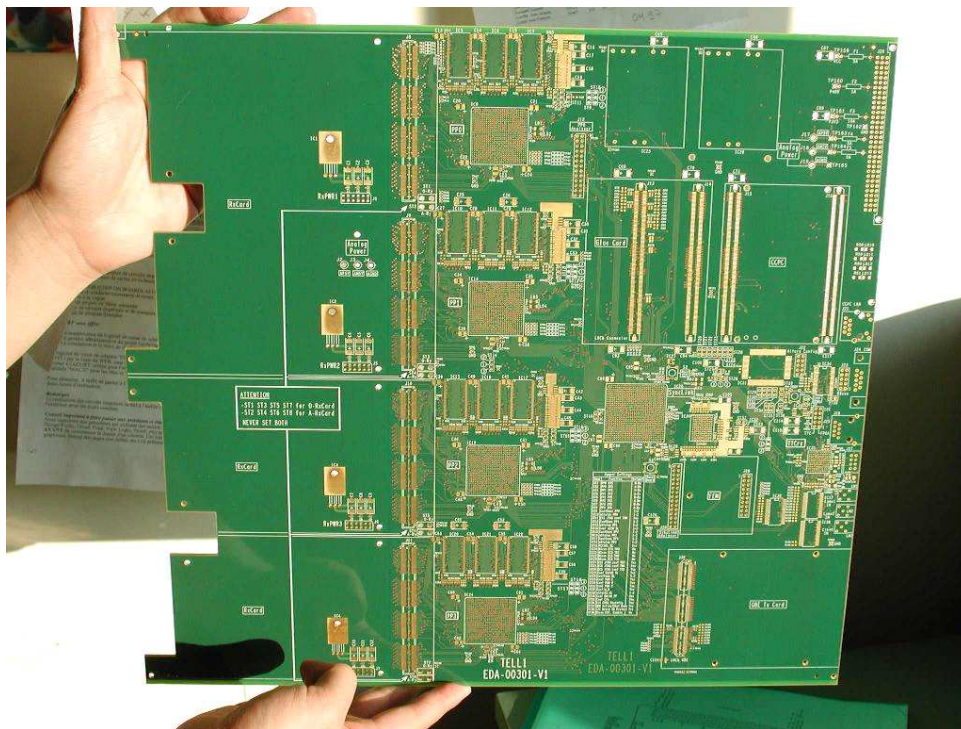


Figure 5.33: *TELL1* prototype PCB.

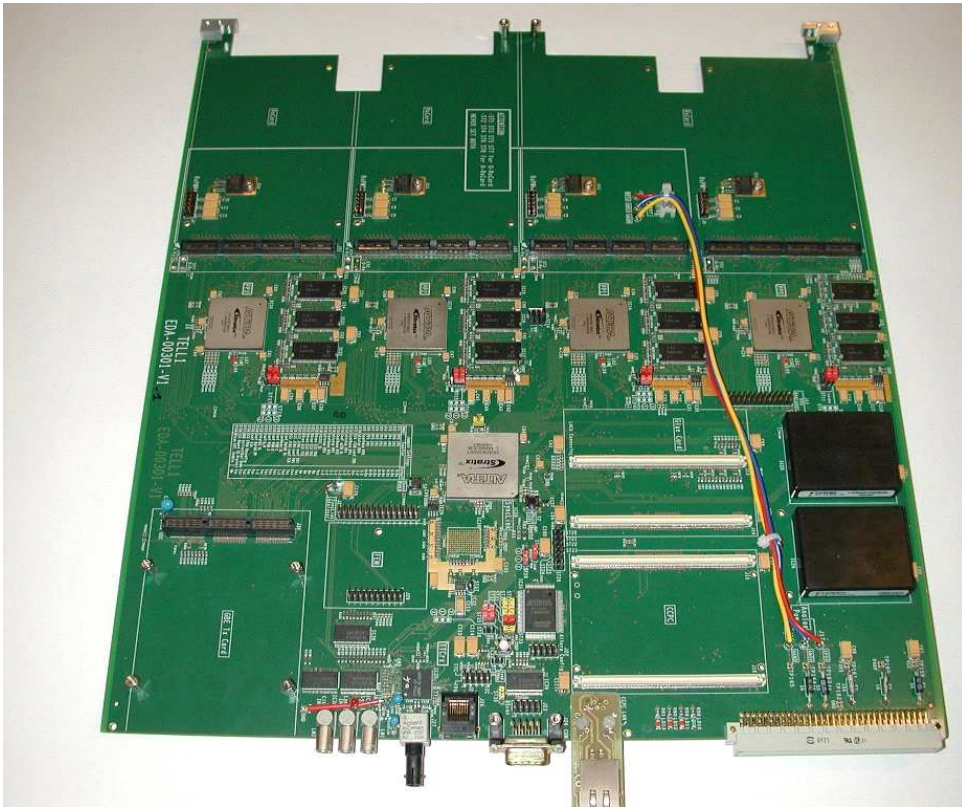


Figure 5.34: *TELL1* prototype without mezzanine cards.



Figure 5.35: *TELL1* prototype with mezzanine cards mounted.

5.5 L1 Buffer implementation studies

In the Technical Proposal (TP) [4] of 1998 the L1 trigger algorithm was using the VeLo detector information only. Based on benchmarks, a L1 trigger latency of $256 \mu\text{s}$ was estimated to be “easily” achievable. Since then, the demand for more trigger latency for different reasons has never stopped and driven it up by almost 3 order of magnitudes. The reasons for this additional demand are the growing complexity of the trigger algorithm using now the information not only of the VeLo but also of the TT station, L0DU and the Calorimeter Selection crate and the change to the commercial Gigabit Ethernet L1 trigger event building network. To cope with the increasing latency the search for solutions on the L1 buffer implementation went in parallel. To understand the development of different implementations an introduction to different aspects on the buffer parameters and the commercial available memories is given below.

