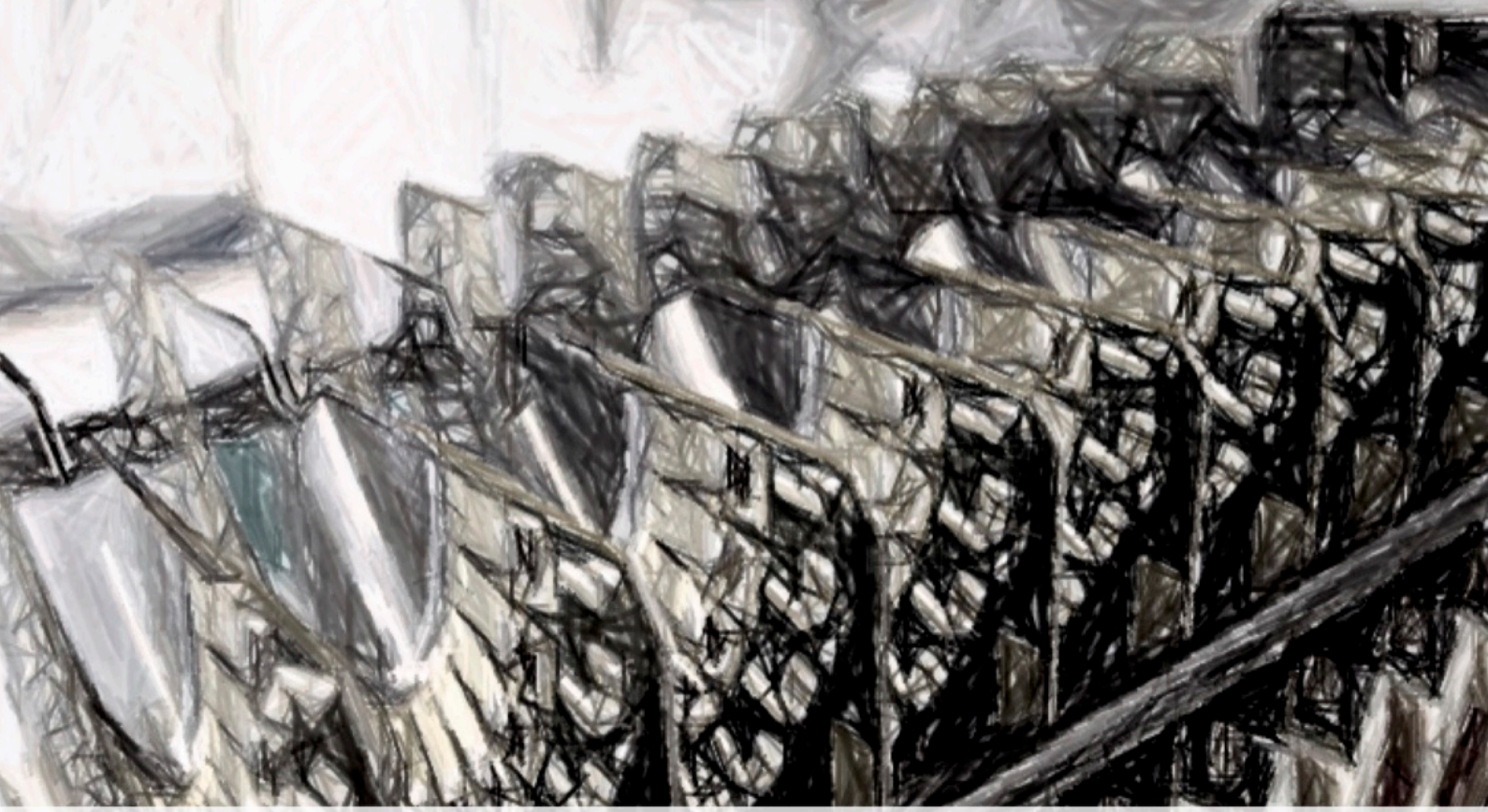


Accreditation to conduct researches

LHC backstage

From LHCb to CMS, through ATLAS

SEBASTIEN VIRET

HABILITATION À DIRIGER DES RECHERCHES

de l'Université Claude Bernard Lyon 1

préparée au sein de l'Institut de Physique des 2 Infinis de Lyon

Spécialité : **Physique des hautes énergies**

présentée par

SEBASTIEN VIRET

LHC backstage

From LHCb to CMS, through ATLAS

Soutenue publiquement le 25 Janvier 2021 devant le Jury composé de :

PAULA COLLINS	Rapporteur	CERN
CHRISTOPHER PARKES	Rapporteur	University of Manchester
LAURENT SERIN	Rapporteur	CNRS/IN2P3
GIOVANNI CALDERINI	Examinateur	CNRS/IN2P3
DAVID CALVET	Examinateur	CNRS/IN2P3
JACQUES MARTEAU	Examinateur	Université Claude Bernard Lyon 1

To Laure, Lisa, Raphaël, and Lucie

CONTENTS

I Introduction 1

II From VELO to stubs 5

1 Contribution to the LHC projects 7
1.1 Few words on the LHC . . . 7
1.2 The aim of calibration . . . 8
1.3 LHCb vertex detector alignment: 2005-2007 . . . 9
1.3.1 The LHCb experiment . . . 9
1.3.2 The alignment problem . . . 11
1.3.3 Personal contribution . . . 12
1.4 Commissioning of ATLAS TileCal LASER calibration system: 2008-2010 . . . 14
1.4.1 The ATLAS experiment . . . 14
1.4.2 The ATLAS tile calorimeter and its calibration . . . 14
1.4.3 LASER system in few words . . . 16
1.4.4 Personal contribution . . . 18

2 The HL-LHC project and CMS stubs 19
2.1 The HL-LHC project . . . 19
2.2 The upgraded CMS tracker . . . 20

III Extracting data 25

3 Tracker frontend 27
3.1 Frontend chain description . . . 27
3.1.1 Principle . . . 27
3.1.2 2S module readout chain . . . 28
3.1.3 PS module readout chain . . . 29
3.2 The stub: fundamental object of the future tracker . . . 30
3.2.1 General description . . . 30

3.2.2 Relationship between stub width and transverse momentum 30
3.2.3 Parallax correction 31
3.3 Stub selection 33
3.3.1 Typology 33
3.3.2 Stub selection at the frontend level 34
3.3.3 Using stub bend in track reconstruction 37

4 Development and realisation of the concentrator 41
4.1 The role of a data aggregator 41
4.1.1 Why an ASIC? 41
4.1.2 General description 42
4.2 Requirements 43
4.2.1 Context 43
4.2.2 CIC working principle 44
4.2.3 I/O data formats definition, study of the dynamic losses. 45
4.2.4 Technical specifications 49
4.3 CIC model 50
4.3.1 Overall description 50
4.3.2 CIC core description 51
4.3.3 Model verification 53
4.4 CIC physical design 54
4.4.1 Design flow description 54
4.4.2 From model to foundry 54
4.5 CIC1 characterisation 57
4.5.1 Test bench description 57
4.5.2 Results 58
4.5.3 CIC1 in 2S module frontend chain 60
4.6 CIC2 development 60
4.6.1 Description and expected performances 60
4.6.2 Next steps 61
4.7 Summary 61

IV Analyzing data 63

5 Building tracks, faster... 65

5.1 Introduction . . . 65
5.1.1 Why? . . . 65
5.1.2 State of the art . . . 66
5.2 A track trigger for CMS . . . 67
5.2.1 Problem . . . 67
5.2.2 Fast pattern recognition in CMS . . . 69

6 An associative memory based approach for CMS 71
6.1 Algorithm description . . . 71
6.1.1 Foreword . . . 71
6.1.2 Bank, patterns, and superstrips . . . 72
6.1.3 Superstrip definition . . . 73
6.1.4 Introducing ternary bits . . . 74
6.2 An AM chip for CMS . . . 76
6.2.1 Bank generator . . . 76
6.2.2 Emulator . . . 78
6.3 Algorithm performance in CMS . . . 79
6.3.1 Bank optimisation . . . 79
6.3.2 Efficiency and fake rate definition . . . 81
6.3.3 Results . . . 83
6.3.4 Summary . . . 85

7 Towards a full demonstrator: the FastTrack project 87
7.1 Introduction . . . 87
7.2 Filter the matched roads: the track candidate builder . . . 88
7.2.1 Principle . . . 88
7.2.2 Algorithm principle . . . 89
7.2.3 Performance of the complete algorithm . . . 92
7.3 Demonstrator . . . 95
7.3.1 General description . . . 95
7.3.2 Algorithm implementation . . . 98
7.3.3 Performance and data/simulation comparisons . . . 99
7.4 Towards a complete system . . . 100
7.5 Concluding remarks . . . 101

V Conclusion **103**

VI Appendix **121**

A LHCb notes **123**

A.1 Alignment procedure 124

A.2 Alignment tests 125

B ATLAS notes **126**

B.1 LASER description 127

B.2 LASER commissioning 128

C CMS notes **129**

C.1 Stubs study 130

C.2 Front end dynamic inefficiencies 131

C.3 CIC1 specifications 132

C.4 Track trigger development 133

1

INTRODUCTION

INTRODUCTION

Fifteen years have passed since my PhD thesis defence in September 2004. At that time, the Large Hadron Collider (LHC) was still under construction and the Higgs boson a hypothesis. Things have changed since ....

Starting with the LHC itself. After a false start in 2008 [1], the collider has been operating without any major issue since 2010. Since then, enormous amounts of data were produced, leading to many discoveries, the most famous being the Higgs boson in 2012.

All these discoveries would probably not have taken place, at least not so promptly, if the huge and complex LHC machinery hadn't worked so well. The same could be said of the four detectors assembled around collision points: CMS, ATLAS, LHCb and ALICE. Before the results, there are detectors, and behind the detectors, there are researchers, engineers, and technicians who have spent many years developing, commissioning, and operating them. LHC is also exciting backstage...

Since 2005, most of my activities have taken place around there, in three of the four main experiments: LHCb (2005-2007), ATLAS (2008-2010), and finally CMS (2011-2019). This document, although it covers all of my contributions, pays particular attention to CMS.

The first part begins with a reminder of my work in LHCb and ATLAS (Chapter 1), and ends with a brief description of my activities in CMS (Chapter 2), where I have devoted since 2012 most of my attention to the tracker upgrade project. This is actually the main subject of the following two parts.

The second part deals with frontend and data extraction protocol of the future tracker. This is an essential element of this detector, because data will have to be used in a very short latency of a few microseconds. The problem is described in more detail in Chapter 3. Chapter 4, on the other hand, presents the development of an essential element of the extraction chain: the concentrator chip.

Finally, the third and final part focuses on how to use this information to reconstruct tracks in real time. An inventory of the existing techniques is drawn up in Chapter 5, then the potential of one of these techniques in the future tracker is evaluated in Chapter 6. Finally, Chapter 7 presents the demonstrator that has been developed to confirm those results.

[1] This was actually not a false start for calibration, where a lot was done with pre-2010 data, for example alignment with TEDs runs,...

II

FROM VELO TO STUBS

1

CONTRIBUTION TO THE LHC PROJECTS

This chapter is a quick overview of my contributions to the LHCb and ATLAS experiments between 2005 and 2010. This work was part of the LHC preparation for the start-up, and was mainly devoted to setting up calibration systems.

1.1/ FEW WORDS ON THE LHC

If you opened this document, you probably have already heard about the LHC. So I will not spend pages describing the Large Hadron Collider that has been in operation since 2010 at CERN in Geneva, many books have already done that in a very detailed and satisfactory way (see [1] among others).

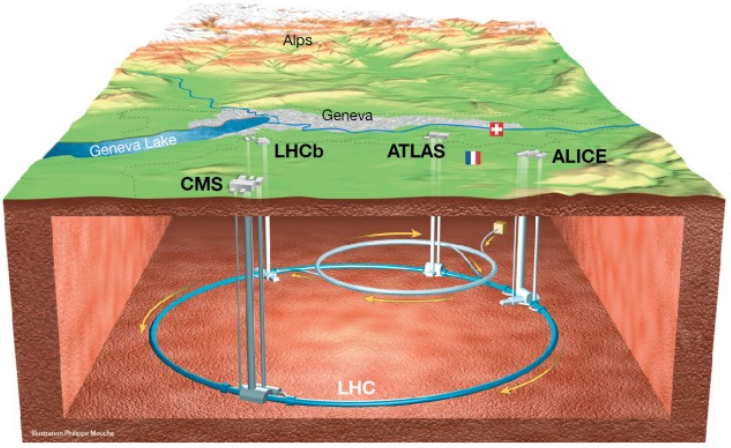


Figure 1.1: LHC

A few words though... The LHC collides particles at an unprecedented level of energy. These collisions occur at four points in the 27 km underground ring. Around each of these

points lies a detector that analyses these collisions. Among these four experiments, there are two more generic systems: ATLAS and CMS, and two more specific detectors: ALICE and LHCb.

The two main objectives of ATLAS and CMS are:

- **Understand** the origin of the mass by establishing the existence of the Higgs boson. This first objective was achieved in 2012 [2, 3].
- **Observe** matter with unprecedented precision in order to search for new phenomena or unknown objects. Indeed one assumes that the standard model is an effective theory, valid up to an energy of the order of TeV $(10^{12}eV)$. Beyond that, new phenomena are expected. Their observation would open a new chapter of our discipline, and prepare the ground for future experiments.

LHCb is an experiment more specifically devoted to the study of heavy quark flavors (b and c), and the ALICE physics program is dedicated to the precise study of the plasma of quarks and gluons, a particular state of matter.

The common point of all these experiments is their complexity. These detectors are giant puzzles built to detect and measure the properties (electric charge, impulsion, energy) of the particles created during collisions of protons circulating in the LHC. Each of these puzzles contains about 100 million independent detection cells whose information must be read and analyzed in a very short time.

The assembly of these detectors, as well as their commissioning and monitoring, were quite an epic. Thousands of people from all over the world contributed to their construction. Hundreds of people contribute every day to their proper functioning.

Without this essential work, the many discoveries of the LHC would probably not have happened, at least not so quickly. The most beautiful restaurant in the world is nothing without its kitchens ...

1.2/ THE AIM OF CALIBRATION

When I defended my PhD thesis in 2004 [4], most of the detectors that currently operate at the LHC were just entering their construction phase. The research and development stages that preceded this phase had been successfully completed and we had to think about their commissioning.

Indeed, the detectors installed around the LHC are not only complex assemblies, they are also very fragile, and their instructions for use are not provided on delivery. Their smooth operation requires very strict monitoring.

The purpose of this monitoring is fairly simple: how to correctly interpret the detected signal and to make sure that it is always available when the collider delivers stable beams? How to make the signal measured by a detector correspond as accurately as possible to reality? The role of calibration is to provide answers to these questions.

Calibration is a vast discipline that depends on the signal being calibrated. For each situation the method is different, but it is always based on the same paradoxical observation:

how to improve the quality of a measurement starting from the only information available before correction, ie unscaled signals?

In a detector, time, energy deposits, and positions are the main observables. Calibration techniques logically cover each of these 3 categories.

At the LHC, time calibration is mainly done at system startup (see for example [5]). Synchronisation of sub-detectors is fundamental; the millions of signals sampled every 25 ns must be pooled in a rigorous way. The energy deposits of the collision N must indeed be compared with the tracks of the same collision. At the detector level, time can also be used to analyse the properties of an energy deposit. Until recently, however, the precision of time measurements did not make it a central part of LHC detection systems. This situation is now evolving rapidly, especially since it will now become necessary to distinguish the temporal origin of the particles within the same bunch crossing. New time-based detection systems are being developed [6], and it is likely that calibration methods in this sector will have to adapt to these new constraints in the coming years.

The calibration in position is also done in several stages, it relates more specifically to tracking detectors. A poorly reconstructed track can indeed radically distort the measured information. We must know precisely the position of each detection channel. For example, in the muon chambers of ATLAS, to estimate the muon charge for very large transverse momentum particle, one must be able to measure a deviation of about 30 micrometers over the entire radius of the detector (about ten meters).

Energy calibration is also a complex and fundamental task, especially since it is sometimes impossible, unlike the two previous cases, to have a precise reference to perform the correction. Difficult, indeed, to know precisely the initial energy of a jet. This area of calibration therefore implements very sophisticated methods and several levels of corrections are generally necessary before reaching an acceptable accuracy.

These last two categories constituted the essential core of my activities between 2005 and 2010. These years were divided in 2 distinct periods:

- **2005-2007**: work on position calibration in LHCb. This activity is described in more detail in Section 1.3.
- **2008-2010**: work on energy calibration in ATLAS. This activity is described in more detail in Section 1.4.

1.3/ LHCB VERTEX DETECTOR ALIGNMENT: 2005-2007

1.3.1/ THE LHCB EXPERIMENT

If ATLAS and CMS have a generalist vocation, LHCb [7] is an experiment resolutely turned towards heavy flavours. Its main objective is to study precisely the physics of the b and c quarks, an extremely rich domain that can provide precise information on the limits of the standard model.

The structure of LHCb (see Fig. 1.2) is relatively different from that of ATLAS or CMS. Since heavy quark pairs are mainly produced at high pseudo-rapidities, the detector does not need to be hermetic and it covers a much smaller acceptance, mainly in the forward

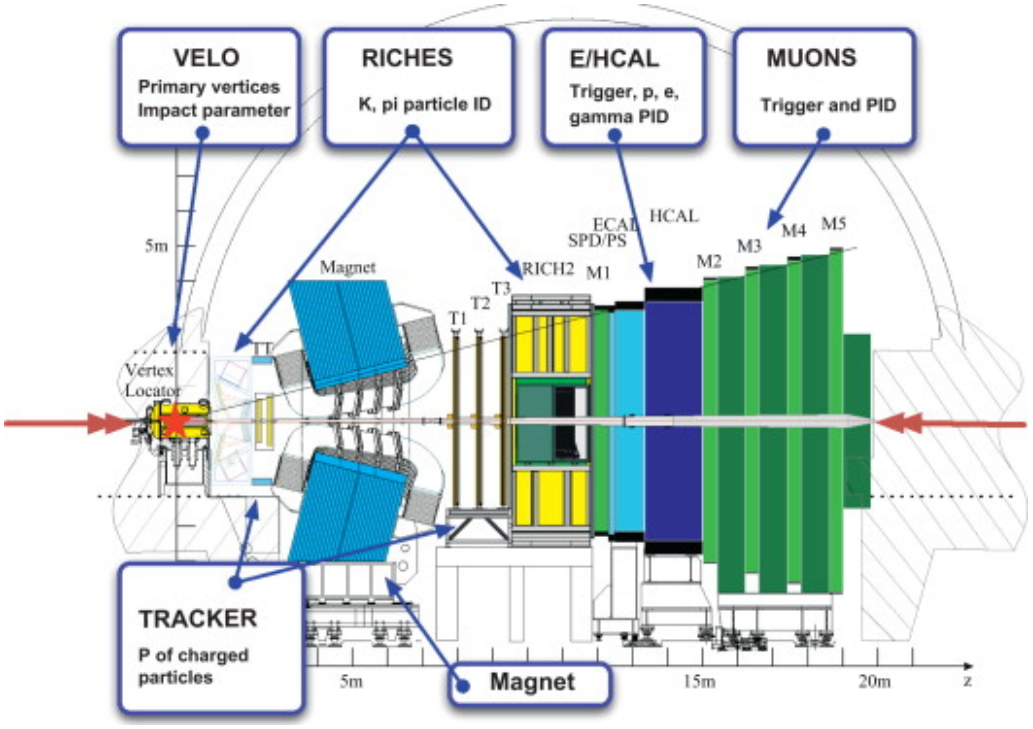


Figure 1.2: The LHCb detector

direction. On the other hand, LHCb is an assembly of sub-detectors whose detection principles are close to those of CMS and ATLAS. The main difference being the presence in LHCb of a Cherenkov imager (RICH), allowing precise identification of light mesons.

In a heavy flavour physics experiment, the identification of the hadron b or c decay vertex is a fundamental point. Indeed, given the relatively long lifetime of beauty and charmed mesons, this secondary vertex can be distant from the primary vertex by a few millimetres. The triggering system of LHCb is largely based on the identification of this particular signature, the detector performing this task is therefore essential.

Figure 1.3: LHCb VELO

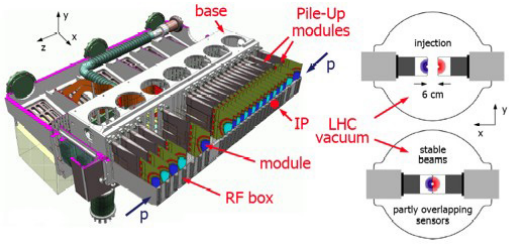


Figure 1.4: VELO working principle

This decay vertex detector is the VELO (contraction of VErtex LOcator) [8]. This pocket tracker (it measures barely 1 m long) must be able to identify a vertex with a spatial resolution of a few microns, near the point of interaction.

This detector (see Figures 1.3 and 1.4) consists of two boxes containing 21 stations,

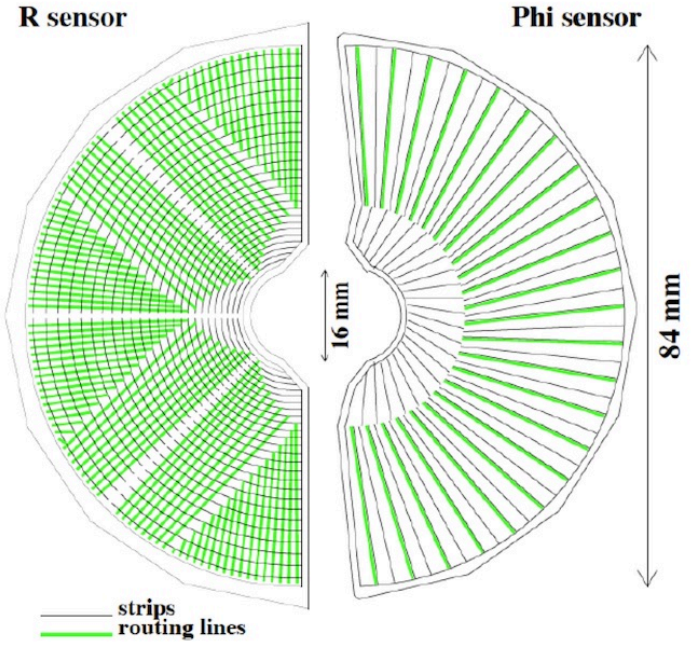


Figure 1.5: Detailed view of the strips arrangement of a VELO module

each station being itself made of two detectors comprising several thousand silicon strips. Tracking is done by these two detectors: one gives the radius, the other the azimuthal position. This novel strips arrangement is presented in Figure 1.5.

In addition to these modules, what makes the VELO very singular compared to other trackers is its proximity to the LHC beam (only 8 mm away). This proximity imposes an additional and very particular condition. In fact, in order to protect the detectors from the instabilities of the beam, the boxes containing the stations are retracted by 3 cm until the latter is stabilised.

However, to ensure optimum operation of the LHCb triggering system, it is necessary to know precisely the position of each detector, and this at any time during data-taking. The frequent movements of the VELO, as well as the accuracy necessary for the proper functioning of the detector, make this calibration particularly complex. It must not only be very precise, but also achievable in a few minutes at the beginning of each data-taking session.

1.3.2/ THE ALIGNMENT PROBLEM

The problem of alignment is explained in Figure 1.6. It shows a very basic detector consisting of 5 modules, crossed by a charged particle (blue line). As can be seen in the left figure, the 5 modules are not perfectly aligned. The line is a little forced in this example, the deformations are obviously much smaller in practice. But when you want to obtain a resolution of a few micrometers, a displacement of 50 microns (the thickness of a hair) must be corrected.

A problem occurs when one tries to reconstruct the track without knowing the deformations, ie considering that the modules are in their nominal place. This is what we see in

Figure 1.6: Alignment and track reconstruction

the figure on the right. In this case, the properties of the trajectory, and in particular its origin, are biased. It can even happen, if the deformations are too large, that the track is not detected at all. It is therefore necessary to establish a procedure to evaluate these deformations in order to realign the detector.

There are many methods to align a detector. Generally this operation is carried out in two stages. Firstly, a precise metrology is done, which makes it possible to know the position of the elements with an accuracy of a few tens of microns. This step serves as a starting point for numerical methods that use the detected signals to minimize the accuracy down to a few microns.

Several solutions are then possible, but the basic principle is always the same: one starts from a known signal (cosmic muon, decay vertex of a known particle) and one compares the reconstructed signal with the expected one. In our case, the signal can be the track position, the vertex, or the mass of the particle. The aim of the alignment algorithm is to minimise the difference between the measured value and the expected value. This gap is called the residual.

Proper alignment therefore requires 2 ingredients: a good algorithm, and a good data sample.

A good algorithm is a process that minimises residuals quickly without biasing the final result (some modes of distortion may be insensitive to minimisation). A good sample of data is a set of information that allows the algorithm to converge as quickly as possible. The more the topology of the tracks used is varied, the lower the risk of bias on the final result.

For a review of the different alignment techniques implemented at the LHC, reference can be made to the following document [9].

1.3.3/ PERSONAL CONTRIBUTION

My work between 2005 and 2007 was to develop and implement the alignment of the LHCb VELO. The procedure developed, which is still, as far as I know, used today, is described in detail in [10] (note reproduced in Appendix A.1).

A priori the VELO is a detector that lends itself relatively well to alignment. There is no magnetic field and few modules. It is therefore possible to implement non-iterative numerical methods (global chi-square minimisation) quite easily. The algorithm was therefore quickly chosen.

The complexity of the problem lies rather in the adaptation of the algorithm to the particular case of the VELO. The polar symmetry of the detection channels, among others, required special attention by setting up an intermediate alignment step. The choice of tracks necessary for the optimal alignment of the system, as well as their selection, also required the implementation of a dedicated strategy. Alignment can not be achieved with only tracks from collisions, the topologies have to be varied, and for example, tracks from muons in the beam halo that pass through the whole detector must be included. But to use them, you must have selected them upstream, and therefore have developed in advance the appropriate selection procedures.

The alignment procedure put in place was verified under real conditions during 2006 testbeam periods [11] (note reproduced in Appendix A.2).

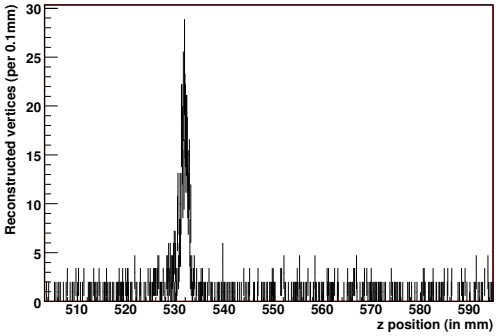


Figure 1.7: Vertices found before alignment

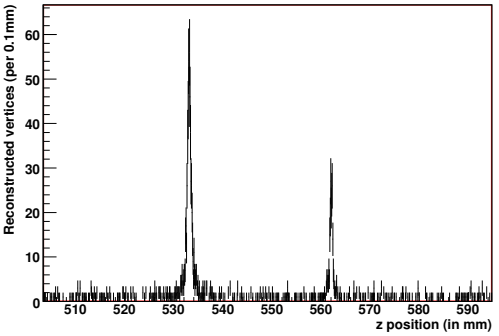


Figure 1.8: Vertex found after alignment

Figures 1.7 and 1.8 illustrate very simply the impact that the alignment can have on the quality of the reconstructed information. During beam tests, 2 targets had been placed upstream of the modules to be tested. The modules receiving the tracks from the interactions of the primary beam with the 2 targets, one could expect to observe a significant increase in the number of reconstructed vertices around these positions. Figure 1.7 shows that before correcting the alignment, only one of the 2 targets was visible. The tracks coming from the 2nd target were not reconstructed because the alignment of the modules receiving those tracks was slightly off. After applying the corrections obtained by the alignment procedure, the second target appeared and the number of interactions from the first target increases significantly (Figure 1.8). The effect is visible and immediate.

As said earlier, this procedure is still used to align the LHCb VELO at the beginning of each data acquisition. For the recorded data, the speed of execution is not a constraint and this alignment is completed by a more complex method which makes it possible to align all the tracking sub-detectors among each others [12].

1.4/ COMMISSIONING OF ATLAS TILECAL LASER CALIBRATION SYSTEM: 2008-2010

1.4.1/ THE ATLAS EXPERIMENT

Without minimising the importance of LHCb and ALICE, one can say that ATLAS and CMS are the two central LHC experiments. Their research program indeed covers a much wider range.

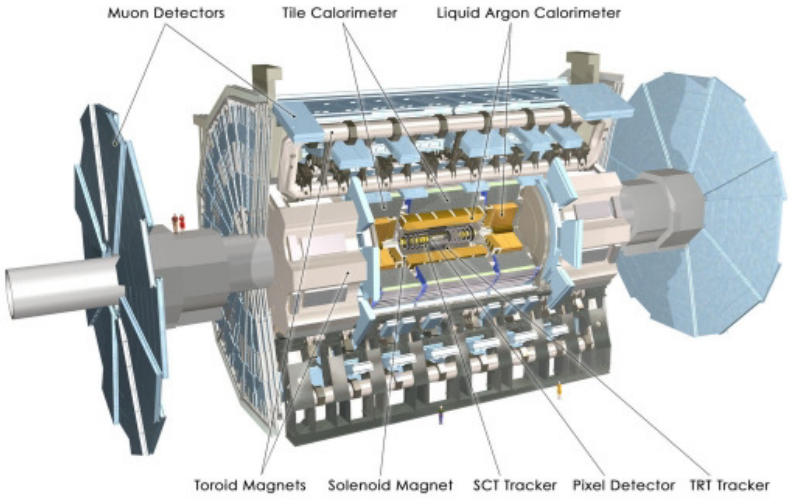


Figure 1.9: The ATLAS detector

As can be seen in the Figure 1.9, ATLAS [13] is a huge detector. Its complexity is huge too, the whole ATLAS detector containing about 100 million independent detection cells. This is one of the most complex machines made to date.

The size of the ATLAS collaboration is also impressive, several thousand people against a few hundred for LHCb (which is quite respectable, by the way). The organisation is necessarily very different. In ATLAS, each sub-detector is an experiment in itself, it is much more difficult to have an accurate perception of the whole detector. This is clearly another way of working, with its advantages and disadvantages.

1.4.2/ THE ATLAS TILE CALORIMETER AND ITS CALIBRATION

The Tile Calorimeter (TileCal) [14] is one of the ATLAS sub-detectors whose purpose is to measure the energy of the particles passing through it. This sub-detector is a sandwich of iron plates and scintillating tiles. The particle crossing the detector induces a shower of lower energy particles. The charged particles cause the tiles to scintillate, and the light thus created is transmitted via optical fibres to photo-multipliers (PMT) (Figure 1.10). The collected charges are then digitised and forwarded to the ATLAS acquisition system. The information of each PMT of the calorimeter (there are exactly 9852) are then summed in order to retrieve the energy of the initial particle.

In order to obtain the most accurate measurements possible, the energy resolution of the detector must be optimal. The purpose of the energy calibration is that the difference

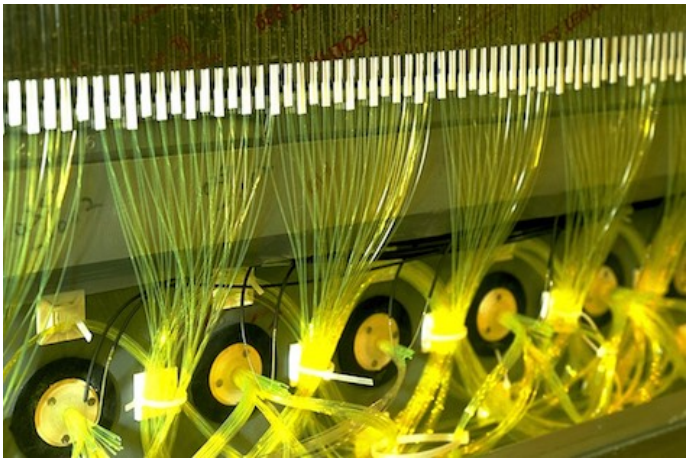


Figure 1.10: View of fibres and photomultipliers from a tile calorimeter sector

between the reconstructed energy and the real energy of the particle is the smallest possible.

To achieve this we implement 2 levels of calibration:

- **Low Level Calibration:** the TileCal is a complex system. This means that its proper functioning requires the monitoring and control of a large number of parameters (noise, PMT gains, ...), which depend on time, temperature, and so on... The knowledge of those informations is critical for ATLAS. For example, an ADC value will be totally useless if we do not have the corresponding noise level (pedestal). Therefore, all these parameters must be measured precisely and taken into account when reconstructing data (via databases).

- **High Level Calibration:** once the previous part is under control, the energy actually received by the detector is known precisely. Nevertheless, it remains necessary to correctly estimate the energy of the initial particle. Indeed, since the calorimeter has a finite granularity, the energy reconstructed by the latter will necessarily be biased. This is corrected by dedicated methods (jet reconstruction algorithm,...).

LASER is one of the low level calibration systems. It provides, along with 2 other systems (cesium and charge injection), a complete calibration of the acquisition chain. These systems make it possible to transform the signal effectively measured by the detector into an energy that can be used by data reconstruction software. Their respective roles are shown in Figure 1.11.

The cesium system [15] provides a calibration of the whole chain (except the readout electronics). A ^{137}Cs source is moved into the calorimeter. The signal then produced is collected and measured for each of the TileCal cells. It is thus possible to obtain a comparison of the response of each cell, and to derive the corrections which enable to uniforms the calorimeter response. It is indeed very important, in order to reach optimal resolution, that the answer of every cell to a given energy deposit gets identical.

The Cesium system measures a signal going through both the active part of the detector

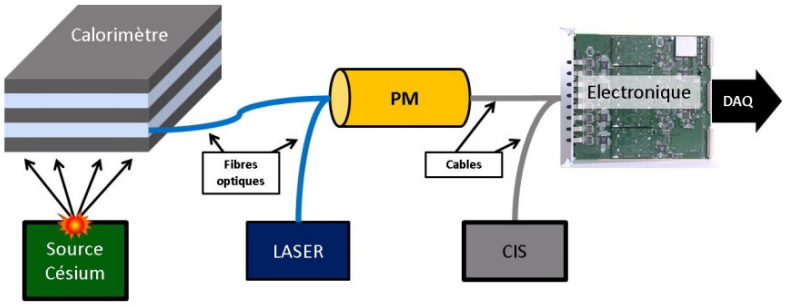


Figure 1.11: TileCal low level calibration scheme

and the PMT. However, if the energy measured in a cell is lower than normal, it may also be due to the instability of a PMT. It is therefore necessary to control these independently. This is where the LASER system [16] comes in. The light of a LASER is sent directly into the PMTs via optical fibres. Since the intensity of the light being sent is known with precision, it is possible to evaluate the stability of the PMT response, but also of the readout electronic (which this time is taken into account) by reconstructing the signal sent. Moreover, the information thus obtained can be used to equalise the response of the calorimeter between 2 Cesium analyses (these are actually carried out every 3-4 months only).

The LASER system also allows to monitor the readout electronics, but in order to characterise the problems specific to this part of the detector, a charge injection system [17] complements the low level calibration procedure.

The Clermont ATLAS group, which I joined after my recruitment at CNRS in 2007, is at the origin of the development of the LASER calibration system. Its contribution concerns, besides the LASER himself, all the control electronics and the whole internal calibration procedure. I was responsible for this system between 2008 and 2010.

1.4.3/ LASER SYSTEM IN FEW WORDS

1.4.3.1/ MECHANICS

The origin of the LASER system in ATLAS dates back to 1993 [18], but this is a technique which was already used successfully with many experiments in high energy physics (for example in the electromagnetic calorimeter of CMS [19]). The idea is to send short light pulses into the PMTs of the detector. They reproduce relatively well the signals created by real particles. The advantage, in the case of LASER, is that one knows precisely the energy of the pulse emitted. This makes it possible to monitor and control efficiently the stability and the linearity of the PMTs.

The system is divided into two distinct parts, the LASER box (figure 1.12) and the light distribution system (figure 1.13).

The LASER box is the core of the system. It contains all the elements necessary to precisely measure the amplitude of the LASER pulse emitted, but also to control its triggering. It is indeed possible to send LASER pulses during the data taking in order to follow the PMTs evolution in real time.

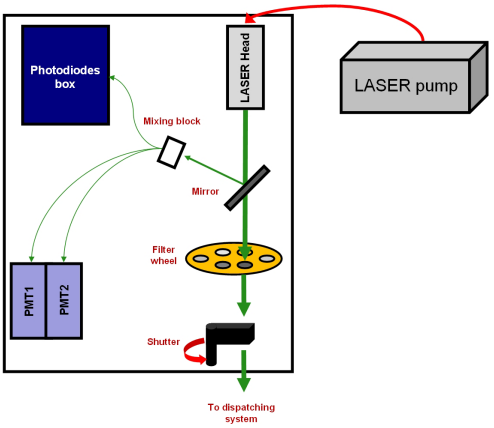


Figure 1.12: LASER box description

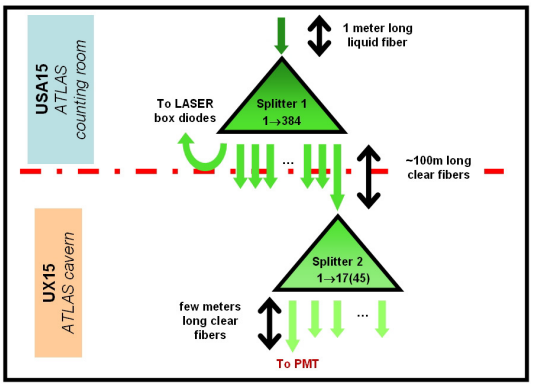


Figure 1.13: LASER system light distribution scheme

Measuring the amplitude of the LASER signal emitted is very important in order to calibrate the signals measured in the PMTs and to detect potential instabilities of the LASER itself. This measurement is made by 4 photodiodes which collect the light at different levels. These diodes are contained in a thermalised box, and their absolute stability is regularly measured using an alpha source of $^{241}\mathrm{Am}$.

The study of the PMTs linearity imposes to cover the full dynamic range of the detector (from 100 MeV to 1.5 TeV by PMT). To achieve this, the LASER box includes a filter wheel which allows, depending on its position, to emit more or less light out of the system.

The light coming out of the LASER box is then distributed to the 9852 PMTs in two stages. A first distribution is performed at the output of the box trough a first element distributing the light to about 400 optical fibres. The amount of light distributed by these fibres can be equalised by means of specific connectors, in order to uniformise the system's response. The light is then transported in each module of the TileCal (100 m further), where it is finally distributed to a bunch of fibres directly connected to the PMTs.

1.4.3.2/ ELECTRONICS

The electronics of the LASER system, fully developed by the LPC electronics department, allows to control the various elements of the LASER box. It also allows to control very precisely the emission of the pulses. As previously stated, the LASER system is able to send pulses while the LHC is running. The procedure, sketched in the figure 1.14, takes advantage of the dead time of $3\ \mu s$ occurring at the end of each cycle of the LHC (abort gap).

The system has been dimensioned to allow LASER pulses to arrive in the TileCal exactly in the middle of the LHC abort gap. It has been shown that a precision of about 15 ns on the arrival time could be obtained, which is more than enough.

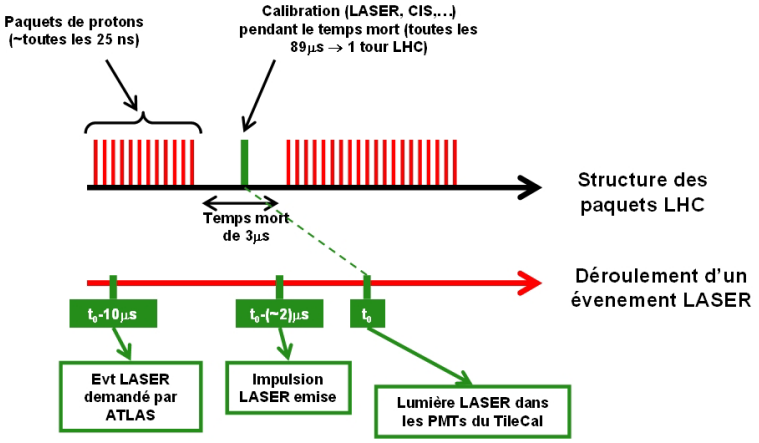


Figure 1.14: LASER event principle

1.4.4/ PERSONAL CONTRIBUTION

In charge of the ATLAS TileCal LASER system between 2008 and 2010, I followed and supervised the steps that allowed ATLAS to operate and integrate in the detector [16]. I then took part to the analysis of the commissioning results. Finally, I developed and validated the procedures that allow to use the LASER system for the monitoring of TileCal PMTs in situ.

In parallel, I have set specifications for the replacement of the initial system by a new calibration LASER more compatible with the system requirements: the LASER II. These specifications served as a basis for the development of this new version, which is now deployed in ATLAS [20]. All this work has been documented in detail in internal notes. These notes are reproduced in the Appendices B.1 and B.2 of this document.

2

THE HL-LHC PROJECT AND CMS STUBS

The HL-LHC project corresponds to a significant increase of the accelerator luminosity. The detectors will undergo profound changes to meet these new operating conditions. The improvement of the CMS tracker fits in this context. This is a very ambitious project that goes well beyond a simple refresh. The main innovation of this new detector, its introduction to the first level of the trigger system, is an unprecedented technical challenge for a system of this size.