5.5.1 Principle of operation of the L1 buffer

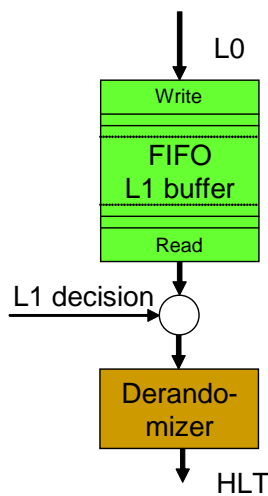


Figure 5.36: *L1 buffer principle with FIFO.*

The operation of the buffer as outlined in the TP is very simple. On the write port, the L0 accepted data is written into a First In First Out (FIFO) buffer. For the read port, on each L1 decision the data is read out from the FIFO and in case of a L1 accept, it is transferred to the output derandomizer buffer. As already explained, a derandomizer is needed to uniform the time spacing between events to match the input bandwidth of the subsequent stage. Rejected events are read from the buffer but not passed to the derandomizer (see figure 5.36). The FIFO based buffer management is in many aspects not ideal but the main reason that it is not applicable is the insufficient memory size. The buffer depth is limited to the available FIFO chips which are expensive and are typically 4 to 8 times smaller than normally available Static RAM (SRAM). In the following a wider range of aspects of the buffer implementation are discussed.

5.5.2 Data access characterization

The L1 buffer data transfer character is well defined. With the L0 accept rate of 1.11 MHz, 36 word per event and per optical link of the tracking system are written to the buffer. The access can therefore be done in a so called “burst access” to consecutive addresses in the buffer. The read access is much less frequent: the L1 accept rate allows to read at a maximum of 40 kHz. The fact that read accesses are “rare” suggests to use a shared read and write port. To arbitrate the access, an intelligent buffer controller is needed.

5.5.3 Buffer size

As already mentioned, the L1 buffer size has been increased by 3 orders of magnitude since the TP. Since 1998 the memory chips available on the market have grown in capacity by about a factor 10. To calculate the number of chips needed on an acquisition board we must determine the memory usage per readout link. All sub-detectors except the VeLo are read out over optical fibers transmitting 32-bit per clock cycle or 1.28 Mbit/s (for the VeLo 4 analog readout links correspond to a data data stream of 32-bit per clock cycle (1.28 Mbit/s). A L1 buffer depth of 58254 events with 36 words per event needs a memory size of 2 Mword or 64 Mbits. It should be pointed out that a read out board with 24 optical links is planned.

The present largest chips available are summarized in table 5.1. It is evident that the

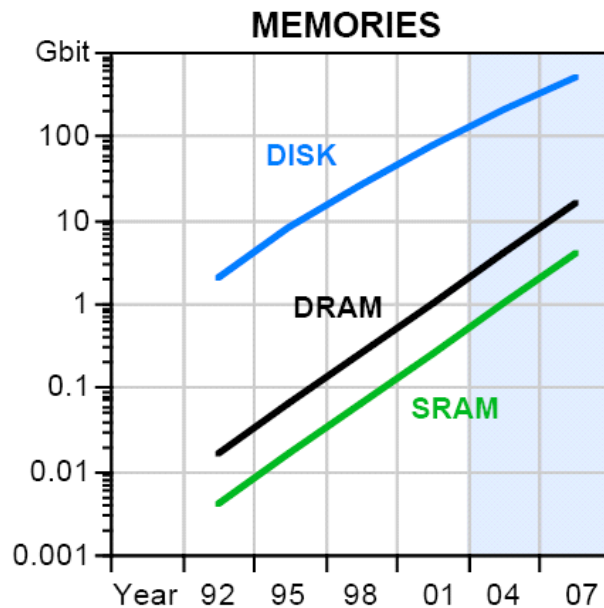


Figure 5.37: Evolution of the memory density over time. Picture taken from [25].

Memory type	Available 2004	Bus width	IO interface	Price / Mbit
FIFO	5 Mbit	20 bit	200 DDR	20 CHF
ZBT SRAM	18 Mbit	36 bit	133 MHz SDR	2 CHF
QDR SRAM	18 Mbit	18 bit	166 MHz DDR II	3 CHF
DDR SDRAM	512 Mbit	16 bit	166 DDR	0.04 CHF

Table 5.1: Based on some typical example the available density the table shows IO performance and price of a given memory technology. A short introduction to the memory types will be given below. For this table the bus width needed to transfer 32-bit data has been given. We should notice that for the Double Data Rate (DDR) interfaces only half the bus width is needed. The values for this table have been collected from the semiconductor distributors EBV [63] and Avnet [64].

change from static RAM to dynamic RAM decreases the cost of the buffer and leaves a very flexible partitioning of the board. The data from several optical links can be buffered in only one SDRAM memory chip.

5.5.4 Memory interface

The IO interface used for a memory defines the bandwidth available. The standard has moved from single data rate using Low Voltage TTL (LVTTL) at 3.3V to double data rate (DDR) using Stub Series Transceiver Logic (SSTL) [65] at 2.5V, typically used for DDR SDRAM, or High Speed Transceiver Logic (HSTL) [66] at 1.5V used in QDR [48] and DDRII. The change of IO standard has significantly accelerated the data transfer and at the same time reduced the power consumption. For the SDRAM typically used on memory modules where 4 or 8 memory chips share the same address bus, parallel signal termination is applied. The power consumption for wide buses is significantly and has been reduced by lowering the IO voltage.

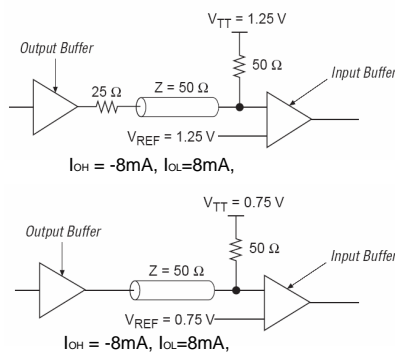


Figure 5.38: In the upper figure, the termination scheme for 2.5V SSTL Class I and in the lower figure for 1.5V HSTL Class I are shown.

of the L1 buffer. To store the data from 2 optical links, a single 16 bit wide DDR interface running at 120 MHz is sufficient for constant L0 event data write and L1 accepted data read.

Figure 5.38 shows the termination scheme for SSTL and HSTL signals. The current at the drivers can be as small as 8 mA because of the reduced IO voltage and the parallel termination to the V_{TT} level of 50% of the IO voltage. For high speed interfaces, the data and address are transferred on the rising and falling edge of the clock which is named "DDR interface". All modern FPGAs are now supporting the SSTL and HSTL IO standard including the termination scheme, and provide the necessary DDR IO registers. The change from Single Data Rate (SDR) to DDR and the increased clock frequency provides sufficient bandwidth on a single port memory, to perform read and write operation

Conclusion

Since the LHCb Technical Proposal the system parameters for the readout electronics of LHCb were continuously adapted to better match the needs of physics. The work presented in this document is the description of an element of the final implementation of the Level 1 electronics: the TELL1 readout board. It is now the last “custom made” piece of hardware in the readout chain. The TELL 1 will interface commercial networking equipment for the L1 and High Level Trigger.