2.1/ THE HL-LHC PROJECT

Since 2015, the LHC has been supplying data with record energy from 13 TeV proton-proton collisions. By 2025, more than 300 fb^{-1} of data will have been recorded with its current configuration. At this point, some detectors and magnets subjected to high doses of radiation will be at the end of their life. An adaptation program of the collider and detectors will then be necessary to go further and consider collecting another 10 times more data. It is the High Luminosity Project of the LHC (HL-LHC) [21], which aims at a collection of at least 3000 fb^{-1} over a period of 10 years from 2026. The different steps of this project are summarised in Figure 2.1.

The significant luminosity increase will have a major impact on the operation of future detectors. This is particularly true regarding the increase of the pile-up, ie the average number of interactions occurring at each collision. An event containing 5 to 10 times more secondary interactions is naturally more complex to analyse, but above all much more complex to select online. However, the selection of events must remain as effective as it is the LHC, otherwise the benefits of an increase in luminosity will be lost. It is therefore necessary to build detectors adapted to this new environment, both more precise and allowing a faster analysis of the information received.

We often talk about upgrade to qualify these new detectors. This term is misleading because it gives the impression of a simple improvement. As we will see in the following, brand new systems are developed under much less favourable conditions. Indeed, these

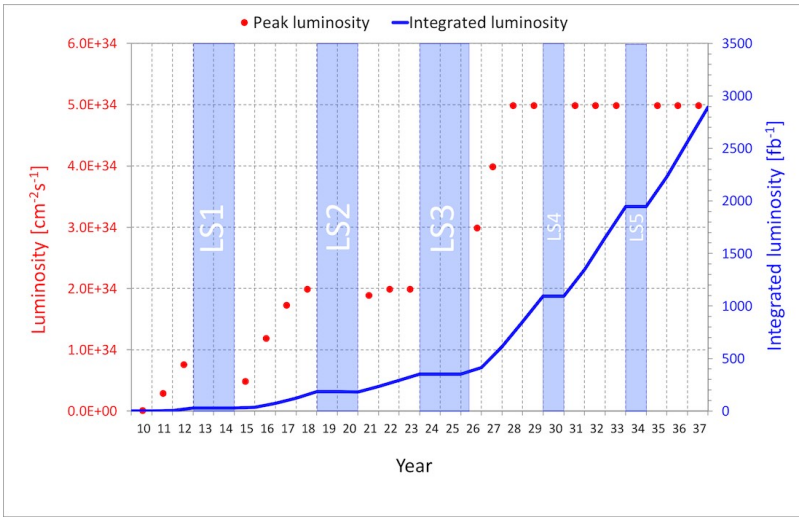


Figure 2.1: LHC and HL-LHC project phases

developments must be made while the current experiments are still in operation, with greatly reduced manpower. In the early 2000s, during the construction of the LHC, most of the technical services of the laboratories involved took part in this effort. The situation of the HL-LHC is very different, the project managers have to adapt to it on a daily basis.

2.2/ THE UPGRADED CMS TRACKER

As we have already said above, the upgrade of the LHC is accompanied by an increase in the complexity of the events. This results in an increase in the occupancy, which raises several problems: more complex reconstruction and less effective online selection.

The direct consequence of these two factors is a degradation in the performance of the detector, which is not really our ultimate goal...

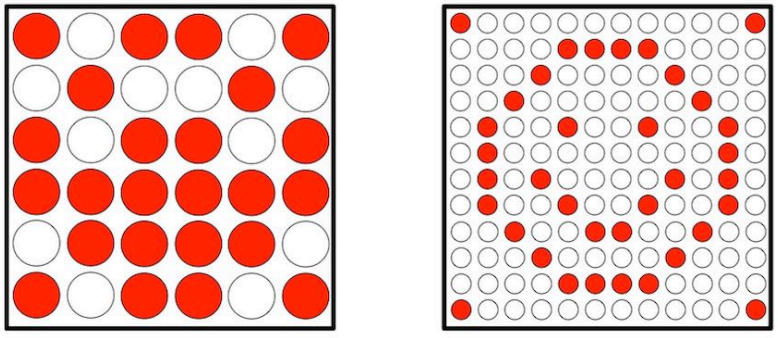


Figure 2.2: Occupancy vs granularity

There are 2 ways to reduce the occupancy in a detector: reduce the activity (less material), and increase the granularity.

The effect of an increase in granularity is illustrated in a simple way in Figure 2.2. With channels 4 times smaller, we divide the occupancy by 4 and improve the resolution. But

the number of readout channels is multiplied by 4, the extraction of information is getting more complex.

It becomes a problem when it comes to events triggering. If you want more accurate information faster, you must deeply change the trigger system. As we will see soon, this will set the most important constraint of the future tracker.

The triggering system of CMS will be very sensitive to the planned power enhancement. The challenge for the new system will be to maintain an excellent level of performance at the first level (read information at 40MHz, latency $12\mu s$), and even to improve it if possible.

So far, CMS L1 trigger is based only on outermost detectors with a much lower average occupancy than the tracker: the muon spectrometer and the calorimeters. In the context of the HL-LHC, the expected pileup rates will be such that it will become impossible to maintain such a strategy without including the tracker into it. The granularity of the external detectors will indeed overcome little or no change, and their occupancy rate will get slightly larger. It will become difficult to interpret their signals without using the tracker information.

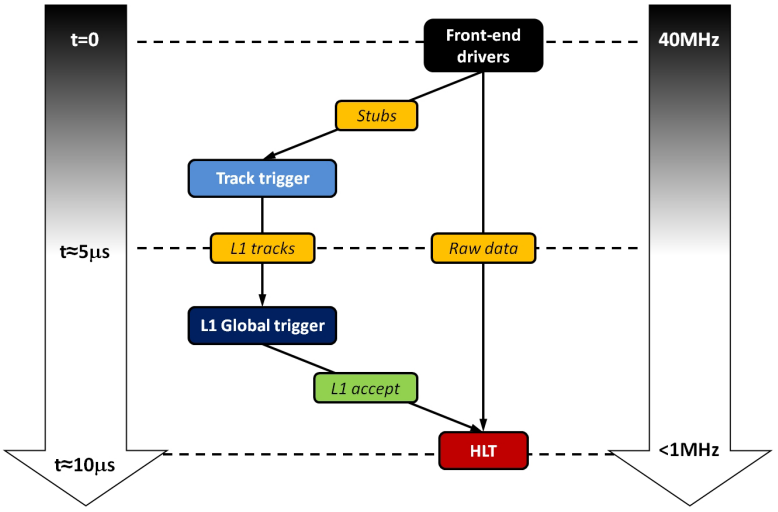


Figure 2.3: Data flow of the upgraded tracker

The use of the tracker at level 1 is an tedious problem, the principle of which is summarised in Figure 2.3. One must be able to extract the useful information of the tracker at 40 MHz, to reconstruct in real-time the tracks associated with an event (track trigger), and to exploit this information with those of the other sub-detectors (L1 global trigger), all that within the available latency that will be, at the HL-LHC, 12.5 microseconds. In parallel, the raw tracker data should be readout for each event passing level 1, at a frequency of up to 1 MHz . It is therefore necessary to extract 2 different data streams in parallel, but also to analyse one of these flows almost in real time (less than 5 microseconds). In the current CMS tracker, a single data stream is extracted at the frequency of 100 kHz. The new detector is a complete paradigm shift.

The Figure 2.4 is a 3 dimensional representation of the future tracker. This is an older version (known as flat geometry), but it gives a clear idea of the overall structure of the system. There are two regions that cover the entire phase space: the central barrel and endcaps. The barrel is composed of 6 layers, and each endcap of 5 disks. The 15000 detection modules are distributed on these supports in order to ensure an optimal cover-

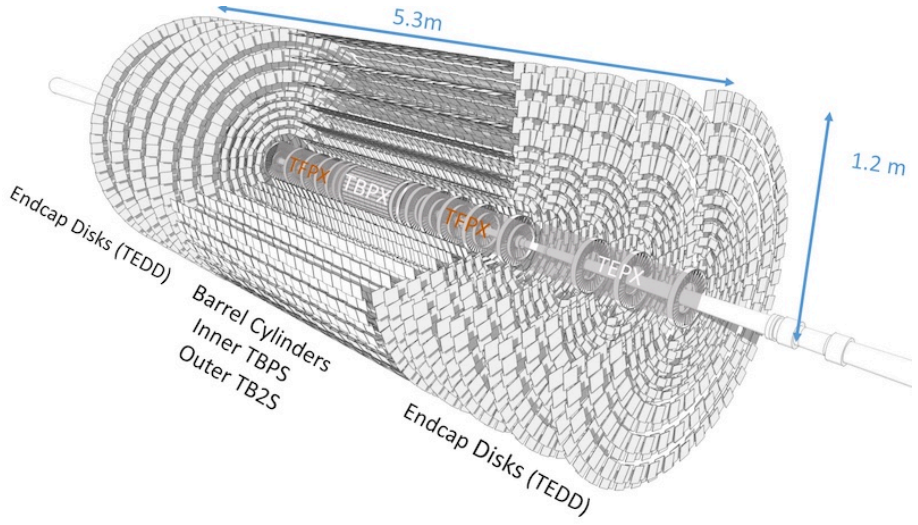


Figure 2.4: CMS phase II tracker

age. In order to reduce the quantity of material to be crossed, the number of layers is less than in the current detector. Nevertheless, a charged particle crossing the future tracker will leave a sufficient number of measurement points for the track trigger (5 to 6 on average). The complete structure of the detector has been completely redesigned: supports, services, modules, ... A general slimming cure has been carried out. On average, the amount of material crossed was divided by 2 compared to the current detector.

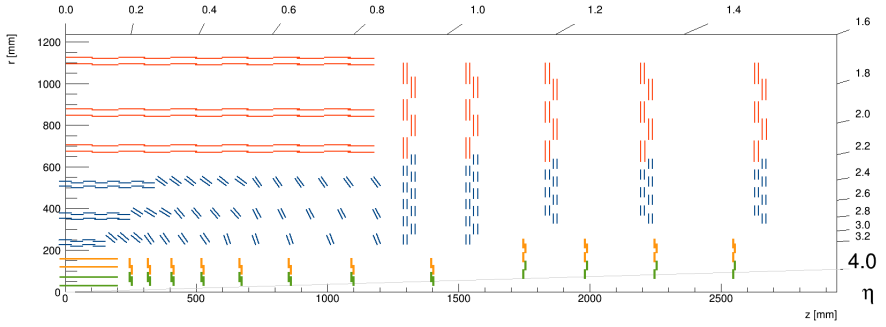


Figure 2.5: Tracker module repartition in the R-Z plane

The detection modules are the fundamental elements of the new readout strategy. The Figure 2.5 provides a more detailed view of the distribution of these modules in the R-Z plane. The red and blue lines indicate the positions of the 2 types of modules that will equip the detector. In order to maintain a reasonable occupancy rate in these internal areas, the blue modules, called PS (Figure 2.7), will have a greater granularity than the red, named 2S (Figure 2.6). The yellow and green modules, meanwhile, correspond to a more internal detector consisting of silicon pixels. For the first level of the future trigger system, only the 2S and PS modules will be used.

Mechanically, these modules consist of 2 rectangular and adjacent silicon layers (the distance between the 2 layers is a few mm), shown in yellow in Figures 2.7 and 2.6 .

For 2S modules, the 2 layers are identical and each one contains about 1000 strips of 90μm wide by 10 cm long. For PS modules, a layer contains strips 1.5 mm long instead of 5cm, hence the greater granularity of these modules. This new configuration explains the

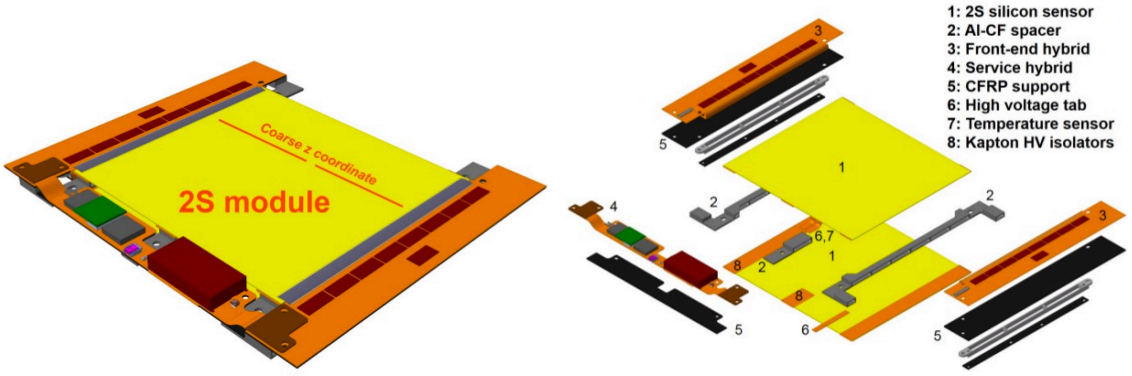


Figure 2.6: 2S module views

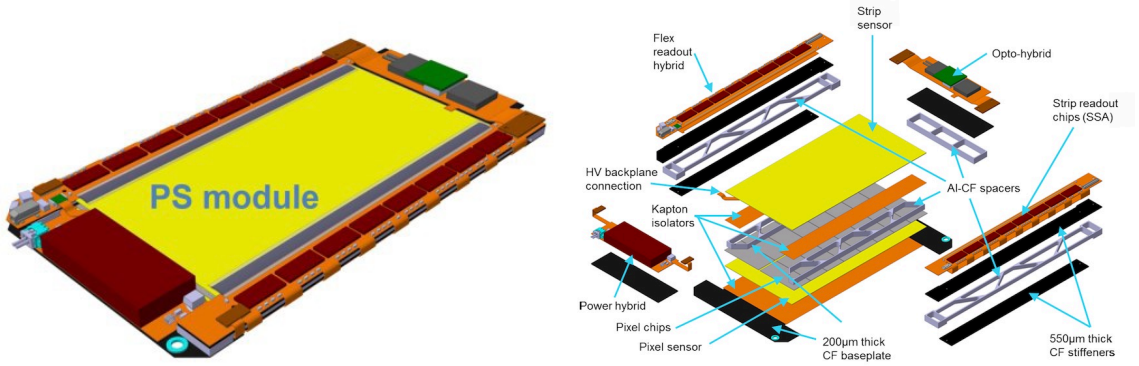


Figure 2.7: PS module views

fact that in the end, the future tracker will contain about 10 times more detection channels than the previous one.

The data flow coming out of the tracker will therefore be slightly greater than in the current detector. At 40MHz, the total signal amount will represent a mass of data of 10^4 Tbps !!! It would be totally unrealistic, and pointless, to try to extract such a large amount of information. Therefore filtering is done on every module. The signals of the two silicon layers are put into coincidence in order to form super-clusters called stubs. The basic principle of a stub is shown in Figure 2.8.

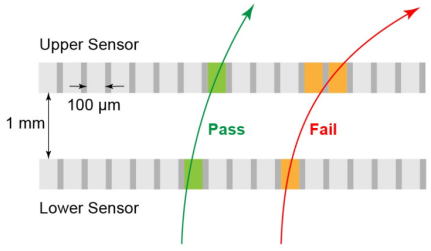


Figure 2.8: Stub: basic description

The use of stubs must divide the total amount of data to be extracted by about 200, which leads to an output data rate of 50Tbps. It is still relatively important, but technically it becomes acceptable. On the other hand, this simple concept requires a very sophisticated

readout electronics, but it is the price to pay to allow the track reconstruction system to have all the necessary information at level 1. We will come back in more detail on the extraction of information in the second part of this document.

Once the problem of the extraction of the data solved, remains that of their treatment. This is the second challenge of this future system, which will be devoted to the third part: how to deal with a flow of 50 Tbps in a few microseconds?

III

EXTRACTING DATA

3

TRACKER FRONTEND

This chapter starts with a detailed description of the frontend electronics of the future tracker. It is thanks to this complex system that the stubs, which are the track trigger fundamental objects, are reconstructed. The second part is devoted to these objects as they will become central in the rest of this document.

3.1/ FRONTEND CHAIN DESCRIPTION

3.1.1/ PRINCIPLE

Modules have been described in the previous section; we will now focus on the element that allows us to reconstruct and shape the signal: the hybrid. Hybrids are printed circuits that connect active areas (silicon strips) to reconstruction systems (ASICs). Their operating principle, common to the 2 types of modules, is shown schematically in Figure 3.1.

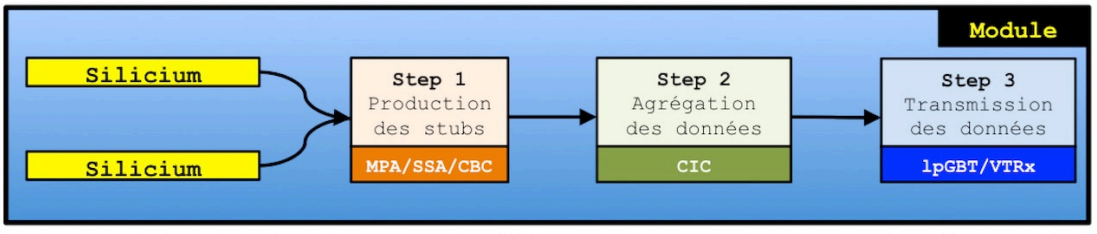


Figure 3.1: Readout: basic principle

The processing and the extraction of the signal are done in three distinct stages. First, the signal collected on the two silicon layers is simultaneously transmitted to a series of dedicated integrated circuits (ASICs) which amplify it, digitise it, and form the stubs. These ASICs are CBC [22] in 2S modules, and MPA [23] and SSA [24] in PS modules. The digitised signals are then transmitted to another ASIC which collects the information and formats it: the concentrator (CIC) [25]. The CIC is also located on the frontend hybrid.

The third stage, on the other hand, takes place on other hybrids, also linked to the modules: service hybrids. There is one such hybrid per 2S module and 2 per PS module.

The role of these components is to ensure the communication between the module and the outside world. It is through them that the power supply of the module arrives, and it is from there that the signal is extracted from the detector after being packaged in a frame to ensure that the data will not be corrupted (the data are valuable and must travel to the counting room without any problems, wrapped up in a lot of bubble wrap ...). This packaging is performed by the lpGBT [26], which is also an ASIC developed especially to meet the constraints of the HL-LHC project. Finally, the carefully packaged electrical signal is transmitted to the counting room via an optical fibre. The optical conversion is also carried out by a special connector developed at CERN: the VTRx [27].

3.1.2/ 2S MODULE READOUT CHAIN

The readout scheme of a 2S module is shown in Figure 3.2. The 2 layers of 2032 silicon strips are separated by 1.8 or 4mm. The gap width depends on the distance of the module from the interaction point. The closer the module is, the greater its thickness will have to be in order to keep an acceptable stub quality.

The strips have a width of 90 microns, and a length of 10 cm divided into 2 independent sections of 5 cm. These two segments define the 2 readout regions of a module, which are read by 2 distinct hybrids (left and right).

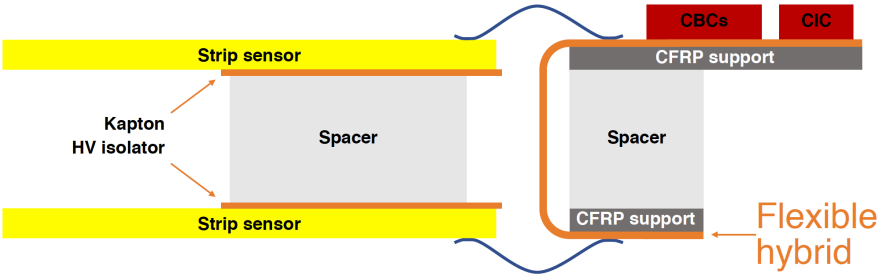


Figure 3.2: Detailed readout principle of a 2S module

Figure 3.2 shows for example the hybrid located on the right side. The strips are connected to it by 2032 wire-bonds. The hybrid being flexible, the signal of the two faces can be grouped and transmitted to a single component. In practice, this task is shared among 8 CBCs that collect the signal of 127 consecutive strips of each side (254 channels in total). CBCs digitise this signal and generate the stubs. In parallel, they transmit the entire raw data for each event passing level 1 of the trigger system.

The signals from the 8 CBCs are routed to a CIC which then formats and transmits the information to the service hybrid.

In total, each 2S module contains about 20 ASICs located on 3 distinct hybrids (left and right frontend hybrids are different). Most of the difficulties, however, lie in the development of frontend hybrids. The realisation of a flexible hybrid with such a complex routing and hosting so many ASICs is a real technological challenge.

3.1.3/ PS MODULE READOUT CHAIN

The readout scheme of the PS module is shown in Figure 3.3. Again the system is divided into 2 sections (left and right) and the signal is collected and transmitted by a CIC/lpGBT/VTRx chain. The main difference is in the granularity of the detector. The bottom silicon layer is made of macro-pixels of 1.5 mm length over 90 microns wide. The total length of the sensor being 5 cm, it is divided into 2 sides of 16 segments each.

As for the 2S module, the signal is collected by a flexible hybrid, but its operation is radically different. Indeed, the macro-pixels are directly connected to each MPA via a network of 1920 balls (one for each channel). All 16 MPAs and the silicon sensor interconnected by 30720 contacts thus form a single object: the MAPSA.

The signal of the strips of the opposite face is reconstructed by another ASIC, the SSA. The SSA information is transmitted via the flexible hybrid to the MPA, which then combines the signals on both sides of the module, and builds the stubs, just like the CBC. As in the case of 2S modules, each MPA/SSA pair covers only a part of a module: 120 strips in the case of PS modules.

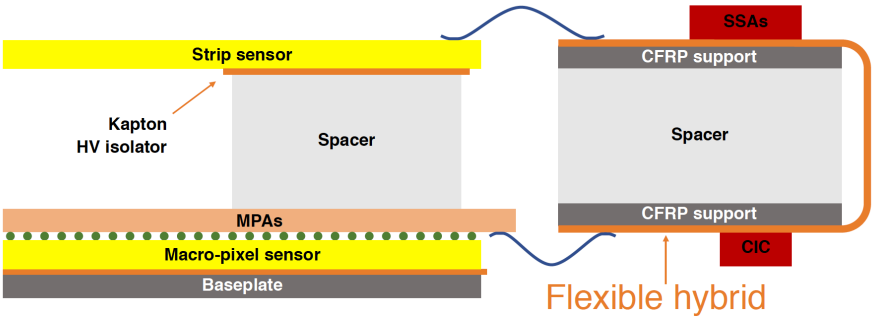


Figure 3.3: Detailed readout principle of a PS module

As for the 2S, the thickness of the PS modules varies between 1.6 and 4 mm, with this time an intermediate value of 2.6 mm.

The extraction of the information is done via a service hybrid relatively different from the one of the 2S module. Indeed, it contains only the elements related to the transmission of information from and to the outside. In contrast to the 2S modules, the power supply of the PS module is managed by a dedicated hybrid located on the other side.

Eventually, PS modules are slightly more complex than 2S modules. MAPSA and frontend hybrid routing are the main challenges to overcome. The multiple ASICs present on the module are also difficult projects to carry out, we will see it in the chapter devoted to the concentrator.

For the PS module as for the 2S, to realise a first functional prototype of such ambitious systems will be already a major milestone.

3.2/ THE STUB: FUNDAMENTAL OBJECT OF THE FUTURE TRACKER

3.2.1/ GENERAL DESCRIPTION

If the PS and 2S modules are so ambitious, it is mainly because they have to provide stubs to the track trigger system. It is time to take a closer look at this fundamental object.

As can be seen in detail in Fig. 3.4, a stub is reconstructed whenever two correlated clusters from an interesting track are detected.

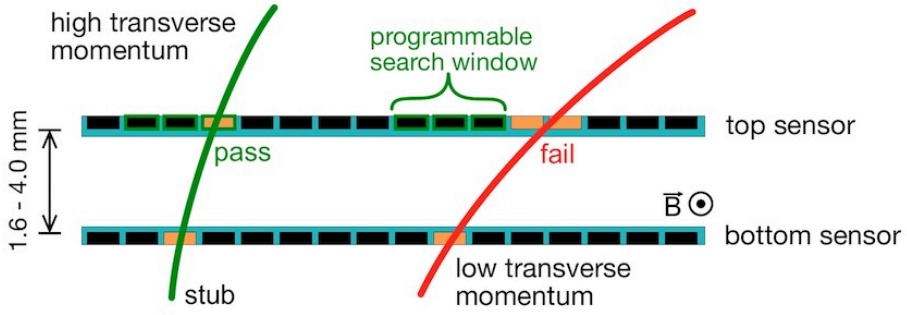


Figure 3.4: Stub: main principle

It is easy to understand the role of stubs in reducing the data throughput. Electronic noise, which can induce a false cluster, is eliminated by construction by this mechanism. Moreover, the width of the stub, which is the distance between the centres of the 2 clusters, is directly related to the transverse momentum (p_T) of the incident particle (if it comes from the interaction point). It is therefore possible, by modulating the width of the detection window used during their reconstruction, to select only the stubs coming from high-p_T tracks. Such cut significantly reduces the impact of pileup, which mainly induces low p_T tracks, on the amount of data to extract.

We will see later in this document that a deep understanding of the stubs can provide extremely useful information. I was personally in charge of their implementation in the official CMS simulation framework between 2014 and 2017. During this period, I was able to deepen my knowledge of these objects, and to develop methods to exploit their full potential. The conclusions of this work are summarised in a note written in 2018 [28] (note reproduced in Appendix C.1). I summarise below some elements of this note which seem to me important for a good understanding of the following chapters.

3.2.2/ RELATIONSHIP BETWEEN STUB WIDTH AND TRANSVERSE MOMENTUM

The width of a stub is fundamental for many reasons: for reducing the flow of data to extract, of course, but also, as we will see, for the reconstruction of tracks. This is indeed a parameter that can be used to check the consistency between several stubs of a track candidate. If we consider only primary particles, the geometric and simplified definition of stubs is described by Figure 3.5. The exact geometry, meanwhile, is shown in Figure 3.6.

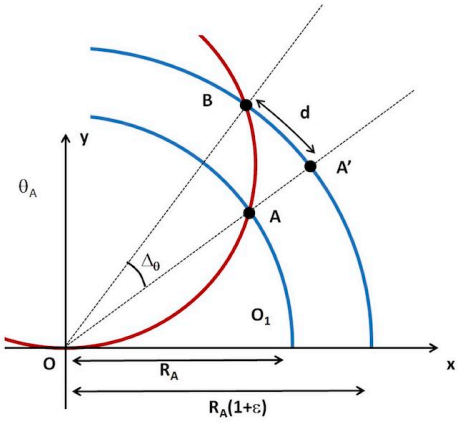


Figure 3.5: Stub definition

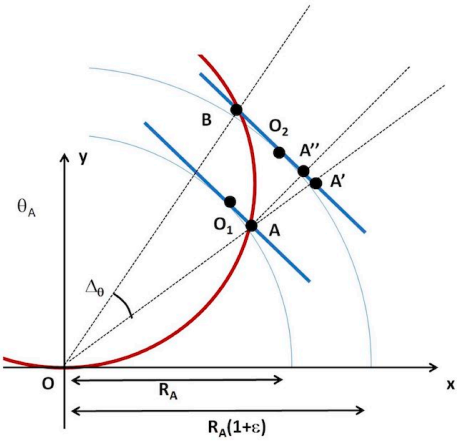


Figure 3.6: Correction applied to the stub width in order to account for the module planarity

Starting from Figure 3.5, one can establish the following relation:

$$p_T = 0.57 \cdot q \cdot R_A \sqrt{1 + \left(\frac{\epsilon'}{pitch \cdot bend}\right)^2} \tag{3.1}$$

where R_A, radius of the cluster closest to the interaction point, is given in meters, ϵ' and $pitch$ in mm, and $bend$ in number of strips. ϵ' is the radial distance between the 2 clusters forming the stub ($\epsilon' = \epsilon \cdot R_A$), $pitch$ the width of a strip and $bend$ the width of the stub.

Using this formula one can establish simple relations between the different parameters. The minimal width reconstructed by the frontend chips being 0.5 strip, one can for example easily estimate the maximum sensitivity of a module, ie the maximum transverse momentum to which it is technically sensitive. If we take for example a classic 2S module, for which we have ϵ'= 1.8 mm and pitch = 0.09 mm, we see that the sensitivity will be 5 GeV/c at 20 cm against 23 GeV/c at 1 m. The message to retain here is that the stub width will always provide a much more accurate information in the outer layers of the detector. It will always be easier to define selection windows in these areas than in the inner layers where the resolution will be less good and consequently the windows less precise. The use of thicker modules in the internal areas will tend to mitigate this phenomenon, but the resolution will always be better in the outer regions.

3.2.3/ PARALLAX CORRECTION

The flatness of the modules must be taken into account, especially in the inner layers where the effect can be relatively important (up to several strips). If we refer to Fig. 3.6, the distance actually measured by the frontend chips is the distance between the points B and A'', while the actual width of the stub is the distance between B and A'. The measure must therefore be corrected for the difference between A' and A'', which can be defined as:

$$A'A'' = \epsilon' \tan \widehat{AOO_1}, \tag{3.2}$$

This distance can easily be expressed in strips. We have indeed $N_{A'A''} = \frac{\epsilon'}{r_{O_1}} N_{AO_1}$. A quick numerical application shows that this value can go up to 5.5 strips for a signal on the edge of the modules of the innermost layers of the detector. This is far from negligible when we know that the size of the selection windows in this area will be at most 2 strips! This correction is therefore implemented directly in the frontend chips (CBC/MPA). 8 different corrections per chip can be pre-recorded (1 value for about 15 strips), which is more than enough to maintain an acceptable resolution. It is worth noting, however, that a more accurate value of the correction can always be calculated at the backend level, in order to improve the final stub width resolution.

In practice, it is possible to determine the exact value of the stub width and the parallax correction only in the case where the module is parallel to the beam axis (flat barrel region). In all the other cases, we can show that the parameter ϵ' will depend on the angle θ_0 between the track and the beam axis, as shown in Figure 3.7. The other parameters are δ, the distance between the 2 silicon layers, and α, the angle between the module and the beam axis. Following this convention, α is equal to $\pi/2$ and 0 for flat barrel and endcap modules respectively.

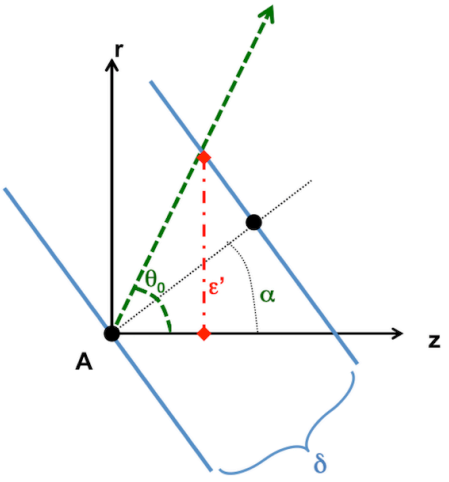


Figure 3.7: Parameters defining the stub in the case of a module tilted with respect to the beam axis.

With this new definition, one can generalise the relation between p_T and stub width:

$$p_T = 0.57 \cdot q \cdot R_A \sqrt{1 + \left(\frac{\sin\theta_0}{\cos(\theta_0 - \alpha)}\right)^2 \cdot \left(\frac{\delta}{pitch \cdot bend}\right)^2} \tag{3.3}$$

One can also generalise the parallax correction:

$$N_{A'A''} = \frac{\delta}{r_{O_1}} \cdot \frac{\sin\theta_0}{\cos(\theta_0 - \alpha)} \cdot N_{AO_1} \tag{3.4}$$

In both cases the term θ_0, which is unknown at the time of creation of the stub, appears. We can verify that this dependence disappears for $\theta_0 = \pi/2$, ie for a stub in the flat barrel region. For other types of modules, the resolution will necessarily be worse. This is what we see in Figures 3.8 and 3.9, where the bend resolutions for stubs from primary muons in the barrel and endcaps are compared respectively. There is indeed a poorer resolution in the endcaps, partly due to the point mentioned above.

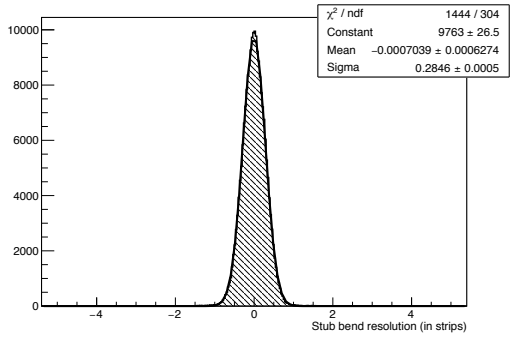


Figure 3.8: Bend resolution for stubs induced by primary muons in the barrel

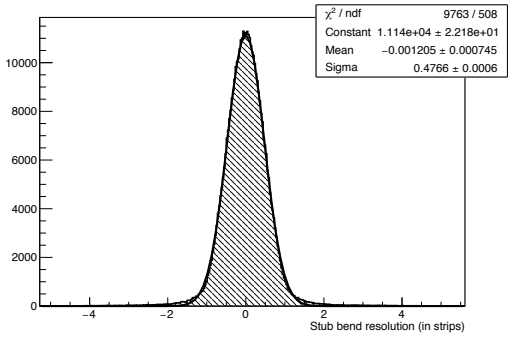


Figure 3.9: Bend resolution for stubs induced by primary muons in the endcaps

3.3/ STUB SELECTION

3.3.1/ TYPOLOGY

Whether they come from a secondary particle or a primary particle, the stubs associated with a particle can be classified into several categories which are summarised in Figure 3.10.

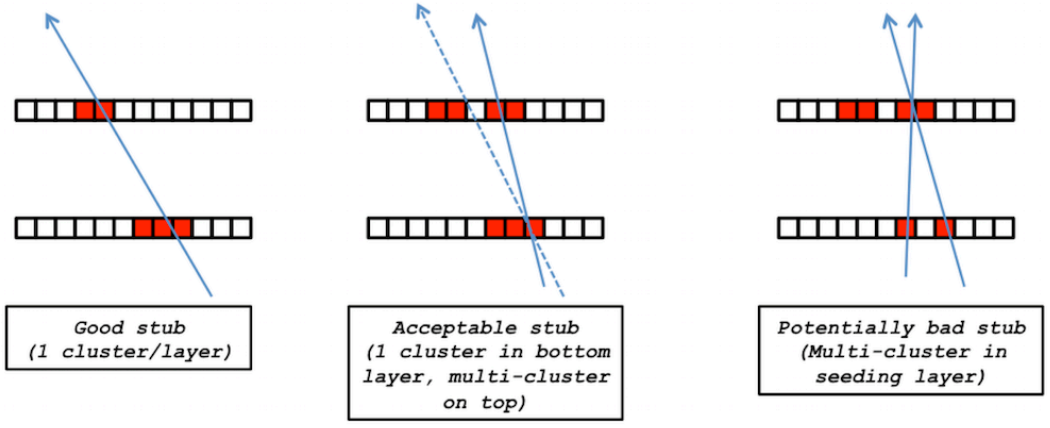


Figure 3.10: The different types of stubs associated with a primary particle

To fully understand the situations depicted in this figure, we need to look back at how stubs are built in CBCs and MPAs. In both cases the ASICs are based on a reference sensor located at the bottom of the module. For a PS module, for example, it is always the macro-pixels sensor. The modules can be placed in different ways in the detector, thus the reference sensor is not always the one that is closest to the point of interaction.

For each cluster on the reference sensor, it is not possible to produce more than one stub. On the other hand, nothing prevents a cluster of the upper layer from being used to form several stubs. Then, in the case where several clusters are found in the window of selection of the upper layer, one builds the stub with the smallest width (after correction of the parallax).

Starting from this observation we can define three types of stubs. It is important to specify

that we are dealing here with stubs associated with a primary particle (at least one of the clusters is associated). These three categories are:

- **Good quality**: exactly 1 cluster per sensor. The width of this stub is well correlated with the particle momentum (assuming this one is primary and in the absence of brehmsthralung). Figures 3.8 and 3.9 were obtained with stubs falling into this category.
- **Medium quality**: only one cluster is induced in the reference sensor, but there are several clusters at the top. This can happen if a strip is dead or if the signal is below the threshold of a strip. In this case, only one stub will be produced. Its position will be correct, but the width will be potentially biased. This stub is still good for a track reconstruction system using only the position information.
- **Poor quality**: there is more than one cluster in the reference sensor. In this case the position and the width are potentially biased. Such a stub, even if it is associated with a good particle, can potentially bias any track reconstruction algorithm.

.

In practice, all these stubs will pass the frontend selection stage. The typology presented here becomes important only in the perspective of track reconstruction. It is indeed tempting to use the width of the stubs to select track candidates and reduce the proportion of false tracks. However, before implementing such a strategy, it is important to understand the impact of the stub type on the width resolution.

3.3.2/ STUB SELECTION AT THE FRONTEND LEVEL

Before turning more specifically to the use of stub width for track reconstruction, let us return to the stub raison d'être: the reduction of the data flow to be extracted from the detector. To do this, the size of the selection window will first be constrained by the rate reduction to obtain, the efficiency coming in second. To define these windows, we must therefore begin by precisely determining the maximum amount of data that can be extracted from the detector.

The maximal average stub rates, per module and per collision, which can be extracted from the different module types, are summarised in the Table 3.1.

Module type	2S	PS (320MHz)	PS (640MHz)
Maximal average occupancy (stubs/module/BX)	4.8	4.4	8.8

Table 3.1: Frontend extraction limits, in average number of stubs per module and per collision.

These values are based on the data extraction capabilities of future modules. We will return to these figures in the following chapter, where we will see that these limits make it possible to ensure relatively low losses in the entire tracker, up to average pileup conditions of more than 250 collisions per event.

From there, there are two possibilities: either the expected occupancy is larger than this value, or it is below. In the vast majority of the cases, for an average pileup of 200, the

occupancy rate per module will be much lower. In this case, the selection window will be set to maximise the reconstruction efficiency down to the minimum selection threshold of 2 GeV/c. Otherwise, we will choose a window that allows to go below the maximum occupancy threshold, to the detriment of efficiency.

Obviously, the size of the windows will depend strongly on the type of particles used to calculate them. Windows determined with electrons will necessarily be larger than that obtained for muons. But wider windows also mean more background noise due to secondary stubs, hence higher occupancy rates. Finally, the occupancy rates depending on the average pileup, the windows will also rely on this parameter.

In order to take all these factors into account deterministically, a semi-automatic procedure for calculating selection windows has been developed [29]. The tracker is divided into 108 regions in which the windows are identical for each of the modules (Fig. 3.11).

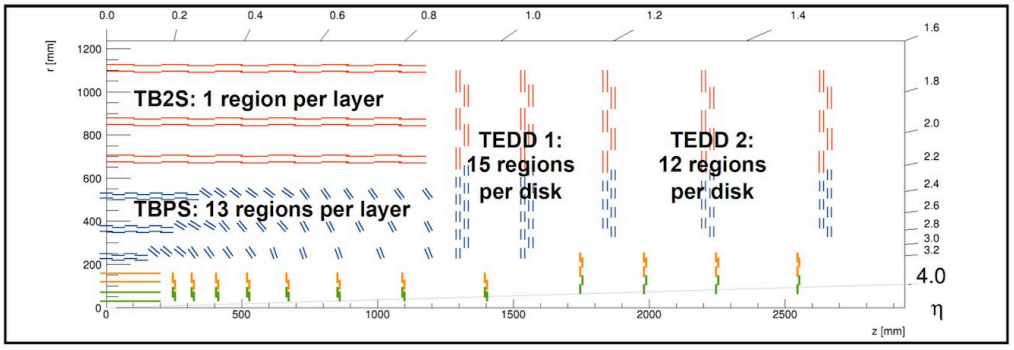


Figure 3.11: Tracker regions requiring a specific stud window tuning.

For each of these regions, and for a given average pileup, 2 set of windows are calculated: a strict window, determined with muons, and a wide window, determined with high p_T electrons and muons. For example, the wide windows obtained for the different regions for events with average pileup of 140 and 200 are given in Figures 3.12 and 3.13. It is clear that values remain unchanged in external regions where occupancy is low and where efficiency matters. On the other hand, in the inner regions, it is the average occupancy that dictates the window size, and the impact can be significant. We see for example that in the innermost ring of disk 2 (TEDD2/R1), the window goes from 1.5 to 0.5 strips !!! The impact on the efficiency of this area, luckily quite small, is far from negligible.

	FLAT	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	TEDD1_1	TEDD1_2	TEDD2_1	TEDD2_2	TEDD2_3	
TB2S3	7,0													7,0	7,0	7,0	7,0	7,0	R15
TB2S2	6,5													7,0	7,0	7,0	7,0	7,0	R14
TB2S1	6,0													6,5	6,5	7,0	7,0	7,0	R13
TBPS3	4,5	5,0	5,0	5,0	5,0	5,0	5,0	5,5	5,0	5,0	5,5	5,5	5,5	6,5	6,5	7,0	6,5	7,0	R12
TBPS2	3,0	4,0	4,0	4,0	4,0	4,0	4,0	4,5	5,0	5,0	5,0	5,0	5,0	6,5	6,5	6,5	6,5	6,5	R11
TBPS1	2,5	3,0	3,0	2,5	3,0	3,0	2,5	2,5	2,5	2,0	2,0	2,0	2,0	6,5	6,5	6,5	6,5	6,5	R10
														6,5	6,5	6,5	6,5	6,5	R9
														6,5	6,5	6,5	6,5	6,5	R8
														6,0	6,0	6,0	6,0	6,5	R7
														5,5	6,0	6,0	6,0	5,0	R6
														5,5	5,5	4,5	4,0	3,5	R5
														5,5	5,0	3,0	2,5	2,0	R4
														2,5	2,5				R3
														2,5	2,5				R2
														2,0	1,5				R1

Figure 3.12: Wide stub windows for an average pileup of 140.