This thesis shows the result of the long but exciting process to work in a large collaboration among physicist and electronics engineers that fight for a better detector. In this process, the requirements to the readout electronics were pushed to the possible limits in technical means while keeping under control the financial aspect. We should mention the always returning discussion on the L1 buffer size that has driven us to use modern DDR SDRAM that can provide 1000 times more buffer space than foreseen in the TDR. TELL1 is the result of a well planed prototyping phase, during which the board has been adapted not only to the readout of the VeLo but for the other sub-detectors in LHCb. All sub-detectors (except the RICH) will use the TELL1 for the data acquisition and L1 trigger interface. It should be mentioned that the common readout board has quite complicated the design. On the other hand it had a positive impact on the decision process and it is clearly the preferred solution regarding the maintenance over the long time of data taking. The specification of the Gigabit Ethernet based network interface card and the event data formatting using Multi Event Packet and IP protocol, have been considerably simplified due to a common readout board.

A design for the common mode noise correction in particular and for the whole readout processing for Level 1 and High Level trigger in general has been developed and successfully tested on the hardware. The adaptation to the optical readout has been carefully considered at the firmware development for the FPGAs as well as for the hardware configuration using the receiver mezzanine cards.

The electronics development with data transfer rates of up to 240 MHz has made modern design technics necessary. All critical interconnects were carefully analyzed with simulations using models of the interconnects given by the geometrical aspect of the PCB and the silicon characterization of driver and receivers. The obtained result have been compared with the measurements on the present prototype.

All interfaces have been tested and minor changes were applied in order to produce 20 board in a pre-production in June 2004, before the final production in 2005-2006 for a total of 300 boards.

Appendix A

The Zoo of memories

We give here a short description of the different kind of memories available on the market.

FIFO

FIFO buffers are currently used in high end Digital Signal Processing (DSP) application where subsystems with different clock domains need to interface and bus width translation is done. The FIFO devices are very convenient to use. Two ports are available to perform independent simultaneous read and write operations where for the address generation is taken care automatically by the memory logic. The latest devices support also read pointer preset commands which allow to increase the read pointer without reading out data (skipping data from read out). Also the majority of the the FIFOs have conventional SDR LVTTL interfaces, modern devices come with DDR interface using HSTL as IO standard. Figure A.1 shows the block diagram of a modern FIFO device.

The highest density devices are 2 to 4 times smaller then standard SRAM at 3 to 4 times higher cost ¹. The fact that modern FPGAs have dual port FIFOs implemented as embedded memory blocks directly on the chip, eliminates the use of external FIFOs.

SRAM

SRAM are currently used as external memory for network processors in networking applications where low access latency is important. For small data packets and random address sequence the performance of SRAM in superior to that of SDRAM. The memory has only one data port and the address port. Read and write operations have to be scheduled by the controller. This drawback is solved with the QDR SRAM devices.

QDR SRAM

The Quad Data Rate (QDR) is an SRAM with two data ports for read and write.

The addressing is done time multiplexed on a single address port. Its called Quad Data Rate because read and write operations can be performed simultaneously and each is transferring the data on a DDR interface. A block diagram is given in the figure A.2. The common address path is chosen to keep the number of pins on the memory and on

¹FIFOs are produced in relatively small quantities compare to standard SRAM of SDRAM.

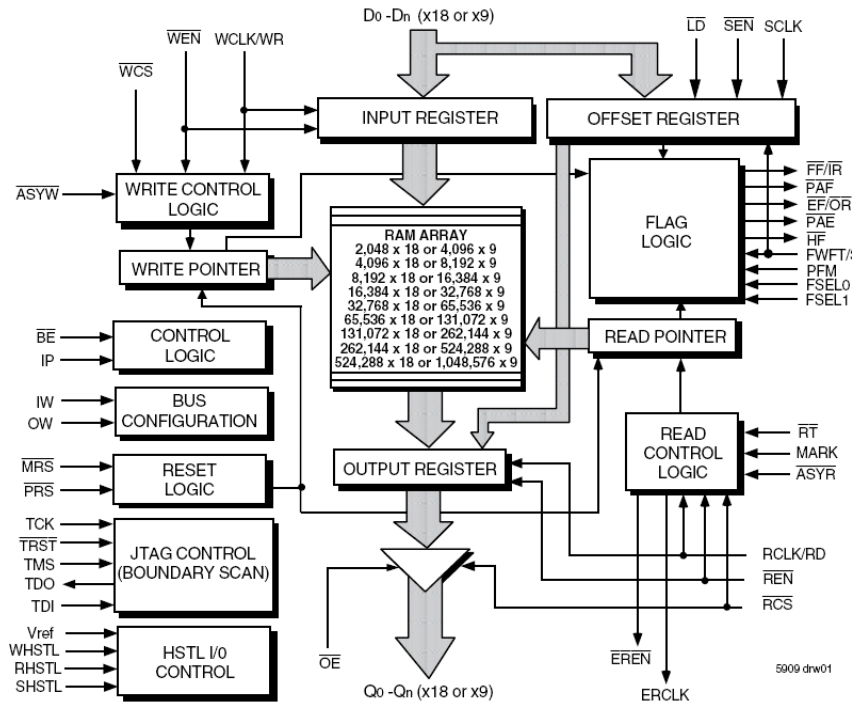


Figure A.1: Block diagram of a 20-bit wide, 5Mbit IDT [67] TeraSync FIFO.

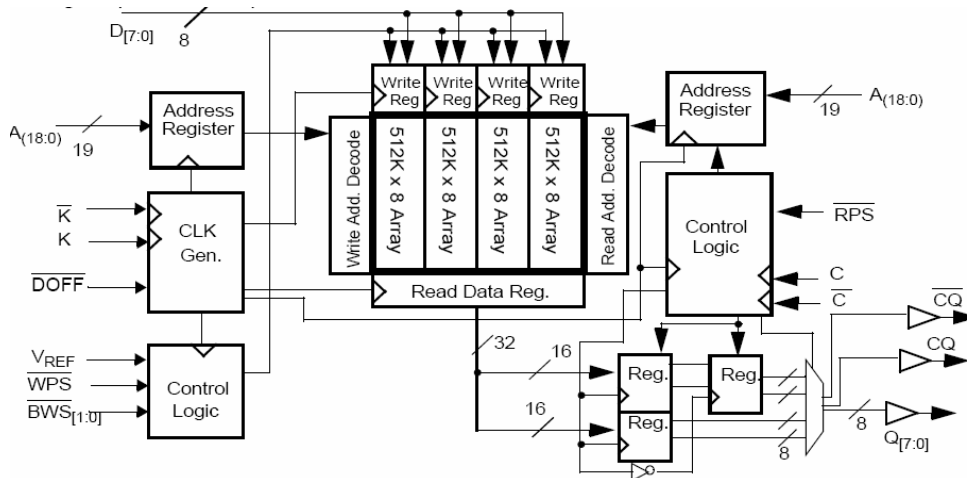


Figure A.2: Block diagram of a Cypress [68] 18-bit wide, 18 Mbit QDR II memory.

the controller low. The data transfer is scheduled in packets of 2 or 4 (burst mode) and the address has to be transmitted for the first word in the burst packet only.

SDRAM

DDR SDRAM is the most common used external memory for servers and personal computers. The highest density chips on the market provide 1 Gbit of memory. A cell consists of only one capacitor and one transistor to switch the charging current. Periodic

re-sampling and re-charging (refresh) of the capacitors charge prevent from losing the information due to leakage currents.

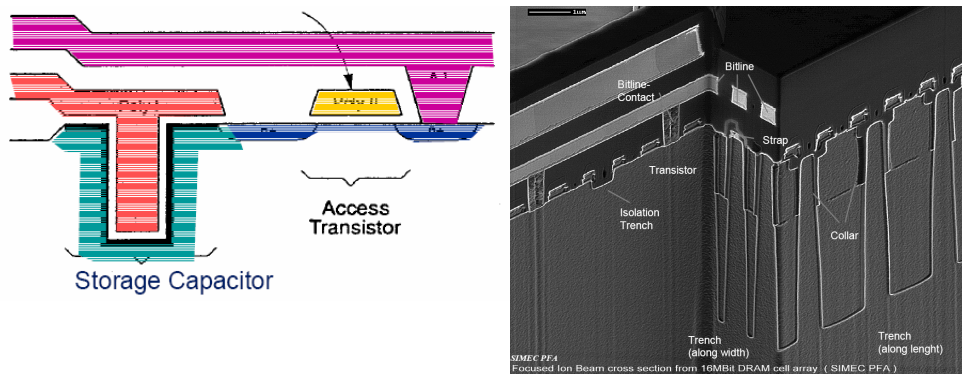


Figure A.4: To save chip surface, the so called "Trench Cells" are used to implement the capacitors. The capacitors are deep holes in the silicon substrate. Picture taken from Infineon [69]

The logic to perform refreshing is implemented directly on the chip. To access the memory contents a certain protocol has to be followed because the chip is organized in banks, columns and rows and uses time multiplexed addressing. To access a memory location on the chip, the row address needs to be transmitted (charged) previous to the column address. Consecutive column locations can be accessed afterward without recharging the row address. Especially for large data blocks which can be transmitted in a burst mode, the access performance for SDRAM is similar to the one of SRAM. Only in circumstances where a lot of read and write access to arbitrary memory locations are performed, the advantage of SRAM gets into play. To interface the memory to a FPGA, SDRAM controller are made available as intellectual property cores (IP-cores). A typically block diagram of an SDRAM memory looks like shown in figure A.5:

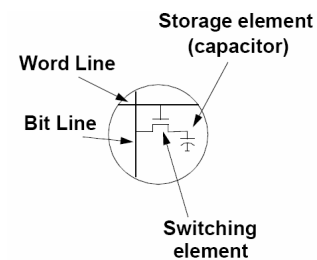


Figure A.3: DRAM cell illustrated.

For the DDR SDRAM, special care is taken to avoid problems with large buses due to signal skew. In general, for each group of 8-bit data lines a separate strobe signal is generated (DQS) which is used on the controller interface and on the memory chip to latch the data on the bidirectional data bus. It should be as well pointed out, that the IO usage for a DDR SDRAM based L1 buffer implementation can be significant lower then for the QDR SRAM based buffer. The read and write operations as well as the address are multiplexed on one data port and one half width address port.

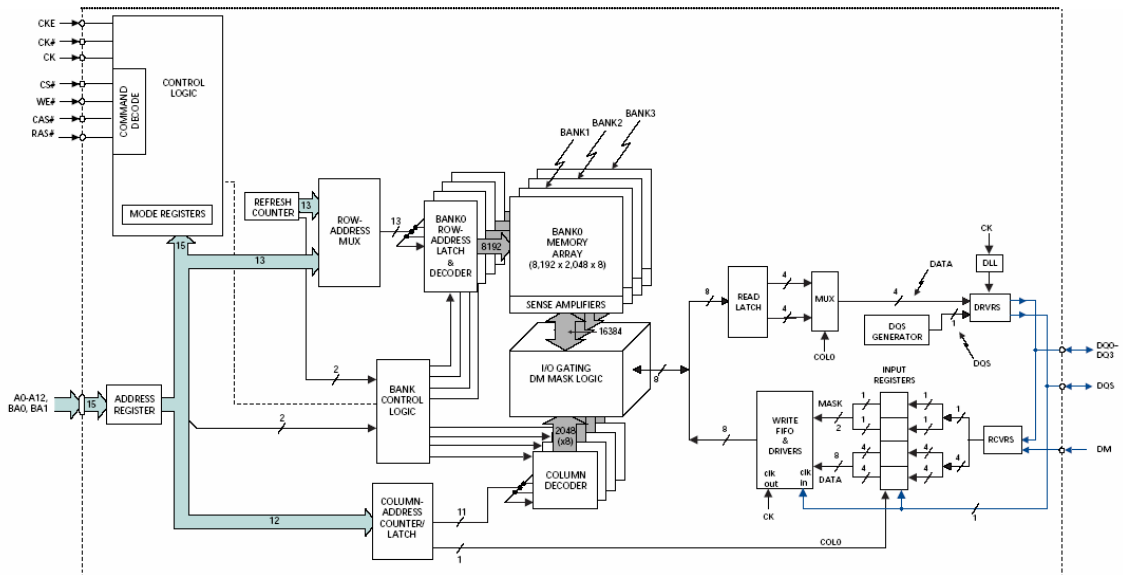


Figure A.5: Micron [70] 16-bit wide 512 Mbit DDR SDRAM.