To evaluate this impact, a complete comparison of the proportion of reconstructible tracks

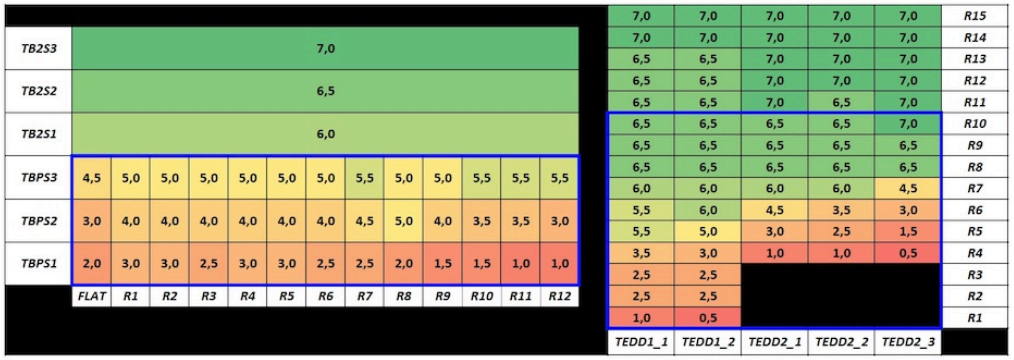


Figure 3.13: Wide stub windows for an average pileup of 200.

obtained using different sets of stub windows was done. We define as reconstructible any track that generates enough stubs in the tracker to be identified by the track trigger, ie at least 1 stub per layer on 4 distinct layers of the tracker, ideally 5. At the denominator we consider all the primary particles produced in the tracker's acceptance ($|\eta| < 2.4$), inducing at least 1 stub in it. The results obtained are summarised in Table 3.2.

	Average pileup of the stub window set	200		140	
Event type	p_T range (GeV)	$N_{hits} \geq 4$	$N_{hits} \geq 5$	$N_{hits} \geq 4$	$N_{hits} \geq 5$
Muon w/o PU	[2,8]	99.85 ± 0.01	98.45 ± 0.01	99.87 ± 0.01	98.52 ± 0.01
	[8,50]	99.87 ± 0.01	98.77 ± 0.01	99.89 ± 0.01	98.78 ± 0.01
Electron w/o PU	[2,8]	93.70 ± 0.01	89.09 ± 0.01	94.06 ± 0.01	89.66 ± 0.01
	[8,50]	98.52 ± 0.01	96.35 ± 0.01	98.58 ± 0.01	96.46 ± 0.01
Primary particles, PU200	[2,8]	92.40 ± 0.01	86.72 ± 0.01		
	[8,50]	95.2 ± 0.1	92.5 ± 0.1		
Primary particles, PU140	[2,8]			92.31 ± 0.01	88.04 ± 0.01
	[8,50]			97.0 ± 0.1	94.3 ± 0.1

Table 3.2: Proportions of reconstructible tracks measured with different types of wide stub windows sets.

Quite predictably, there is a decline in efficiency at PU200. This reduction remains relatively low tough, because the number of modules for which the occupancy rate is important stays relatively small.

For a given PU, we observe as expected a significant difference in efficiency between wide and tight tuning for electrons. The Table 3.3 compares the proportion of reconstructible electrons for wide and tight windows, optimised for average PU of 140.

	Stub window type	Wide	Tight
Electron w/o PU	[2,8]	94.06 ± 0.01	89.82 ± 0.01
	[8,50]	98.58 ± 0.01	97.77 ± 0.01

Table 3.3: Proportions of reconstructible electrons measured with wide and tight stub windows tunings define at PU140.

This difference is particularly important at low p_T where the energy losses are larger.

Ultimately, the choice of tuning will depend on the particles required by the track reconstruction system. The method presented here allows to choose between different scenario. If you are mainly interested in muons, you will go to the tight windows. On the other

hand, if it becomes important to also recover the electrons, we will be able to use the wide windows, the price to pay being a non negligible increase of the stubs multiplicity (30%). The stub windows can anyway be modified at anytime in the CBC and MPA.

3.3.3/ USING STUB BEND IN TRACK RECONSTRUCTION

3.3.3.1/ BEND RESOLUTION

Assuming that parallax correction can be calculated accurately (which is true for all modules parallel to the beam axis ($\alpha = \pi/2$), the main source of systematic error on the width is the position of the 2 clusters of the stub. As the information provided by the frontend is purely binary (a strip is on or off), the digital error on the position of a strip is given by the usual relation $pitch/\sqrt{12}$ [30]. From there we can infer the expected error on the width of the stubs in the case where it is made of two one-strip clusters: $pitch/\sqrt{6} \approx 0.41 \cdot pitch$. In practice we observe a better result, especially in the barrel (see Figure 3.8 showing the resolution obtained for good stubs coming from muons, in the case of barrel modules ($\alpha = \pi/2$)). This is due to the higher proportion of 2-strip clusters in this area.

Conversely, in the disks, single strip cluster are the majority, and the parallax correction can not be exactly calculated. We then find a value closer to the digital resolution. The observed resolution is even a little worse, as shown in Figure 3.9, showing the resolution obtained for good stubs from muons endcaps ($\alpha = 0$).

The analysis of these figures shows that, for good quality stubs, we can estimate a bend resolution of 0.5 strips as a reasonable approximation, which means that in 99% of cases, for this category of stubs, the difference between expected and measured value will be less than 1.5 strips. We will see in a later section how this information can be exploited.

However, this observation applies only to good quality stubs from particles whose trajectory is circular. For imperfect or electron-derived stubs, the resolution may be much worse, as can be seen in Fig. 3.14, obtained with low-quality stubs induced by muons, or in Fig. 3.15, obtained with good quality stubs from electrons. It will therefore be necessary to use the resolution of the stubs with caution.

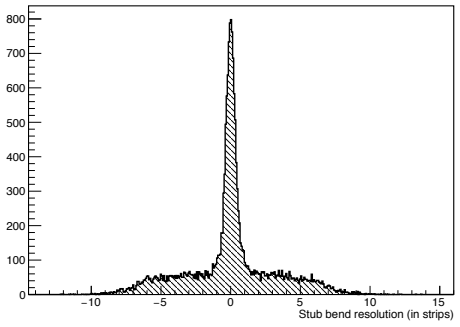


Figure 3.14: Resolution of poor quality stubs coming from primary muons in the barrel

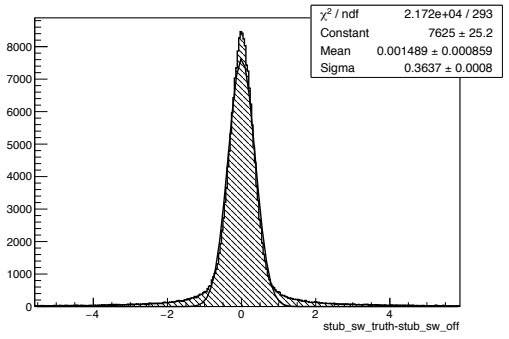


Figure 3.15: Resolution of good quality stubs coming from primary electrons in the barrel

3.3.3.2/ BEND-BASED STUB SELECTION

The figures 3.16 and 3.17 illustrate very clearly the influence of the stub type on the resolution. The two figures were made with stubs associated with primary muons, in the outermost layer of the barrel, where the lever arm is the most important. It is clear that correlation with momentum is much better in the case of good quality stubs.

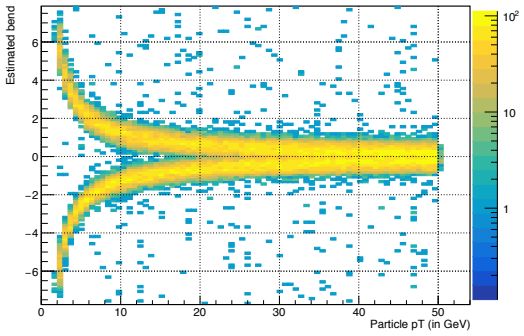


Figure 3.16: Width of good quality stubs induced by primary muons as a function of the transverse momentum of the muon.

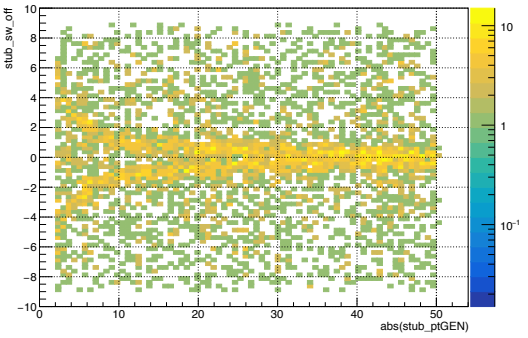


Figure 3.17: Width of poor quality stubs induced by primary muons as a function of the transverse momentum of the muon.

In the study we carried out, the proportion of poor quality stubs induced by a normal primary particle can be up to 5% in some areas. This is far from negligible, especially for stubs that are destined for the track reconstruction system. Again, all these stubs are not completely lost, especially for an algorithm using only the position.

To illustrate more precisely the risks involved in using the stub width, the proportion of reconstructible tracks with or without bend-based cut has been evaluated.

The cutoff on the width used is the following: the stub is rejected if the difference between the expected value and the measurement is greater than 1.5 strips. This type of very simple cut is exactly the kind of selection that could be applied effectively at the level of pattern recognition, where the expected p_T of the track seed can be pre-estimated.

In the table 3.4 the impact of this cut on the proportion of reconstructible tracks is measured for muons and electrons, without any pileup (the pileup effect is anyway irrelevant here).

	Bend cut applied	No		Yes	
Particle type	p_T range (GeV)	$N_{hits} \geq 4$	$N_{hits} \geq 5$	$N_{hits} \geq 4$	$N_{hits} \geq 5$
Muon	[2,8]	99.87 ± 0.01	98.52 ± 0.01	99.19 ± 0.01	95.09 ± 0.01
	[8,50]	99.89 ± 0.01	98.78 ± 0.01	99.88 ± 0.01	98.62 ± 0.01
Electron	[2,8]	94.06 ± 0.01	89.66 ± 0.01	81.50 ± 0.01	72.70 ± 0.01
	[8,50]	98.58 ± 0.01	96.46 ± 0.01	94.38 ± 0.01	90.64 ± 0.01

Table 3.4: Proportion of reconstructible tracks with or without bend-based selection.

As expected, it is in cases where the stub is more affected (electron, low p_T) that the impact of a cut is the most important. On the other hand, for muons of high transverse momentum, the loss is small and therefore the cutoff can become very effective.

It is however important to understand that this type of selection can have a significant

impact on the performance of the algorithm, and especially an impact that can vary over time because the proportion of stubs of poor quality is directly related to the degradation of the detector. Dead strips, in particular, could play an important role in the longer term.

In conclusion, the use of the stub width to make fine selections must be made with discernment. It should be remembered that initially the stubs were not introduced for this purpose. For the record, when we started to define the output data formats of the concentrator (see next chapter), it was not even planned to transmit the width of the stubs to the track reconstruction system!

4

DEVELOPMENT AND REALISATION OF THE CONCENTRATOR

This chapter is dedicated to the concentrator chip, a project for which I assumed the scientific responsibility from 2013. The role of this element of the acquisition chain of the future tracker is presented in a thorough way, as well as the long way leading from the requirements definition to the first version which was successfully tested in 2019.

4.1/ THE ROLE OF A DATA AGGREGATOR

4.1.1/ WHY AN ASIC?

The role of the concentrator was briefly mentioned in the previous chapter. It is partly thanks to this component, which groups and formats the data produced by MPAs and CBCs, that the stubs can be distributed to the fast track reconstruction system.

This chip is an ASIC (Application-Specific Integrated Circuit). Unlike the processors that equip our computers and other smartphones, ASICs are specially developed to perform a particular task.

Building these components requires significant technical expertise, and one can first wonder why we need them, while the tools available on the market, especially FPGAs, can perform a lot of very sophisticated tasks. In fact, if we look at the constraints that the components installed on the modules must fulfill, one quickly realises that an ASIC is the only alternative.

The first argument in favour of an ASIC is power. The available budget for the data concentrator is a few hundred milliwatts only. We are very far from the few tens of watts of a conventional high-end FPGA.

The other major constraint is radiation hardness. The tracker is close to the beam, it is therefore subjected to very large doses, and is permanently crossed by a multitude of charged particles that can distort the signal traveling through the modules. By inverting the value of a bit, this type of event (called SEU for Single Event Upset) can have a

devastating effect on data acquisition. The component must therefore be able to withstand this type of phenomenon. This constraint alone is enough to eliminate a large majority of FPGAs.

Even if these two arguments are insufficient, one could also mention the constraints related to the data processing latency, the form factor of the component, and so on. All the reasons why it is necessary to develop an ASIC.

4.1.2/ GENERAL DESCRIPTION

Throughout this chapter we will use several denominations for CIC: CIC1, CIC2, or simply CIC. CIC1 and CIC2 are the two physical versions of CIC completed to date. They perform the same data processing, and differ only in certain technical aspects which we will come back to later. Regarding the general behaviour of the ASIC, CIC1 and CIC2 are equivalent.

The very simplified operating principle of the CIC is shown in Figure 4.1. As mentioned in the previous chapter, the circuit receives the stubs, but also raw data, provided by 8 CBCs or MPAs. It reformats this data, processes it, and groups it into packets before sending it to the lpGBT.

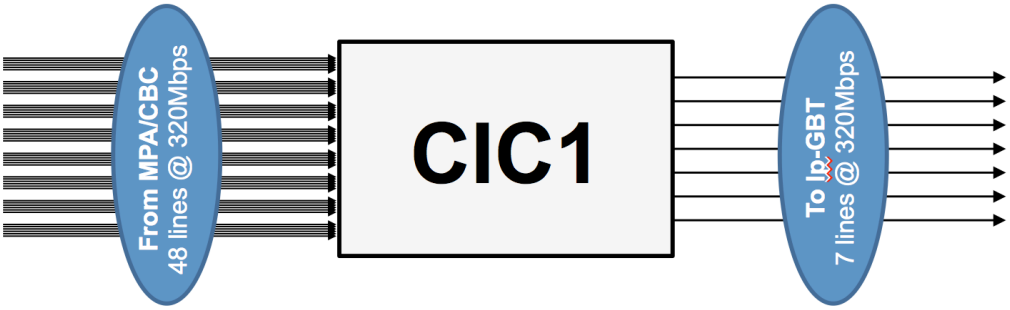


Figure 4.1: CIC basic concept

The CIC must therefore be compatible with 2 different types of modules. That's what makes it so complex. An ASIC is by definition specific, asking him to be flexible is necessarily unnatural ...

In this case, the reason why there is not a different aggregator for each modules flavour is historical. Originally, it was not foreseen to have this feature on the PS modules.

Such a modification of the specifications should not have been possible if the project had been clearly defined from the start. Indeed, the normal process of developing an ASIC can be summed up in the following steps:

- 1. Establishment of detailed and precise specifications
- 2. Development of a numerical model and behavioural simulation
- 3. Physical design, validation, and simulation of routed design

Respecting this approach is extremely important. Indeed, once the ASIC is produced, there is no way to modify it. It is therefore essential to simulate in detail its operation

before sending it to the foundry. In the same extent, the specifications must be defined in the most precise way possible from the beginning of the project. A simple modification on the paper may prove impossible to put in place on a too advanced design.

If you do not respect the rules of this game, you are inevitably subject to delays and other complex situations to manage, the CIC is a good example. The following parts, which describe the project in detail, do not necessarily follow a chronological order, but they reflect the order in which I would carry out this project if I was starting it today.

4.2/ REQUIREMENTS

4.2.1/ CONTEXT

The definition of the specifications is a fundamental step. As soon as this milestone is reached, a team of experienced developers can make a first version of an ASIC in about 18 months. Conversely, if the development of ASIC begins before the completion of this step, the situation can quickly get stuck.

The specifications are defined first and foremost by the users of the component, that is to say the physicists. At a minimum, they must provide engineers with the input and output data formats of the chip, but also the speed at which these data will enter and quit the ASIC. With this information, it is possible to evaluate the technical feasibility of the project and to converge, with some iterations, towards a realistic solution. From there, the role of the physicist is normally limited to monitoring the progress of the project.

The requirements are also defined by the technical specifications of the host module. Indeed, its form factor and pinout will be strongly constrained by the hybrid to which it will be connected. Its supply voltage and its consumption will also have to follow very precise rules. The technology used, and in particular the process thickness, is also a very important constraint for the project. These elements, if they do not prevent the development of the CIC model, must be known as soon as possible in order to be able to approach the physical design of the chip in good conditions.

All these points are obvious if you have already designed ASICs. Otherwise, it is not so simple. It is undoubtedly this lack of expertise that jeopardised this project when it started at the end of 2011. The CIC was introduced when it became clear that the data from the CBCs had to be compressed to be able to exit 2S modules. This was more or less the only, and rather vague, specification for few years.

Little by little, things got more precise. The inclusion of CIC in the PS module, for example, dates back to the end of 2013, almost two years after the start of the project. The definition of data formats and output speeds between 2014 and 2015 really marked the start of the model development, but the physical design really started only in 2017. The specifications as presented in this document have not been finalised until the beginning of 2018. No wonder, in this context, that the first version of the CIC was only available in 2019, when it was originally planned for 2015.

The message to remember here is the following: without data format, no model, and without complete specifications, no ASIC. In retrospect, I am convinced that knowledge of data formats is the absolute minimum for accepting to enter into this type of project.

4.2.2/ CIC WORKING PRINCIPLE

The inputs/outputs of a data processing ASIC are the first information to be defined. This information, in the case of CIC, is summarised in Figure 4.2. They are of two types: information related to the data in transit in the chip, and those related to its control.

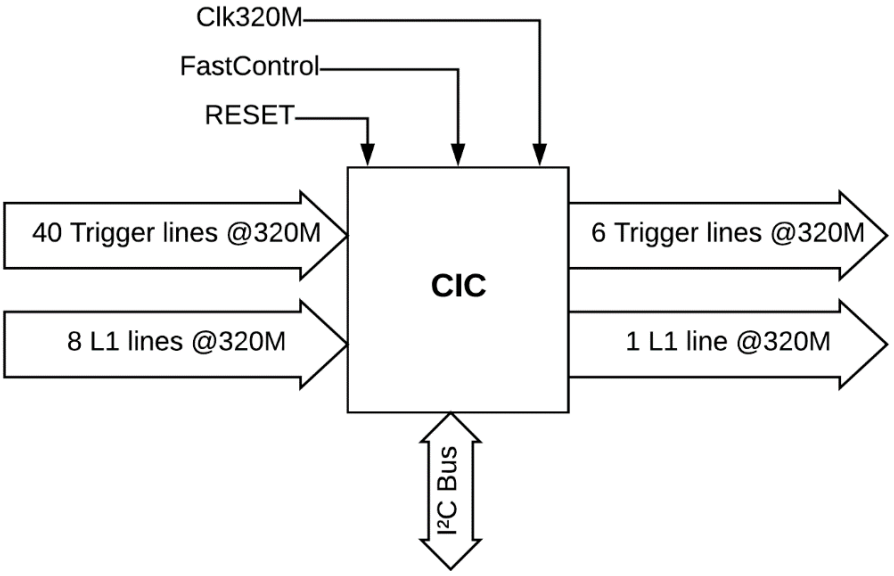


Figure 4.2: CIC I/Os

As we have already said several times, CBCs and MPAs provide two independent data streams:

- **Trigger path**: data generated synchronously at the frequency of 40MHz. These are the stubs. These must be transmitted as quickly as possible to the track reconstruction system located outside the detector. Here we can lose data (we'll see which), but we cannot waste time.
- **Data path**: raw data, which is retrieved on demand for each event passing the first level of the trigger system. This information, by definition asynchronous, is transmitted at an average frequency up to 750kHz. Here we can waste time, but we must not lose data.

All this information arrives at the CIC via differential lines clocked at 320MHz. Each CBC/MPA has 6 output lines: 5 for the trigger channel and 1 for the data channel. In all, therefore, each CIC has 48 entries: 40 for the trigger, and 8 for the data. This repartition is fixed and definitive.

At the output, the initial specifications were of 6 lines at 320MHz (5 trigger and 1 data), but a detailed study of the dynamic losses of the CIC (see next section) has lead to change this for PS modules. For the latter, the CIC will have 7 output lines (6 trigger and 1 data) that can operate not only at 320MHz, but also at 640MHz for some PS modules very close to the interaction point.

The control of the ASIC is carried out by the following lines:

- **Clock**: the clock is what allows the CIC to be synchronised with the LHC, and generate the set of internal clocks of the ASIC. In both cases, this clock is provided by the lpGBT, at the maximum frequency required by the CIC, ie 320MHz in the 2S modules and 640MHz in the PS modules.
- **Fast control**: those signals correspond to an 8-bit frame sent synchronously to the CIC at a frequency of 40MHz. This 8-bit combination provides the CIC with information on the actions to be performed immediately: resynchronisation, reception of an L1A, ... These control signals are sent during data acquisition.
- **Slow control**: Slow control, on the other hand, corresponds to the set of signals used to initialise the ASIC or to change its operating mode between 2 data taking phases. A set of configuration parameters can be modified via the slow control block. For example, slow control can be used to choose whether the CIC should operate in PS mode or in 2S mode. Each parameter is contained in a register accessible for reading and writing. Some registers are only readable (eg error counters).
- **Reset**: this line is used to trigger a complete restart of the ASIC, at any time. This is called a hard reset, as opposed to the resynchronisation which is triggered in a synchronous way by the fast control and during which only certain parameters are erased (the configuration parameters, for example, are retained).

4.2.3/ I/O DATA FORMATS DEFINITION, STUDY OF THE DYNAMIC LOSSES.

The definition of the inputs and outputs is closely related to the data flows that pass through the concentrator. These flows depend on the occupancy rate of the detector, but also the size of the data frames, and therefore their format. As explained above, this work is done upstream by the physicist. This was logically my first contribution when I took over the project at the end of 2013.

The information described in this section is taken from the document summarising the specifications of CIC1 [31] C.3, as well as a series of notes devoted to the study of dynamic losses in the concentrator [32], the most recent version of which is reproduced in the Appendix C.4.

4.2.3.1/ INPUT FORMATS

What goes into the CIC is what comes out of the MPA or the CBC. As the CIC was developed later than these ASICs, the incoming data format was already largely frozen in 2013. This is both a good and a bad thing. Good because we have immediately the information we need, bad because we know it will be very difficult to change these formats (we do not change the specifications). In a frontend chain containing multiple ASICs, it is important to develop the components in close collaboration. If an ASIC is too early, like the CBC, it will impose its operation on others. If an ASIC is too late, like the CIC, it will serve as a trashcan to others. The chain must be thought in its entirety from the start.

For the 2S modules, and therefore the CBC, the data of the trigger path (stubs) is formatted on a 40-bits block that contains all the stubs reconstructed by the corresponding CBC during a collision (BX). Each block can contain a maximum of 3 stubs, the maximum number of stubs that can reach 1 CIC for a given BX is therefore 24. For the data path,

the raw information is transmitted. This means that the CIC receives from each CBC a fixed size frame containing the signal of all 254 strips. Even channels without signal are transmitted. This is what we call non-sparsified data, or data without zero suppression.

For PS modules, the MPA formats the stubs on an 80-bits frame containing the information of 2 collisions. This grouping makes it possible to transmit up to 5 stubs per MPA for 2 BXs, and thus makes it possible to absorb an event for which the number of stubs would be larger than the average. It can be said that the MPA plays the role of pre-concentrator, on 2BX instead of 8. As for the raw data, they are transmitted to the CIC sparsified. This means that only the activated strips are transmitted, in the form of clusters. There may indeed be several consecutive strips activated. The fact of transmitting sparsified data makes it possible to significantly reduce, in the vast majority of cases, the size of the frame to be transmitted. The only drawback of this method is that the size of the frame sent by the MPA is not fixed and depends on the number of clusters. This requires special treatment at the CIC level which must gather information that reaches it in a totally asynchronous way.

4.2.3.2/ OUTPUT FORMATS

For the trigger path, CIC groups information over time and space. An output packet containing the 8CBC/MPA stubs recorded for 8 collisions is prepared and sent every 200 ns to the lpGBT. The length of this block is fixed and depends on the data output frequency: 384 bits at 320MHz and 768 bits at 640MHz. If the number of stubs to be transmitted is less than the storage capacity, the frame is completed by 0 (padding bits). In the opposite case, one selects in priority the stubs with small width (we will return later on the way this selection is operated).

For the data path, in the 2S case, the CIC forms the clusters from the frames from each of the 8 CBCs, and groups them into a single frame containing all the half-module clusters for the corresponding event. For PS modules, because clusters are already formed, the CIC only performs grouping, but the recovery of input frames, which vary in size, is more complex.

In both cases, the format of the output frames is the same for the two types of modules (modulo the encoding of clusters and stubs, the granularity of the modules being different). This standardisation is important because it simplifies data processing at the next level. For the trigger channel, where latency is critical, it is essential.

4.2.3.3/ DYNAMIC INEFFICIENCIES

CIC trigger path has 40 inputs at 320MHz for 6 outputs at 320/640MHz. The compression ratio to be achieved is therefore at least 6.7 or 3.35. For the data path, the compression ratio is 8 or 4. One must therefore ensure that the average occupancy of the detectors will be sufficiently low to reach this ratio everywhere.

This is especially true for the trigger path where no sparsification is performed. The reduction factor will thus only depend on the average occupancy in the modules. If it is too high, the CIC will receive too many stubs and it will not be able to transmit all of them. It is therefore important, in order to dimension correctly the CIC output capabilities, to evaluate the potential losses, and to implement, if necessary, appropriate countermea-

sures. The sizing of the selection windows presented in the previous chapter is one of the possibilities.

This evaluation was carried out with samples of events simulated under the conditions expected at the HL-LHC, that is to say with an average pileup of up to 200, and even 250 in order to account for a safety factor. In order to accurately estimate the losses expected in the CIC for a given module, sequences of several thousand events must be generated. Since the CMS framework is not suitable for this kind of study, a standalone application has been developed.

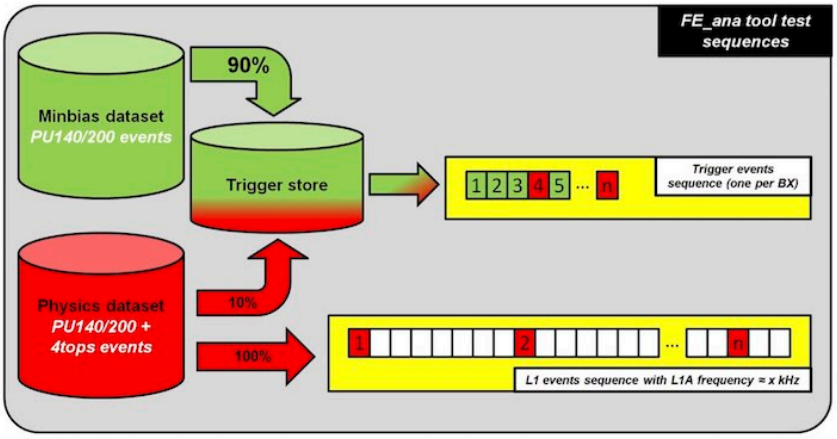


Figure 4.3: FE_ana code working principle

The operating principle of the program set up to study losses (named FE_ana) is described in Figure 4.3. Two datasets are created from events simulated in the tracker: a sample containing only minimum bias events, and another with events that may pass first trigger level (which have an average occupancy 10/20% larger than minbias). For each dataset, one stores all the information related to the frontend chain: stubs, strips activated, ...

From these packets containing few thousands of events, random data sequences are created to test the two CIC channels. For the trigger path, information from the 2 sets is mixed in a predefined proportion, whereas for the data path only events of the appropriate packet are selected. Concerning the size of the sequence, one generates approximately 10000 BXs for the trigger channel (1 event for each BX), and 100000 BXs for the data path (L1 events being randomly distributed over time so as to obtain a final frequency of approximately 750kHz). Once the sequence is created, the losses can be calculated in a relatively simple way.

This is especially true for the trigger path. Indeed, the size of the data blocks is predetermined in this case. For example, with 5 lines at 320MHz, the output block has a fixed size of 320 bits, which makes it possible to transmit a maximum of 16 stubs for a 2S module, and 13 for a PS module.

We can then establish, for each of the areas defined in the previous chapter, a complete map of the losses in the tracker. Figure 4.4 shows the measured loss map for an average pileup of 250, using a 320MHz output speed and 5 output lines for all CICs in the tracker.

Losses are estimated only for the stubs induced by a primary particle with a transverse momentum greater than $2\ GeV/c$. One can indeed tolerate losses in the other categories

R15	0,00	0,00	0,00	0,00	0,00				0,00	0,00	0,00	0,00	0,00	R15
R14	0,00	0,00	0,00	0,00	0,00	L6	0,00	L6	0,00	0,00	0,00	0,00	0,00	R14
R13	0,00	0,00	0,00	0,00	0,00				0,00	0,00	0,00	0,00	0,00	R13
R12	0,00	0,00	0,00	0,00	0,00	L5	0,01	L5	0,00	0,00	0,00	0,00	0,00	R12
R11	0,00	0,00	0,00	0,00	0,00				0,00	0,00	0,00	0,00	0,00	R11
R10	0,02	0,03	0,02	0,01	0,01	L4	0,08	L4	0,01	0,01	0,02	0,03	0,02	R10
R9	0,06	0,02	0,02	0,02	0,02				0,02	0,02	0,02	0,02	0,06	R9
R8	0,09	0,09	0,05	0,05	0,03	L3	0,13	L3	0,03	0,05	0,05	0,09	0,09	R8
R7	0,37	0,09	0,10	0,04	0,03				0,03	0,04	0,10	0,09	0,37	R7
R6	0,34	0,59	0,17	0,18	0,07	L2	1,59	L2	0,07	0,18	0,17	0,59	0,34	R6
R5		1,12	1,88	0,81	0,83				0,83	0,81	1,88	1,12		R5
R4		6,15	8,40	4,85	2,33	L1	22,43	L1	2,33	4,85	8,40	6,15		R4
R3			12,59	15,21	9,44				9,44	15,21	12,59			R3
R2				22,23	16,13				16,13	22,23				R2
R1				37,93	35,23				35,23	37,93				R1
	D5-	D4-	D3-	D2-	D1-	BARREL			D1+	D2+	D3+	D4+	D5+	

Figure 4.4: Proportion of good stubs lost after the CIC chips (in %), for PU250 events sequences, using 320MHz output speed.

of stubs. To evaluate these numbers we use strict stub windows tuning based on an average pileup of 200, in order to maintain maximum efficiency everywhere.

The threshold below which the losses are considered to be acceptable is fixed at 1%. It is clear that in this configuration, which was originally the basic configuration for the entire detector, this threshold is passed in the most internal areas. These results showed that it was necessary to go to 640MHz output in the first 4 rings of the discs and in the first layer of the barrel. These are important changes for the CIC, but also for other elements of the detector such as hybrids. It clearly demonstrate the importance of these analysis. It is essential to carry it out as quickly as possible.

In areas where the threshold between 1 and 5%, the transmission rate stays at 320MHz, but an additional output line is added, as well as a different transmission protocol at the lpGBT level (lighter error correction), freeing up space for the output frame. In case of persistent problem, the size of the stub selection window provides a last lever arm. But, as we have said before, this cut is used as a last resort because it has a direct impact on the reconstruction efficiency.

Regarding the L1 data path, given the variable size of the frames and the intrinsic asynchronicity of the data packets, the losses are a little more complex to evaluate. To measure them, one evaluates the extraction time of the output frame. This latency is defined as the difference between the BX where the last frame bit leaves the concentrator and the one where the first bit enters the chip. Under normal operating conditions, the average value of the extraction time must be constant over time, even tough it can present significant variations (an empty frame will take much less time to process than a frame containing a large number of clusters). On the other hand, if the detector occupancy gets too important, the average size of the frames increases and they become more time consuming to extract. In this case, the risk of filling the concentrator increases dangerously. When the storage capacity is exhausted, the chip enters in overflow mode and begins to lose data. If this overflow is linked to a too high occupancy, the situation is not reversible and the concentrator must be resynchronised, until the next overflow. The study of the average extraction time is therefore very important because it makes it possible to prevent this type of problem, and provides valuable indications on the required output speed and on the storage size needed.

The Figure 4.5 shows the evolution of the number of events stored in a CIC, obtained

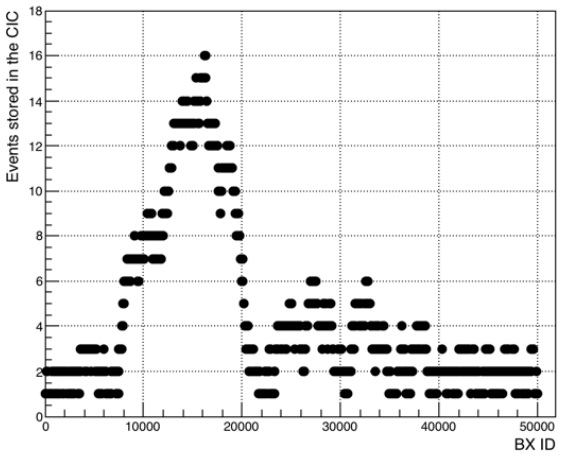


Figure 4.5: Number of L1 events stored in a CIC as a function of time.

with this simulation, in the case of an internal module where the size of events is large. One sees that there can be up to 16 events in parallel, but that the output speed in this case is large enough for the FIFO to empty. In other cases, we see the number of events increase indefinitely. It is to address those cases that the doubling of the output speed has also been necessary for the L1 path.

In the end, a depth of 16 was chosen for CIC's FIFO. The situation observed in Figure 4.5 is very rare and only concerns a very limited number of modules for which the output speed has not been doubled. We expect to observe some situations of this type, but in a very small proportion.

In conclusion, the study of the dynamic losses led to profound modifications of the CIC, but also of the PS detection module. This confirms that the definition of the data format and the study of the potential losses are essential elements for the specifications of the concentrator, and that they must be realised from the beginning of the project.

4.2.4/ TECHNICAL SPECIFICATIONS

The other elements of the specifications are related to the physical dimensioning of the component. Some of these constraints are related to the hybrid (pinout, voltage supply, form factor), others are related to the detector cooling constraints (engraving thickness). The main constraints are summarised in the Table 4.1. Other technical constraints, such as the form factor, arise from these initial conditions.

Here also, a late definition of these parameters can lead to complex situations. The fact that CIC core voltage is different for the 2 modules is a good illustration of this problem. An ASIC is not intended to operate at 2 different voltages, it is a source of problems. However, the CIC arriving last, we had to deal with the power available on each of the modules. The 1.2V being not available in sufficient quantities on the PS modules, it was necessary to go to 1V. Why in this case do not switch to 1V also on the 2S module? Simply because this voltage is not available at all on 2S modules...

Technology	65nm CMOS with 7 metal layers
Inputs	48 differential lines at 320Mbps
Outputs	7 differential lines at 320/640Mbps
I/Os voltage	1.2V
Core voltage	1V (PS) or 1.2V (2S)
Power budget (in mW)	250 (PS) or 310 (2S)
Connectivity	Bumped bare die (Flip chip)

Table 4.1: CIC technical specifications.

4.3/ CIC MODEL

4.3.1/ OVERALL DESCRIPTION

Once the data formats and I/Os rates are defined, it is possible to develop the model. Since the concentrator is a purely digital ASIC (apart few peripheral blocks that have not been developed by our group), its model is central. It conditions the feasibility of the physical component, and must be developed from the beginning with this constraint in mind.

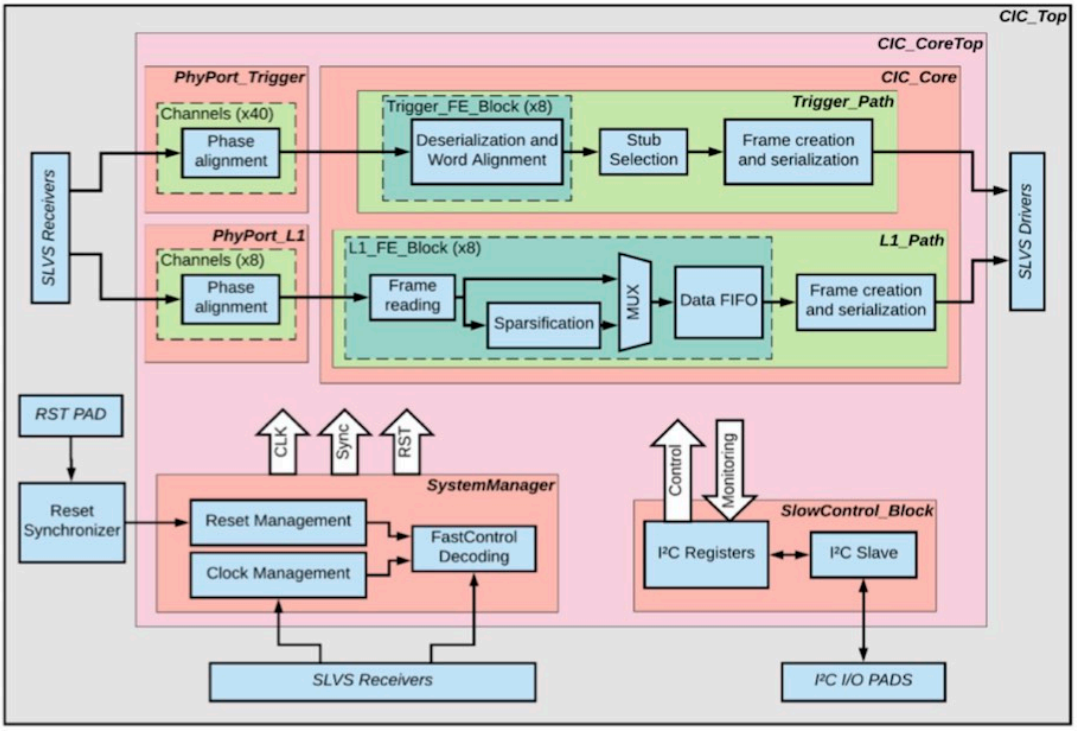


Figure 4.6: CIC block diagram

The concentrator model is presented in the block diagram of Figure 4.6. The basic blocks are represented in the light blue boxes, organised into four large structures (red frames). The data transfer blocks, which are located at the periphery, don't belong to these structures. They are implemented via precompiled elements provided by INFN [33] that can be introduced directly into the physical design (IP blocks), and are therefore not included

in the model.

The model itself is the block named **CIC_CoreTop** in Figure 4.6. The 4 major structures it contains are:

- **PhyPort**: this block is a simplified version of the lpGBT phase aligner unit, whose role is to re-phase all the inputs of the concentrator with respect to the 320MHz input clock. Minor phase shifts can indeed be induced between the different CBC/MPA output lines. We can not really talk about an IP block there, because small modifications had to be made to the model, and it was recompiled, but most of the work was already done upstream.
- **SystemManager**: this block manages all the clocks and reset procedures of the concentrator. To this end, it receives and decodes the Fast Control signal. It has been entirely developed by the Lyon group.
- **SlowControl**: manages slow control operations. This block was provided by the CERN microelectronics department.
- **CIC_Core**: this is the block that contains the data processing core of the concentrator. There are 2 independent channels: trigger and data. Each of these channels is itself divided into 2 steps: 8 data reception blocks (**FE_blocks**, in dark green), and a block processing and formatting the output information. As the SysManager, the CIC_Core has been fully developed at IP2I.

This modular structure has several advantages. The system is much simpler to understand and the blocks can be developed independently of each other. In terms of physical design, it is also much simpler to start from a very hierarchical model. This greatly facilitates the placement and routing of the different elements.

4.3.2/ CIC CORE DESCRIPTION

4.3.2.1/ TRIGGER PATH

In order to correctly interpret the data blocks from the dedicated lines of an MPA or CBC, phase alignment provided by the PhyPort block is not sufficient. Indeed, the first bit of line 1 must correspond to the first bit of the 4 other lines, otherwise the information will not be decoded correctly. It is therefore necessary for the 5 input lines to be in phase with each other. Moreover, the blocks being transmitted at 40 MHz to the CIC, it is also necessary to align all the lines with the corresponding clock.

The first step of the trigger model is therefore this word alignment sequence, the principle of which is shown in Figure 4.7, in the case of the 2S module. Once this procedure is applied, the 5 lines of each CBC/MPA entering the CIC are aligned with a 40/20MHz clock respectively.

The final alignment step is to phase-in the 8 CBC/MPA data. The bunch crossing N of the CBC1 must be interpreted at the same time as the corresponding BX of the other CBCs. A procedure similar to the previous one is then applied, but this time for all 40 lines.