Appendix B

Introduction to processing implementation techniques with FPGA

In the recent years the binary treatment of data has been boosted by the arrival on the market of microprocessors. Also in the domain of analogue signal processing fast ADCs and DACs together with DSPs have simplified the life of the designers. On the other hand several problems can only be executed in a pipe-lined way, due to time constraints. This is our case. For several problems it would be economically impossible to develop ASICs with the required logic burned in. The Field Programmable Gate Array (FPGA) are the key elements of several implementations when a reduced number of electronic boards need to be produced: from 1 to a few thousand. We discuss hereafter several aspects of this philosophy of construction.

The Field Programmable Gate Array, FPGA

FPGAs are programmable chips, providing connectivity and resources for digital signal processing. Three types of technologies are currently well know, non volatile, one time programmable (anti-fuse), non volatile, re-programmable (flash) and volatile (sram) FPGAs. The technology has evolved far from "an array of gates" and has become very attractive for new designs where fast development and low cost at small quantities are important.

The FPGAs from different manufacturers are classed in families. Altera STRATIX [71] and Xilinx Virtex-II Pro [72] are examples of the two leading chip vendors. During the last years, the FPGAs didn't only grow in size but also changed the architecture significantly. A short introduction to the most important features for the two example families Stratix and Xilinx-II Pro are given below.

Distributed logic units

A modern FPGAs architecture is constructed from a sea of basic logic units, where each unit consists of a four-input look-up table (LUT), programmable register, and any associated specialized circuits, such as carry chain, cascade logic, primitive logic gates and multiplexers. Even the FPGAs contain additional dedicated circuits, the number of this generic basic units can be used to measure the size of a device. In the Altera Stratix

architecture the basic cell is the Logic Element (LE) and 10 LEs form a Logic Array Block (LAB) (see figure B.1,B.2). In Virtex-II Pro terminology these are the Slice and the Configurable Logic Block (CLB). The LAB and the CLB provide about equal logic resources for a design.

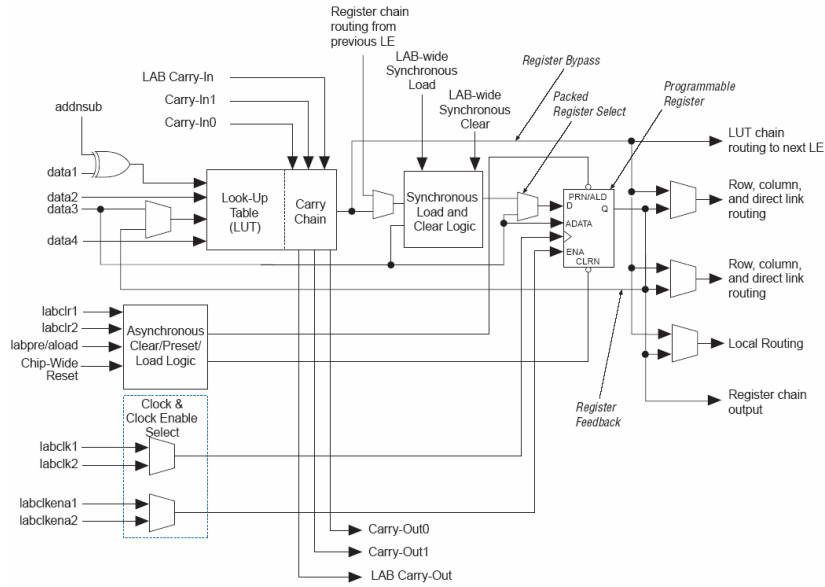


Figure B.1: Block diagram of a basic logic element (LE) of the Altera Stratix architecture. Picture taken from [73].

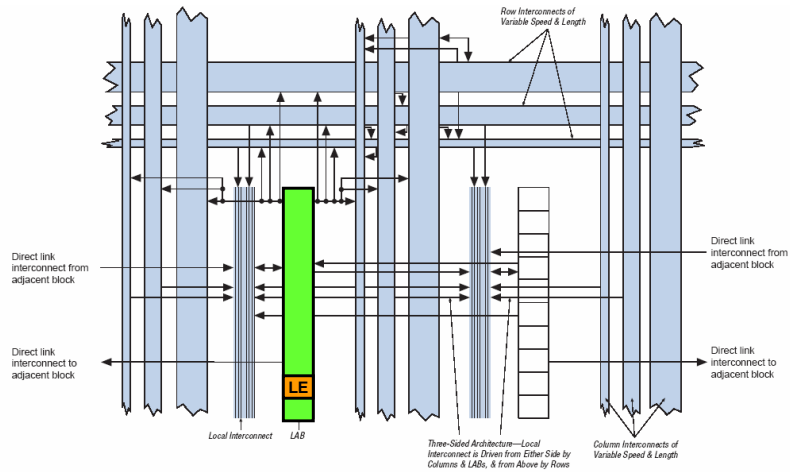


Figure B.2: Block diagram of a Logic Array Block (LAB) of the Altera Stratix architecture. The interconnection on the FPGA is organized on several hierarchies. Picture taken from [73].

On chip memory

Dedicated small block memories distributed over the chip has become the standard way to solve the memory needs on the chip. Dedicated programmable logic around these embedded block is used to define the memory operation as FIFO, RAM or ROM. As for the basic elements, different chip families use its specific bus and memory architecture. A summary of the functionality for the Stratix device is given in figure B.3.

Feature	M512 Block	M4K Block	M-RAM Block
Performance	319 MHz	290 MHz	287 MHz
Total RAM bits (including parity bits)	576	4,608	589,824
Configurations	512 × 1 256 × 2 128 × 4 64 × 8 64 × 9 32 × 16 32 × 18	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 16K × 36 8K × 64 8K × 72 4K × 128 4K × 144
Parity bits	✓	✓	✓
Byte enable		✓	✓
Single-port memory	✓	✓	✓
Simple dual-port memory	✓	✓	✓
True dual-port memory		✓	✓
Embedded shift register	✓	✓	
ROM	✓	✓	
FIFO buffer	✓	✓	✓
Simple dual-port mixed width support	✓	✓	✓
True dual-port mixed width support		✓	✓
Memory initialization (.mif)	✓	✓	
Mixed-clock mode	✓	✓	✓
Power-up condition	Outputs cleared	Outputs cleared	Outputs unknown
Register clears	Input and output registers (1)	Input and output registers (2)	Output registers
Same-port read-during-write	New data available at positive clock edge	New data available at positive clock edge	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data	Outputs set to unknown or old data	Unknown output

Figure B.3: Three types of embedded memory blocks with size 512 bit, 4 kbit and 512 kbit are implemented. All blocks support dual port operation with one clock domain per port. Picture taken from [73].