Once the input channels are aligned, the trigger data can be decoded by the CIC. During 8 collisions, the stubs of each CBC/MPA are collected and grouped into a temporary

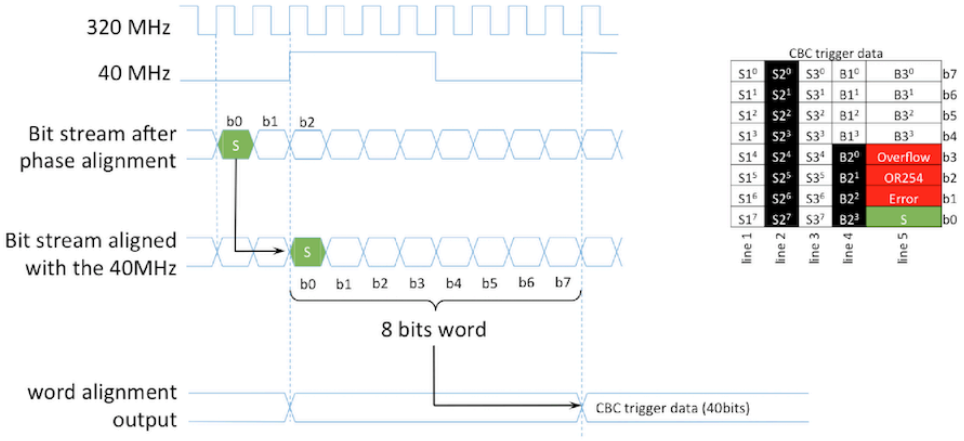


Figure 4.7: Word alignment in the 2S case

register. In the case where the number of stubs exceeds the size of the register, priority is given to the stubs with the smaller absolute width. In order to optimise the latency and minimise the complexity of the routing, a bitonic sort has been implemented in order to perform the sorting on-the-fly [34].

Once the stubs of the last collision have been retrieved and stored, the output block is ready to be transmitted to the lpGBT. The characteristics of the packet (number of stubs, potential errors, position in the cycle) are added to a predefined header, which is common to all types of modules, at the end of the frame formatting. The architecture of the trigger path is entirely pipelined, and the latency between the reception of the first stubs and the emission of the output block is constant and below 400ns (this value was confirmed during the CIC1 tests). Low latency is a very important point with respect to the track trigger.

4.3.2.2/ L1 DATA PATH

The operating principle of the L1 channel is slightly more complex than the trigger one. The type of data received by this block is indeed very heterogeneous: non-sparsified data for the CBC, sparsified for the MPA. The 8 MPA data are totally asynchronous, while the 8 CBC frames arrive at the same time for the 8 chips,... The management of these discrepancies imposes a significant flexibility to the L1 path implementation, synonym with complexity. Due to tight schedule constraints, a simplified version of L1 path was implemented in the CIC1. This version handled the normal data correctly, but its power consumption was not optimised and the error handling was incomplete. Here we describe the final version of the L1 channel, which has been put in place in CIC2.

Its working principle is sketched in Figure 4.8. As for the trigger channel, data reception is done separately for the 8 ASICs. In each of the 8 reception blocks, the information is received by an L1 Data Controller unit. This fundamental block controls in particular the validity of the received L1 ID. This identifier tags the input frame, and it is thanks to it that the 8 frames of a given L1 event can be merged. Particular attention is therefore paid to the reception of this parameter, and especially to its non-reception. Indeed, the data being sent asynchronously, the CIC must report to the DAQ when a frame is missing.

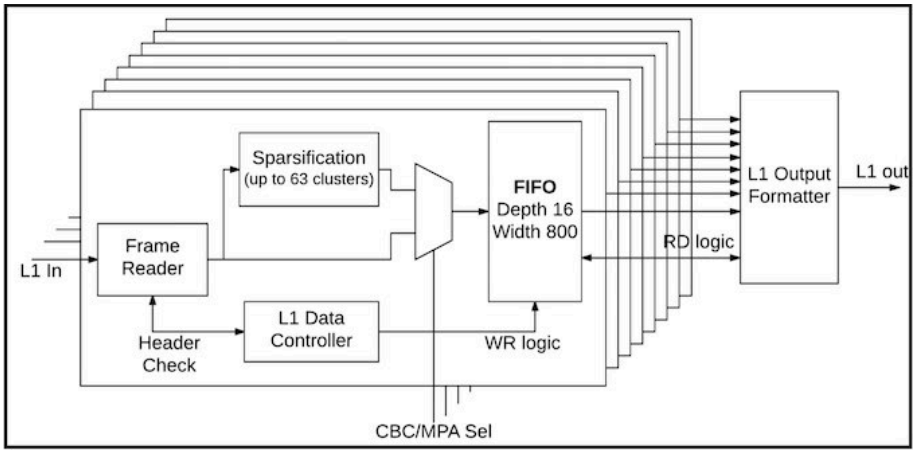


Figure 4.8: L1 path data flow

Once processed, the data received by a block is stored in a FIFO of depth 16. This means that up to 16 L1 events can be stored in parallel in a CIC. For each of these events, the frame size must not exceed 800 bits. These parameters stem from the studies presented previously. We note in passing that the CIC must contain a memory space of respectable size: 8x16x800=102400 bits.

In addition to these 8 FIFOs, the received L1IDs are stored in a master FIFO, which is then used by the L1 output formatter block as a reference to retrieve and validate the information. This last block retrieves, for each L1ID contained in the master FIFO, the 8 corresponding frames in the FE blocks, in order to form the final frame. If one or more blocks are missing, error bits are enabled in the output frame header. This operating mode ensures the continuity of the data taking: the CIC does not get stuck in the absence of one or more frames.

We described here the main features of the L1 path. However, this part of the model has many subtleties that make it extremely complex to develop, and the interested reader will find a detailed description of these elements in the note describing the specifications of the CIC in appendix C.3.

4.3.3/ MODEL VERIFICATION

Before engaging into the physical design of the ASIC, it is important to check that the model is functional. This condition is obviously not enough to say that the ASIC will work, but it is highly necessary.

The verification system developed for the CIC is based on the test bench that was developed for the track trigger system, which is described in the third part of this document. The basic principle is modularity, the idea being to use the same software framework for all development stages. Thus, the basic model can be tested with the same environment as the final chip. The other advantage is that we are sure that we inject at each stage the same frames, at the same frequency, ...

4.4/ CIC PHYSICAL DESIGN

4.4.1/ DESIGN FLOW DESCRIPTION

The different steps of the design of an ASIC are sketched in Figure 4.9. It is an iterative process, which can be divided into 2 main consecutive parts: logical and physical design.

The logical design corresponds to the model, and the repartition of the system into distinct blocks. This is what we described in the previous section, which is summarised in Figure 4.6. The final product of this part is what is called the synthesised netlist. This netlist corresponds to the final CIC wiring scheme. It contains all the components and their interconnections, but no information about their placement in the chip.

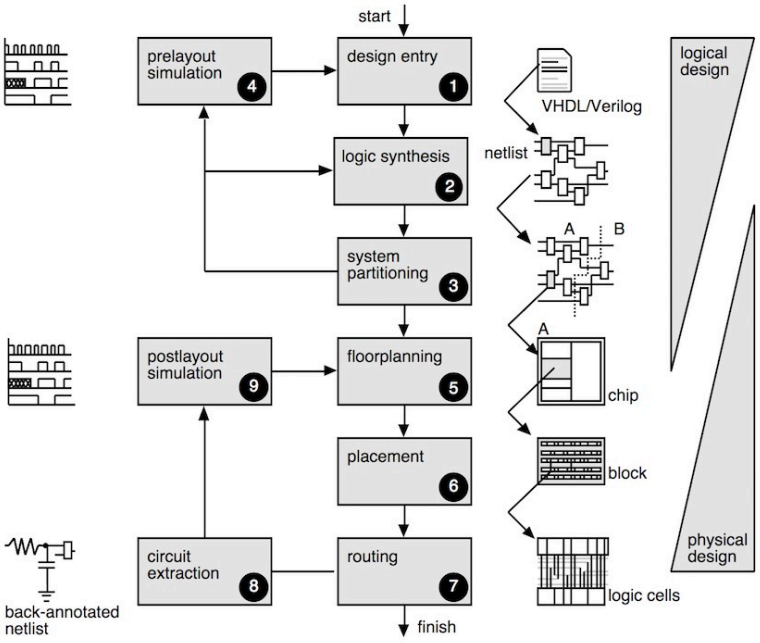


Figure 4.9: ASIC design steps [35]

This netlist is the starting point for the physical design, which aims to place the components in a reduced space to achieve a functional ASIC. This is a complex step because it must respect a very large number of constraints. In the case of CIC1, we are talking about 500,000 logic gates to be placed and interconnected in about 15mm^2.

Of course, all this is done using dedicated software tools. But that does not mean that there is nothing to do. A program can do a lot, provided you explain beforehand exactly what you want. The definition of the constraints and the initial conditions allowing a good convergence of these tools is the fundamental step of the physical design of an ASIC. It requires special attention and a very good understanding of the technology used.

4.4.2/ FROM MODEL TO FOUNDRY

The first step is the floorplan: the ASIC layout. This one is constrained by the dimensions of the CIC, which are themselves defined by the distribution of its connections with the hybrid. In the case of CIC, an additional constraint comes from the fact that the same

ASIC must be compatible with 2 totally different hybrids (PS and 2S). These were happily developed by the same team, which helped to minimise the risks associated with this point.

Once the dimensions and pinout are fixed, we can begin to place the IP blocks in the available space (Figure 4.10). We also define the circuits that will be used to power up the CIC cells. The power rails are contained in dedicated layers of the ASIC [1], and must be defined before others because they span over the entire surface of the ASIC (powering must be as uniform as possible) and thus constrain routing in all other layers. The very regular structure of the CIC2 power grid is shown in Figure 4.11. Once the routing is completed, dedicated tests (IR drop) are performed in order to check the validity of the powering scheme.

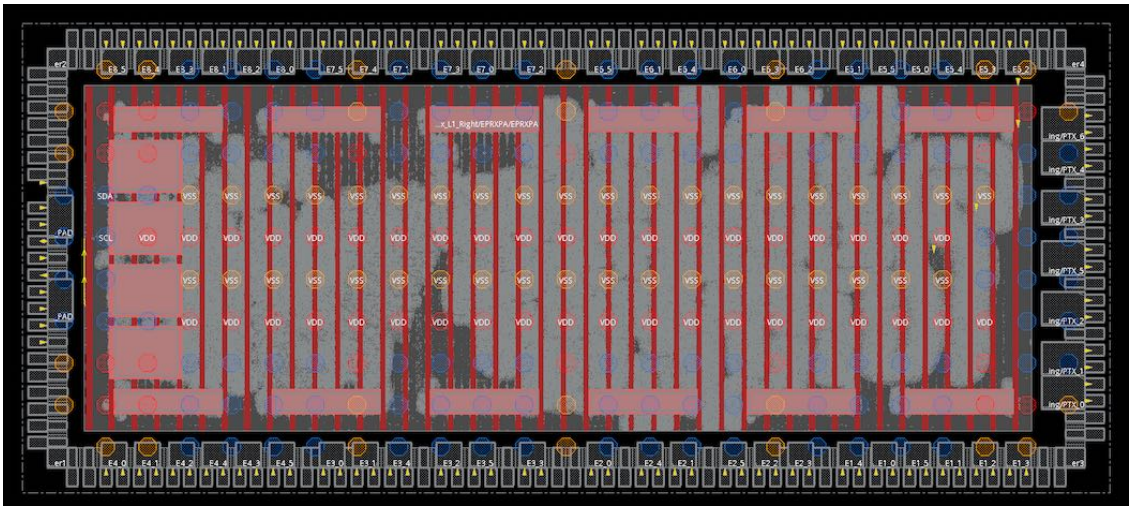


Figure 4.10: CIC2 floorplan

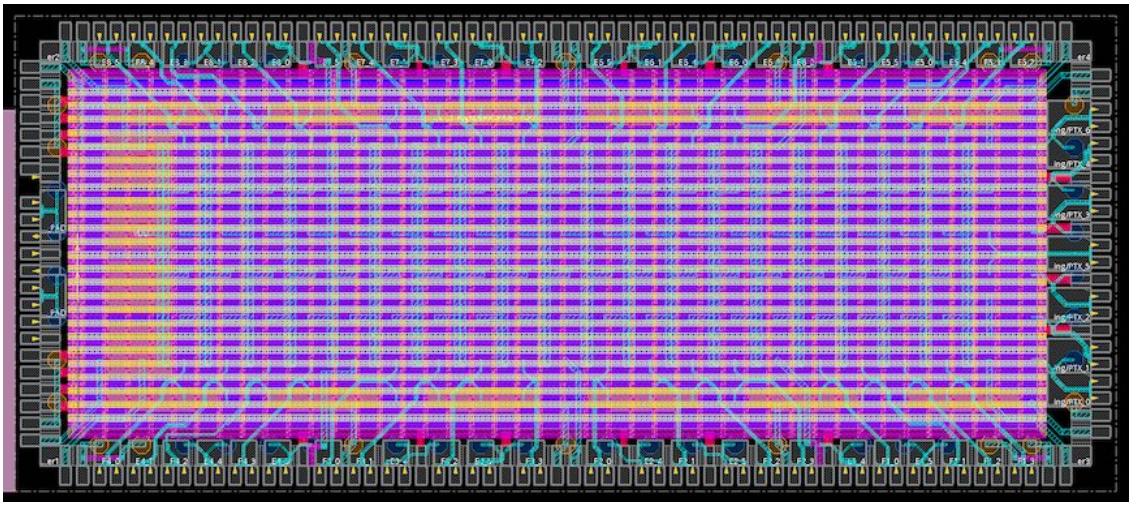


Figure 4.11: CIC2 power grid

Once these steps have been completed, the program performs a first quick placement of the whole synthesised netlist. This is what is know as floorplan. However, we are still

[1] The CIC has a total of 8 superimposed layers.

far from the final design, especially because all the cells are not yet synchronised to any clock.

The propagation of the clock (or rather clocks) to the different cells of the ASIC is the next step. This is a complex operation because all cells must be provided with information that enables them to correctly process signals. For a system like CIC that contains several hundreds of thousands of logical gates and receives information at a relatively high frequency, it is a delicate operation that usually takes several hours on a powerful machine. But it's this step, known as clock tree synthesis, that puts the ASIC into real world. Indeed, once it is successfully completed, one almost has a realistic version of the ASIC. There is only one remaining step: routing.

The routing consists in optimising all the connections of the ASIC, now clocked at the right frequency, so that all the signals can propagate correctly. A signal entering the CIC at a given time has in fact a limited latency to move from one cell to the other. If it arrives too late or too early, the software performing the routing reports a timing violation. After the first routing stage, there are still hundreds or even thousands of such violations. Optimisations are then performed, which are additional steps during which the software will redefine the connections to remove the violations. It is an iterative process because new violations will appear, but if the initial conditions of the project have been correctly defined, 4 or 5 iterations are generally sufficient to obtain a circuit without violation. This very important milestone is called timing-closure.

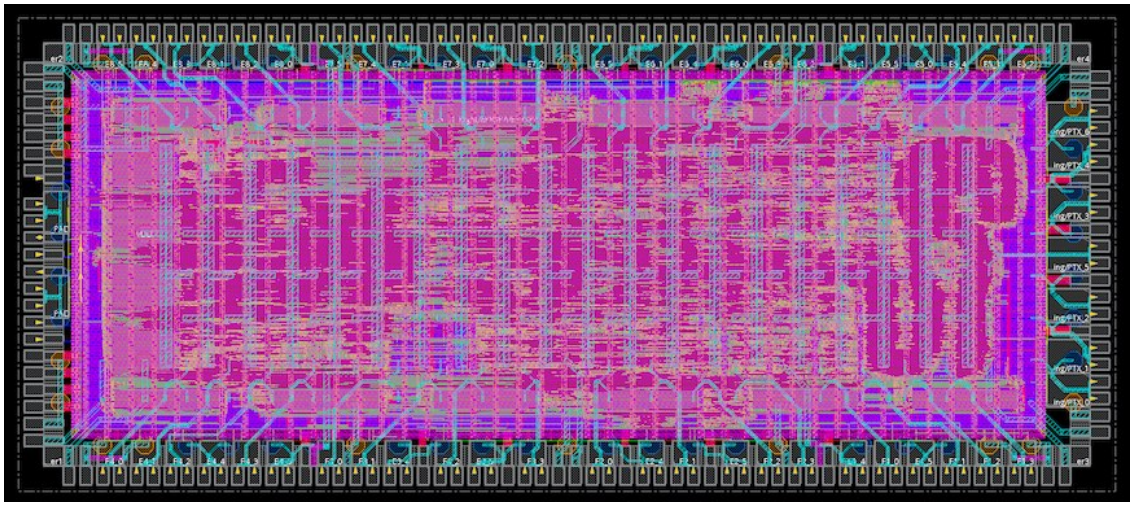


Figure 4.12: Routed CIC2

In the opposite case, the constraints of the program, or even the model if the violations are irreducible or too important, are reviewed. Obtaining the timing closure is definitely the longest stage of the physical design but also the most important. Once this is completed, we have a realistic system: the routed design (Figure 4.12).

But before sending this circuit to foundry, it must still overcome a whole series of verifications. As said above, what is engraved is engraved, so it is important to make sure that what we are going to produce corresponds to the specifications.

First of all, one checks with the simulation tools used for the model that the behaviour of the final system corresponds to what is expected. This is a second behavioural verification, but this time with a realistic system (in particular with parasitics).

Then, we estimate the ASIC power consumption, as well as the voltage drops on the whole system (IR drop). The power must remain reasonably below the nominal values, and the IR drop (Figure 4.13) must be less than 5% of the supply voltage. Indeed, if a cell is not powered properly, it may induce a global malfunction. This is the reason why power grid is so important.

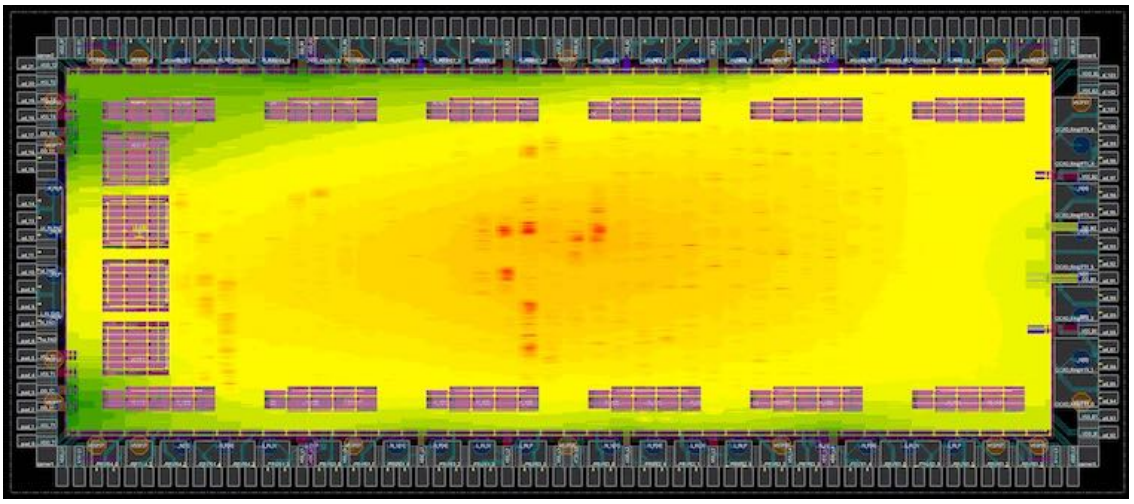


Figure 4.13: CIC2 IR drop analysis. Scale of the largest IR drop (in red) is 10mV.

Once all these verifications have been carried out, a last batch of very important checks is done: DRC and LVS.

The DRC (for Design Rule Check) is a control of the final design conformity with the foundry rules (in our case TSMC). Some rules must be respected, but it also possible to waive some of them (the founder obviously discharging any responsibility in case of problems).

The LVS (for Layout Versus Schematic) is a verification of the compatibility between the final netlist and the synthesised one. Behavioural simulation does indeed not verify that all the connections of the final design are strictly identical to the connections initially defined. These could be modified by the routing software during the development process. One must ensures that these changes did not have any impact on the overall behaviour of the ASIC.

Once these 2 last steps have been validated, a complete image of the ASIC is produced and sent to the founder via an intermediary who performs a final crosscheck. The final component comes back about 2 months later. We can then go trough the testing phase.

4.5/ CIC1 CHARACTERISATION

4.5.1/ TEST BENCH DESCRIPTION

The purpose of the CIC standalone tests is to verify that the chip meets the functional specifications. The points to check are consumption, data processing, and also radiation hardness, if applicable.

Once these verifications are done, the CIC can be put in its final environment. This test,

which we will briefly discuss at the end of this chapter, is not under the responsibility of our group.

The CIC1 contains all the functionalities of the final chip and has the same footprint and form factor. The main difference was the radiation hardness, which was not implemented yet. Nevertheless, the test system of both versions will be identical. It is shown in Figure 4.14 and consists of three cards: the CIC carrier, the interface board, and the control board. The first 2 boards were developed at IP2I, the third is a Xilinx KCU105 evaluation board.

The CIC carrier board is a small FMC mezzanine on which the ASIC is wire-bonded. One of these equipped with a CIC1, can be seen in Figure 4.15. This is a test system, the ASIC is soldered permanently and can not be reused afterwards. About thirty such cards have been produced and tested for CIC1.