The large memory blocks (512 kbit) are designed to be on chip program and data memory to allow the use of fast microprocessors.

IO pins

To interface the chip to the external world, configurable IO cells are used. The range of signaling standard support goes from single ended 3.3V-LVTTL over 1.5V-HSTL to differential LVDS. A schematics of the IO element of the Stratix family is given in figure B.4.

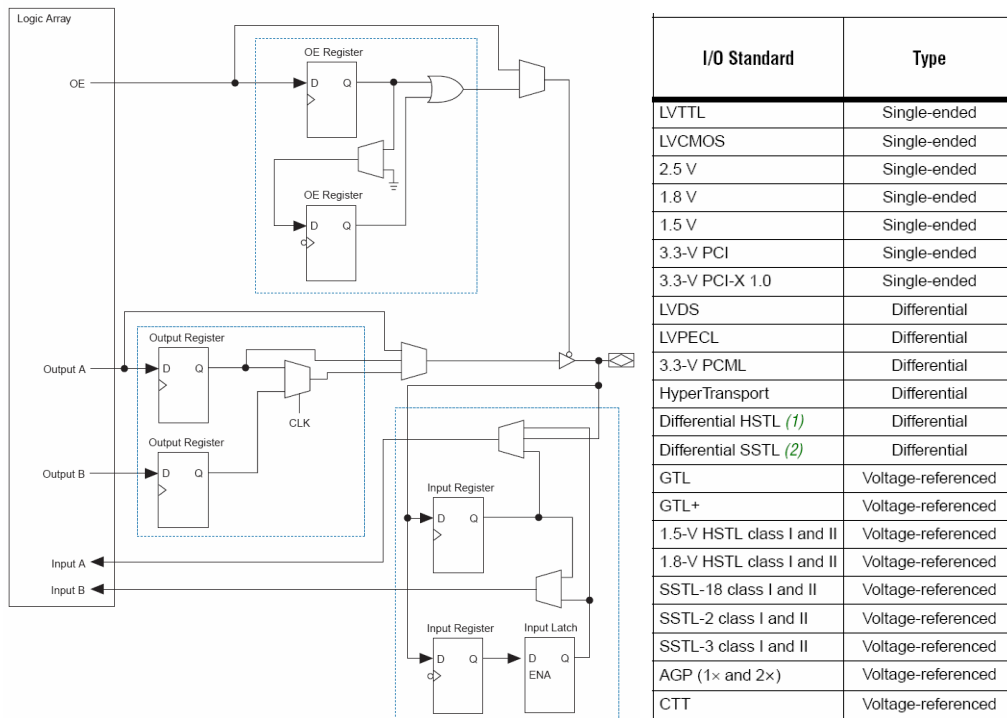


Figure B.4: *Stratix IO element on the left and a list of the supported signaling standards on the right. Picture taken from [73].*

The possibility to interface to a wide range industry standards ensures the possible usage of the FPGA for a give application. Interfacing memory chips as DDR SDRAM require in addition to the correct IO standard as well dedicated circuitry for the data strobe signal which are also implemented. The high number of IO pins required and the enhanced timing performance makes all high density FPGAs being packaged in Ball Grid Array (BGA) offering up to 1500 GPIOs.

Clock management

To cope with complex designs where interfaces to several external devices is required, on chip clock management based on Phase Lock Loop (PLL) circuits are provided. A typical Altera Stratix device has 6 independent PLLs integrated.

High speed transceivers

Interfacing to optical devices running at GHz speed has been done traditionally by dedicated high speed transceiver chips. Some of the latest families of FPGAs for example Stratix GX and Virtex-II Pro integrate these circuitry for serializing and de-serializing including data encoding on chip.

Embedded DSP blocks

Using the distributed basic logic cells to form large multipliers takes a lot of resources. In the Stratix and since the Virtex-II, hardwired (not configurable) arithmetic units called DSP blocks are included in the design. The typical digital signal operations, Multiply ACcumulate (MAC), can be performed in these "ASIC like" circuits and save significant distributed logic for other design parts. The DSP blocks can be configured to operate in different modes. For example out of one 36-bit by 36-bit multiplier (one DSP block), 8 independent 9-bit by 9-bit operation can be performed simultaneously.

Abbreviations

A-RxCard	Analog Receiver Card
ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
BGA	Ball Grid Array
CAT6	Category 6 (common network cable)
CCPC	Credit Card PC
CKM	Cabibbo Kobayashi Maskawa
CLB	Configurable Logic Block
CM	Common Mode
CMS	Common Mode Suppression
CP	Charge Parity transformation
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DAC	Digital to Analog Converter
DAQ	Data acquisition for L1T and HLT trigger data
DDR	Double Data Rate
DSP	Digital Signal Processor
ECS	Experiment Control System
FEM	Front end EMulator
FIFO	First In First Out buffer
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
GBE	Gigabit Ethernet
GMII	Gigabit Medium Independent Interface, 8-bit parallel PHY interface (networking terminology)
GOL	Cern implementation of a radiation hard 1.6-Gbit/s serializer
GPIO	General Purpose Input Output
HEP	High Energy Physics
HLT	High Level Trigger
HST	Hubble Space Telescope
HSTL	High Speed Transceiver Logic
HUDF	Hubble Ultra Deep Field
I2C	Inter-Integrated circuit Control bus (Philips Semiconductors)
IO	Input Output

IT	Inner Tracker of LHCb
JTAG	Joint Test Action Group
L0DU	Level 0 Decision Unit
LAB	Logic Array Block
LAN	Local Area Network
LCMS	Linear Common Mode Suppression
LE	Logic Element
LEP	Large Electron Collider
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signalling
MAC	Multiplication Accumulation or Medium Access Controller (networking terminology)
MEP	Multi Event Packet. Term used for an aggregation of several events to one packet in order to achieve maximal performance on the Gigabit Ethernet based read out network.
MSB	Most Significant Bit
MXI	Memory eXtension Interface
NIC	Network Interface Card
O-RxCard	Optical Receiver Card
OSI	Open Systems Interconnect Model (networking terminology)
OT	Outer Tracker of LHCb
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PHY	Physical layer device (networking terminology)
PLL	Phase Lock Loop
POS	Saturn compatible Packet Over Sonet interface level 3 used for 1 Gigabit Ethernet
PP-FPGA	Pre Processor FPGA
QDR	Quad Data Rate
RAM	Random Access Memory
RF	Radio Frequency
RICH	Ring Imaging Cherenkov detector
RO-TxCard	Read Out Transmitter Card
ROM	Read Only Memory
RS	Readout Supervisor
RxCard	Receiver Card (stands for A-RxCard and O-RxCard)
SDR	Single Data Rate
SERDES	Serializer and de-serializer circuit
SI	Signal Integrity
SM	Standard Model
SPI-3	SATURN Development Group's "POS-PHY Level 3" interface (networking terminology)
SSTL	Stub Series Transceiver Logic
ST	Silicon Tracker of LHCb (IT and TT together)

SUSY	Super Symmetry
SyncLink-FPGA	Synchronization and Link FPGA
TDR	Technical Design Report
TELL1	Trigger ELectronics and L1 board
TFC	Timing and Fast Control
TLK2501	Texas Instruments SERDES chip
TTC	Timing and Trigger Control
TTCrx	TTC receiver chip
TTL	Transistor Transistor Logic
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VME	Versa Module Eurocard
VTT	Voltage Termination Terminal
VeLo	The Vertex Locator of LHCb
Via	Vertical Interconnect on PCB

Bibliography

- [1] NASA web site.
http://www.nasa.gov/vision/universe/starsgalaxies/hubble_UDF.html
- [2] A. D. Sakharov, JETP Lett. 6,(1967) 21.
- [3] M. Kajantie et al. Phys. Rev. Lett. **77**, 2887, 1996.
- [4] S. Amato et al. (LHCb collab.), “LHCb Technical Proposal: A large hadron collider beauty experiment for precision measurements of CP violation and rare decays”, CERN/LHCC/1998-4, LHCC/P4, February 1998.
- [5] LHCb reoptimized Detector Design and Performance Technical Design Report, LHCb, CERN LHCC 2003-030.
- [6] S. Amato et al. (LHCb collab.), “LHCb magnet technical design report”, LHCb TDR 1, CERN/LHCC 2000-007, December 1999.
S. Amato et al. (LHCb collab.), “LHCb calorimeters technical design report”, LHCb TDR 2, CERN/LHCC 2000-036, September 2000.
S. Amato et al. (LHCb collab.), “LHCb RICH technical design report”, LHCb TDR 3, CERN/LHCC 2000-037, September 2000.
P.R. Barbosa Marinho et al. (LHCb collab.), “LHCb muon system technical design report”, LHCb TDR 4, CERN/LHCC 2001-010, May 2001.
P.R. Barbosa Marinho et al. (LHCb collab.), “LHCb vertex locator technical design report”, LHCb TDR 5, CERN/LHCC 2001-011, May 2001.
P.R. Barbosa Marinho et al. (LHCb collab.), “LHCb outer tracker technical design report”, LHCb TDR 6, CERN/LHCC 2001-024, September 2001.
P.R. Barbosa Marinho et al. (LHCb collab.), “LHCb online system (data acquisition and experiment control) technical design report”, LHCb TDR 7, CERN/LHCC 2001-040, December 2001.
A. Franco Barbosa et al. (LHCb collab.), “LHCb inner tracker technical design report”, LHCb TDR 8, CERN/LHCC 2002-029, November 2002; see <http://lhcb.web.cern.ch/lhcb/TDR/TDR.htm>.
R. Antunes Nobrega et al. (LHCb collab.), LHCb reoptimized detector (design and performance) technical design report, CERN/LHCC 2003-030, LHCb TDR 9, September 2003.
R. Antunes Nobrega et al. (LHCb collab.), LHCb trigger system technical design report, CERN/LHCC 2003-031, LHCb TDR 10, September 2003.

- [7] CERN Public web pages.
<http://www.cern.ch>
- [8] CERN document server.
<http://weblib.cern.ch>
- [9] LHC Machine Home Page
<http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/>
- [10] J.-C. Dran “Accelerators in Art and Archaeology” presented at the 8th European Particle Accelerator Conference EPAC 2002, Paris, France, June 2002
- [11] LHCb Collaboration, P.R. Barbosa Marinho *et al.*, Vertex Locator Technical Design Report, CERN–LHCC/2001–11.
- [12] LHCb Collaboration, S. Amato *et al.*, RICH Technical Design Report, CERN–LHCC/2000–37.
- [13] LHCb Collaboration, S. Amato *et al.*, Magnet Technical Design Report, CERN–LHCC/2000–7.
- [14] LHCb Collaboration, P.R. Barbosa Marinho *et al.*, Outer Tracker Technical Design Report, CERN–LHCC/2001–24.
- [15] LHCb Collaboration, A. Franca Barbosa *et al.*, Inner Tracker Technical Design Report, CERN–LHCC/2002–29.
- [16] LHCb Collaboration, R. Antunes Nobrega *et al.*, Trigger Technical Design Report, CERN–LHCC/2003–31.
- [17] LHCb Collaboration, S. Amato *et al.*, Calorimeter System Technical Design Report, CERN–LHCC/2000–36.
- [18] LHCb Collaboration, P.R. Barbosa Marinho *et al.*, Muon System Technical Design Report, CERN–LHCC/2001–10.
- [19] LHCb Collaboration, Addendum to the Muon System Technical Design Report, CERN–LHCC/2003–2.
- [20] M. Ashton *et al.*, “Status report on the RD12 project: timing, trigger and control systems for LHC detectors”, CERN/LHCC 2000-002, LEB status report/RD12 (2000).
- [21] Y. Ermoline, “Vertex detector electronics – ODE to ECS interface”, IPHE note **2000-007**, LHCb note **2000-012**.
- [22] Y. Ermoline, “Vertex detector electronics – L1 electronics prototyping”, LHCb note **TRAC 1998-069**.
Y. Ermoline, “Vertex detector electronics – Off-detector electronics pre-prototype”, IPHE note **2000-008**, LHCb note **VeLo 2001-057**.

- [23] A. Bay et al., “Tests on the L1-electronics board prototype RB2”, IPHE note **2002-008**, LHCb note **2002-033**.
- [24] Y. Ermoline, “Vertex Detector Electronics RB3 Specification”, IPHE note 2001-002, LHCb note **2001-050**.
- [25] CMS Outreach
http://cmsinfo.cern.ch/Welcome.html/CMSdocuments/CMSchallenges/CMSchallenges_index.html
- [26] Atlas L1 Trigger web page
<http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/LEVEL1/level1.html>
- [27] N. Ellis, First-level trigger systems at LHC, LECC Workshop, Colmar, 9-13 September 2002
<http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/LEVEL1/LEB0902ellis.pdf>
- [28] Richard Jacobsson, Beat Jost, “Timing and Fast Control”, LHCb Note 2001-16.
- [29] Richard Jacobsson, Beat Jost, Zbigniew Guzik, “Readout Supervisor Design Specifications”, LHCb Note 2001-012.
- [30] Jorgen.Christiansen, “Requirements to the L0 front-end electronics”, LHCb Note **2001-014**.
- [31] Jorgen.Christiansen, “Requirements to the L1 front-end electronics”, LHCb Note **2003-078**.
- [32] EDMS Cern <https://edms.cern.ch>
- [33] Beetle 1.3 Reference Manual
http://wwwasic.kip.uni-heidelberg.de/lhcb/Publications/BeetleRefMan_v1_3.pdf
- [34] Aurelio Bay, Guido Haefeli, Patrick Koppenburg “LHCb VeLo Off Detector Electronics Preprocessor and Interface to the Level 1 Trigger”, LHCb Note **2001-043**.
- [35] Mike Koratzinos. The Vertex Detector Trigger Data Model. LHCb TRIG **1998-070**.
- [36] Niels.Tuning “L1-type Clustering in the VeLo on Test-beam Data and Simulation”, LHCb Note **2003-073**.
- [37] Guido Haefeli *et al.* “TELL1 Specification for a common read out board for LHCb”, IPHE Note **2003-02**, LHCb Note **2003-007**.
- [38] Altera Apex 20K FPGA family
<http://www.altera.com/literature/lit-apx.jsp>
- [39] Texas Instruments DSP Home Page
<http://dspvillage.ti.com/docs/dspvillagehome.jhtml>

- [40] Federica Legger, Guido Haefeli, Aurelio Bay, Laurent Locatelli “A L1 Buffer implementation with DSP technology for LHCb Readout Board”, LHCb Note **2003-109**.
- [41] Hans Muller, Jose Toledo, Angel Guirao, Francois Bal “Readout Unit”, LHCb Note **2001-136**.
- [42] A Barczyk, J-P. Dufey, B. Jost and N. Neufeld “A common implementation of the Level 1 trigger and HLT Data Acquisition”, LHCb Note **2003-079**.
- [43] Owen Boyle, Robert McLaren, Erik van der Bij, “The S-Link Interface Specification”, <http://hsi.web.cern.ch/HSI/s-link/spec/spec/s-link.pdf>
- [44] Information on the SCI standard can be found on the web <http://www.dolphinics.no/sci/>
- [45] A. Walsch, “Architecture and Prototype of a Real-Time Processor Farm Running at 1 MHz”, Inaugural Doctoral Thesis, University of Mannheim, Sept. 2002. (online: http://bibserv7.bib.uni-mannheim.de/madoc/volltexte/2002/58/pdf/58_1.pdf)
- [46] B. Jost and N. Neufeld, “A Versatile Network Processor based Electronics Module for the LHCb Data Acquisition System”, LHCb 2001-132.
- [47] Hans Muller, Angel Guirao, Francois Bal, Xue Tao, “HLT and L1T data streams”, LHCb Note **2004-028**.
- [48] SRAM based memory family supported by leading semiconductor vendors <http://www.qdrsram.com>
- [49] B. Jost and N. Neufeld, “Raw-data transport format”, LHCb 2003-063.
- [50] POS-PHY. SATURN Compatible Packet Over SONET Interface Specification for Physical Layer Devices (Level 3). PMC-980495, Issue 3, November 1998.
- [51] Dirk Wiedner *et al.* “Prototype for an Optical 12 input Receiver Card for the LHCb TELL1 Board”, LHCb Note **2003-137**.
- [52] P. Moreira et al. “GOL reference manual” http://projgol.web.cern.ch/proj-gol/gol_manual.pdf
- [53] TLK2501 1.6 to 2.5 GHz deserializer - Texas Instruments <http://focus.ti.com/lit/ds/symlink/tlk2501.pdf>
- [54] H. Deppe, M. Feuerstack-Raible, U. Stange, U. Trunk, U. Uwer “OTIS A Radiation Hard TDC For LHCb” <http://doc.cern.ch/archive/cernrep/2002/2002-003/p87.pdf>
- [55] Rutger Hierck, Marcel Merk, Matthew Needham “Outer Tracker occupancies and detector optimisation”, LHCb 2001-093.

- [56] Intel IXF1104 4-port Gigabit Ethernet MAC,
<http://www.intel.com/design/network/products/lan/controllers/ixf1104.htm>
- [57] Howard.W.Johnson,Martin.Graham, “High-Speed Digital Design, a handbook of black magic”, Prentice Hall PTR, Englewood cliffs, New Jersey, 1993.
- [58] Stephen H. Hall, Garrett W. Hall, James A.McCall, “High-Speed Digital System Design”, John Wiley and Sons, Inc, 2000.
- [59] Polar Home Page
<http://www.polarinstruments.com/>
- [60] SPECCTRAQuest Signal Explorer, PCB simulation software from Cadence
http://www.cadence.com/products/si_pk_bd/specctraquest/index_ds.aspx
- [61] IO buffer specification homepage, IBIS
<http://www.eigroup.org/ibis/ibis.htm>
- [62] Application note “Termination for Point-to-Point systems”
<http://download.micron.com/pdf/technotes/TN4606.pdf>
- [63] EBV website
<http://www.ebv.com>
- [64] Avnet website
<http://www.em.avnet.com>
- [65] JEDEC Stub Series Transceiver Logic for 2.5 Volts(SSTL_2)
<http://www.jedec.org/download/search/jesd8-9b.pdf>
- [66] JEDEC High Speed Transceiver Logic for 1.5 Volts(HSTL)
<http://www.jedec.org/download/search/jesd8-6.pdf>
- [67] IDT homepage
<http://www.idt.com>
- [68] Cypress homepage
<http://www.cypress.com>
- [69] Infineon homepage
<http://www.infineon.com>
- [70] Micron homepage
<http://www.micron.com>
- [71] Altera homepage
<http://www.altera.com>
- [72] Xilinx homepage
<http://www.xilinx.com>

- [73] Altera Stratix FPGA datasheet
http://www.altera.com/literature/lit-stx.jsp?xy=dev_sdh