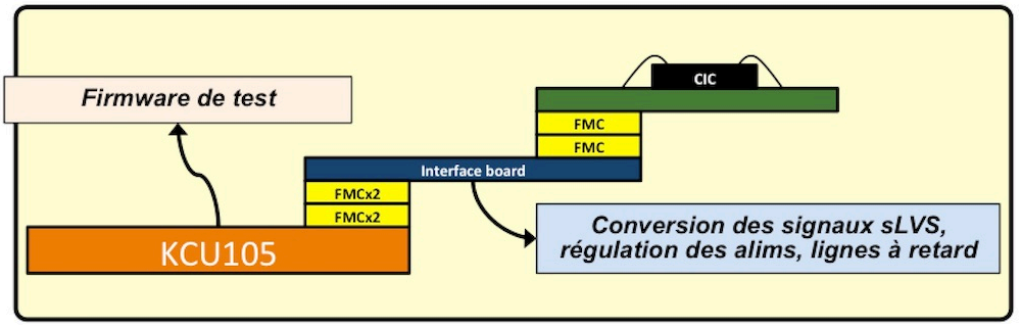


Figure 4.14: Standalone CIC test bench

The interface board is used to precisely monitor the ASIC powering, and to convert the signals sent by the evaluation board into a format understandable by the CIC. It can be seen as a transit system.

Finally, the evaluation board contains all the test firmware and features necessary to run smoothly (eg some RAM to store test frames). It is very largely inspired by the system developed for our track trigger test bench, which is described in the last chapter of this document.

The control of the whole system is handled by a software framework encoded in Python langage, interfaced from a PC with the evaluation board. Again, the track trigger project developed during the preceding years (cf. Chapter 7) provided a solid starting point that allowed us to be operational upon receipt of the first CIC1s in January 2019.

4.5.2/ RESULTS

When receiving an ASIC, the first thing to check is that it is alive and able to receive and send back information. The first block that we test is therefore the slow control, because without it, it is impossible to communicate with the component. Once this first test is validated, we can move on to more serious stuff. Otherwise, we test another chip, and if the problem persists, we can start to panic... Fortunately, this was not the case for CIC1, and we were able to immediately move to the next step .

The CIC is a digital ASIC that processes data, so it must be verified that it correctly processes the information sent in a normal situation. Thanks to the functional verification

Figure 4.15: Wire-bonded CIC1 on the carrier board.

tools developed to test the model and the netlist, this test could be done very quickly. Within few days it has been possible to establish the proper functioning of the CIC for all operating stages (including configuration), and for all types of data streams.

After this validation, the power consumption of the chip was measured. The result of this study is presented in Figure 4.16. It shows the power budget of a CIC1 in 2S mode with transmission of information on the L1 data path, for all the operating steps of the ASIC, from startup to data sending. It was thus possible to verify that the consumption was slightly lower than expected, while remaining compatible with the estimations, and in any cases lower than the available budget.

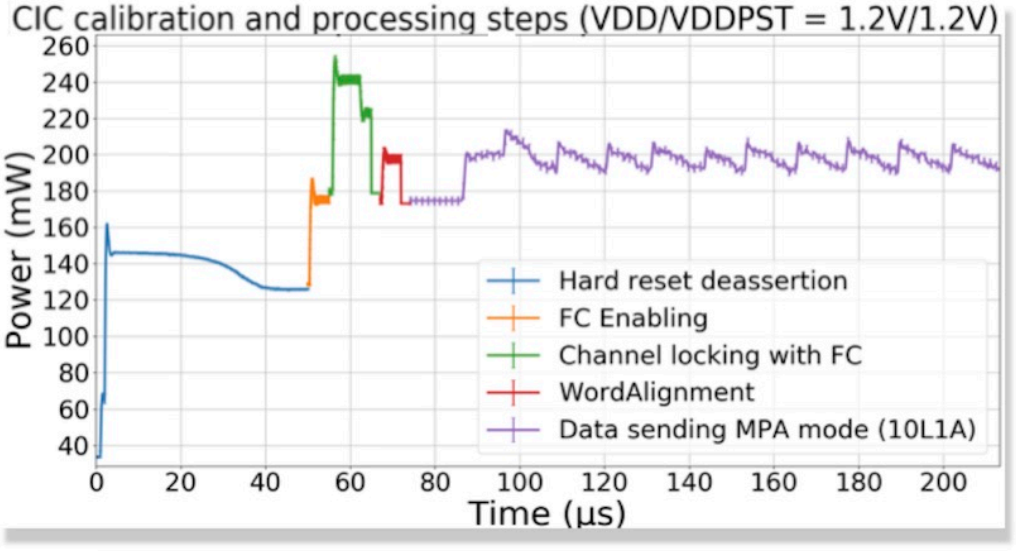


Figure 4.16: CIC1 power consumption during different operating stages.

Finally, CIC1 was exposed to a dose of 2.1 MGray X-rays (the total expected dose over the entire operation of HL-LHC being 1MGray). This TID test did not reveal premature degradation of the component due to radiation. This of course does not mean that CIC should not be hardened. Hardening prevents SEUs (Single Event Upset), which are not related to ageing due to total irradiation dose.

4.5.3/ CIC1 IN 2S MODULE FRONTEND CHAIN

CIC1 has also been tested in a more complete acquisition chain. These tests were conducted at CERN in spring 2019. The CIC1 was connected with 8 CBCs, thus emulating a full frontend hybrid of the 2S module. These tests were successful and fully validated the communication between these 2 ASICs, as well as the functioning of CIC1 under realistic conditions. This is a major milestone towards the realisation of a first complete 2S module prototype.

Eventually, apart from a few minor corrections, CIC1 worked perfectly. This allowed us to begin the development of CIC2 with a relative confidence.

4.6/ CIC2 DEVELOPMENT

4.6.1/ DESCRIPTION AND EXPECTED PERFORMANCES

CIC1 had to be developed quickly in order to meet very strict submission deadlines. As a result, some elements were deliberately put on hold. These points were radiation hardening, 640MHz operation, and L1 path power optimisation.

They were taken into account during the development of the CIC2, which took place from November 2018 to July 2019. The behaviour of the ASIC has not been modified, but its internal architecture has been thoroughly reviewed.

4.6.1.1/ RADIATION HARDENING

Due to their unpredictability, SEUs will be the main enemies at HL-LHC. Their rate is relatively difficult to anticipate. One can at best make sure that the component can react correctly in case a SEU occurs.

To make an ASIC robust against SEUs, a triplication procedure is applied. The most important ASIC registers are tripled, and the value of the register is measured by a majority vote. In this way, even if one SEU passes through one of the registers, the two others are not affected and the signal is not corrupted.

This simple technique is very effective, but by construction it triples the circuit complexity. Howevers, the registers do not all have the same importance an in practice, only the control registers are triplicated, and the data is not protected. The consequences of a SEU on a resynchronisation signal can indeed be much more serious than those of a SEU on a stub address.

This is the procedure that has been applied for CIC2. Some important data tough, such as the headers of the output frames, have also been protected. In the end, triplication increased total consumption by about 15%, which is quite acceptable.

4.6.1.2/ POWER CONSUMPTION OPTIMISATION

The consumption of CIC1 has not been optimised. In particular, we knew that the data path could be significantly improved.

This work required the rewriting of all the blocks linked to the L1 channel, but the effect of these optimisations is, as can be seen on the Table 4.2, significant.

Version	Maximum power consumption (in mw)
CIC1 (1V)	241
CIC2 (1V)	207
CIC1 (1.2V)	322
CIC2 (1.2V)	235

Table 4.2: Comparing power consumption estimated for CIC1 and CIC2.

On this table are compared the maximum power consumption of the two versions of the CIC, in the 2 modes, estimated with simulations. Although triplication was added, CIC2 consumption decreased significantly compared to CIC1. It shows the efficiency of L1 path optimization. This result remains to be confirmed with the physical version of CIC2, but it is very encouraging.

4.6.1.3/ PROJECT STATUS

The CIC2 design was completed in July 2019, when the project was submitted to TSMC. The ASICs were back in October and are currently being tested at IP2I. The tests, which will this time include a testbeam campaign, will extend until the first quarter of 2020.

4.6.2/ NEXT STEPS

CIC2 is expected to be a pre-production ASIC. This means that if it everything goes according to plan, it should be possible to use it on the final tracker modules, and to launch the final production of the 30000 final chips, which is currently planned for summer 2020.

4.7/ SUMMARY

At the time of writing, I do not know yet whether CIC2 will meet expectations or not. But the CIC1 is fully functional, and it is the first complete ASIC realised by the IN2P3 in this technology node. One can therefore already consider this project as a success.

To get there, we had to go through a number of events. The list is long, and I will not dwell on it here, but if there is one lesson to be learnt, it is that a project like the CIC can not be improvised. The rules stated in the part 4.1.2 are definitely the necessary conditions to start with this type of business.

IV

Analyzing data

5

BUILDING TRACKS, FASTER...

How to extract and process very rapidly tracker data ? We have devoted the previous part to solve the first problem, we must now tackle the second. This chapter draws up an inventory of the problematic and briefly presents the solutions envisaged for the CMS tracker.

5.1/ INTRODUCTION

5.1.1/ WHY?

As already stated in a previous chapter, online event selection will be a major problem at the HL-LHC. How do you know, when every collision induces a huge amount of pileup interaction, whether an event should be recorded or not?

The tracker is a very powerful selection tool, but so far it was not possible to use its data within a few microseconds. We saw in the previous part that the future CMS tracker will provide a solution to this problem thanks to the stubs.

But what to do with those signals? At HL-LHC, with an average pileup of 200, approximately 20000 stubs will be collected at each beam crossing. We will then have a very short amount of time to reconstruct tracks. How to proceed?

Generally, the best solution for solving a problem is to divide it as much as possible into independent subtasks. This observation applies perfectly to the track trigger problem, where we can immediately identify two independent stages: data distribution and track reconstruction. If we will quickly address the first point in the context of CMS, we will focus primarily on the second step in the rest of this document

Regarding the track reconstruction, it is again possible to reduce the work to several separate tasks. Indeed, this process is done systematically in two stages: the identification of the track candidates (pattern recognition), and the adjustment of their parameters (fit). In terms of latency, pattern recognition is the bottleneck. Any track trigger algorithm developer must first and foremost work to unlock this first latch.

5.1.2/ STATE OF THE ART

Until very recently, almost all real-time track reconstruction systems deployed in tracking detectors were based on associative memories.

An associative memory relies on a set of content addressable memories (CAM). CAM are a particular type of memory cell used primarily in very high speed search applications. Unlike the standard memory of a computer (RAM), for which the user provides an address and retrieves the data word stored therein, a CAM performs the reverse operation: the user provides a data word and retrieves, where applicable, all the addresses in which the word is stored.

This type of operation, which may seem strange at first glance, is of enormous interest for a real-time pattern recognition problem like ours. Indeed, since the data request is broadcasted simultaneously on all the elements of the memory, the processing latency depends linearly on the quantity of input stubs, where most of the classical methods will have a quadratic complexity. The potential of this process is therefore very clear, the main task is to adapt it to pattern recognition in a tracking detector.

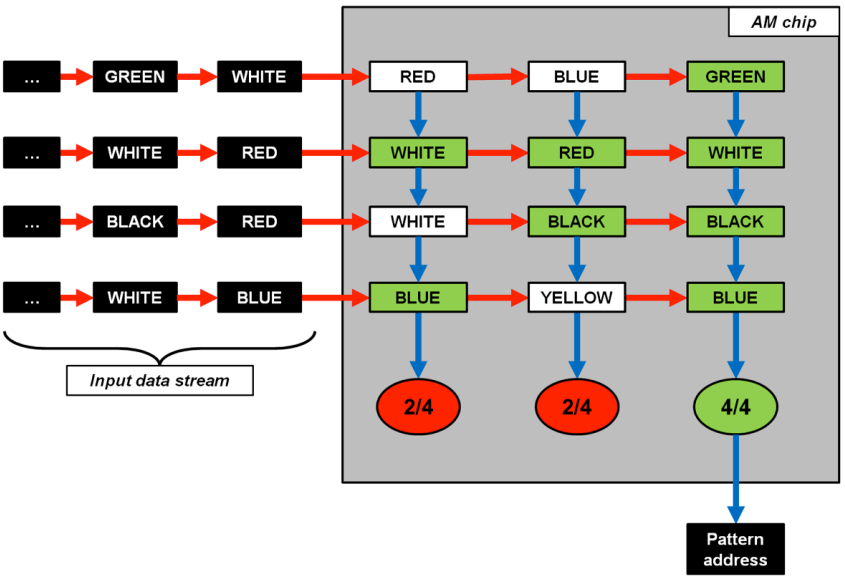


Figure 5.1: AM chip working principle.

Commercial CAMs were used in H1 experiment of the HERA positron-proton collider in the early 1990s [36]. But the principle of associative memory as we prospected it in CMS, and as it is presented in Figure 5.1 was proposed for the first time at the end of the eighties [37] and implemented in the late 90s in the CDF experiment [38, 39].

In this system, each track template (pattern) is stored in a single memory area, but this consists of N independent words: the addresses. Each address can be identified to the position of a potential signal on a detector layer (eg a stub). The matching is performed internally for each of the patterns, on the basis of a majority vote of the same type than the one one used to triplicate the CIC registers.

The Figure 5.1 allows to simply explain the pattern matching principle. In this example, the pattern is made of four words. As can be seen, the four input lines are independent of one another. The words of a line can be transmitted in random order; all matches are

known as soon as the last word is passed into the system. As the number of words can differ from one line to another, the time required to match on a line depends linearly on the number of words.

The pattern matching using associative memories is particularly suitable for ASIC porting. The first AM chips were developed in the mid-1990s. It was with this first generation, which contained only a few hundred patterns, that this technique was successfully used in CDF. This allowed the detector to significantly improve its trigger system and make important discoveries, especially in the field of heavy flavors. The developed system has become a reference, so it is currently planned to put it in place in the ATLAS detector [40]. For this purpose, the seventh generation AM chip is being finalized. The chips installed in ATLAS will contain several hundreds of thousands of patterns.

This system has logically been the basis of the project developed for CMS since 2007 [41]. However, we will see that other methods have been developed since then in CMS.

5.2/ A TRACK TRIGGER FOR CMS

5.2.1/ PROBLEM

Implementing a fast track reconstruction system in CMS or ATLAS involves taking an order of magnitude in terms of size and complexity compared to previous systems. In the case of CMS, the challenge is the following: how to handle a stream of 50Tbps of data in a few microseconds? These 2 constraints are usual in high energy physics, but it is much less common to find them merged into the same problem.

As mentioned above, it is important to consider the following two steps, data distribution and track reconstruction, as independently as possible. In the case of CMS, it can be considered that the latency available for these two steps is about 2 and 4 microseconds respectively. Before we focus more specifically on the track reconstruction in the following chapters, we will describe the outline of a complete system in the case of CMS.

5.2.1.1/ DATA DISTRIBUTION

The track reconstruction will be done outside the detector on boards located in the counting rooms. At the end of the previous part we left the stubs in the optical fibers leaving the modules. How to route them to the board that will build the tracks? At which speed?

Here again we must divide the problem. Indeed, it is quite unimaginable to route 50Tb of data every second into a single processing board. Therefore we divide the information in space, but also in time. The final system is thus composed of a certain number of boards which each process the information of a part of the detector (reduction of the quantity of data to be processed) for a part of the collisions (increase of the time available for processing).

These independent sectors are called trigger towers. A projection in the R/Z plane of the 48 CMS towers (the division into 8 equal sections in ϕ leads to 48 sectors), corresponding to the AM approach, are shown in Figure 5.2.

The data of each tower are managed by a set of 11 ATCA boards hosted in a shelf

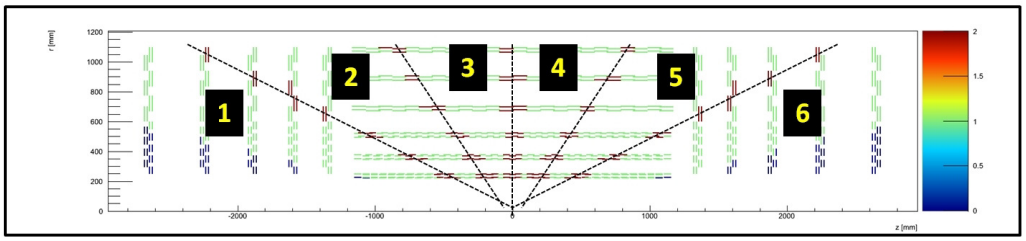


Figure 5.2: Data distribution, space multiplexing

adapted to this standard very widely used in telecommunications. The principle of time multiplexing implemented for the CMS AM approach is described in Figure 5.3.

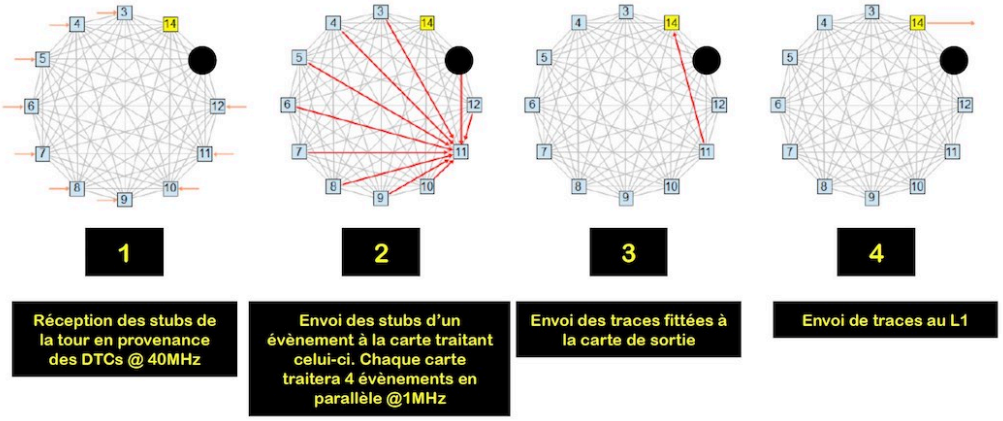


Figure 5.3: Data distribution, time multiplexing

Every 25 ns, the stubs of the event N, contained in the tower considered, are sent to the corresponding shelves where they are distributed in parallel to 10 processing boards (step 1). Each board then redistributes the fragments of event N to the board processing this event (step 2). Each board hosting 4 reconstruction units, it is thus possible to process 40 events in parallel, and thus have 1 microsecond instead of 25 ns to perform the first step of reconstruction (pattern recognition). Time multiplexing thus enables to reach the latency required to perform the various steps of the reconstruction, provided that the latency of each step does not exceed 1 microsecond.

5.2.1.2/ TRACK RECONSTRUCTION

Once the data is correctly transmitted, we have all the stubs of collision N distributed in each of the 48 sectors. To reconstruct the tracks in a given sector, we have 3 to 4 microseconds in total. Based on that, given the time multiplexing, the reconstruction must be done in successive steps of about a microsecond.

As we have seen above, the track reconstruction is actually based on consecutive stages, pattern recognition and fit. The pattern recognition is obviously the most complex to achieve in a very short latency. Therefore, it is not very surprising to see that the three concurrent approaches that have been reviewed in CMS in the preceding years mostly differ on this particular step.

Indeed, even if these methods may look distinct on paper, a careful study shows that

the main difference lies mainly in the pattern recognition stage. The other steps (data distribution and track fit) are very easily adaptable from one method to another.

5.2.2/ FAST PATTERN RECOGNITION IN CMS

5.2.2.1/ ASSOCIATIVE MEMORIES

This approach has long been, for reasons mentioned above, the only one studied in CMS. The main bottleneck of this technique is the development of an AM chip dedicated to CMS. We have seen that the realization of an ASIC is anything but a peaceful journey, the risk of delays has always been one of the main arguments against this approach. We will come back to this in more detail in the following chapters but, as you will understand, this is not the approach ultimately chosen by CMS.

5.2.2.2/ FPGA

Field-Programmable Gate-Array (FPGA) are reconfigurable chips. The capabilities of these systems have grown exponentially over the last 20 years, and today the most powerful components can perform extremely complex operations such as a fast pattern recognition algorithm. Today, FPGAs are ubiquitous in counting rooms, so it's not surprising to see them playing currently a major role in real-time tracking.

The fact that they are reconfigurable can be seen as a significant advantage over an ASIC such as the AM chip [1] because it is possible at any time to improve system performance by reprogramming a more powerful algorithm in the FPGA. On the other hand, the consumption of an FPGA is greater than that of an ASIC, and the identification by associative memories, which is not yet adaptable to the current FPGAs, is the only method totally linear and pileup independent. But algorithms such as the Hough transform, or a simplified Kalman filter [42, 43], have provided sufficiently promising results to convince CMS tracker management to favor this approach at the end of the year 2016.

From a strictly pragmatic point of view, the development of an algorithm on a FPGA is less risky than the development of an ASIC. If the estimated performance is comparable, the FPGA will generally have a significant advantage.

[1] We will see however later that an AM chip also has some flexibility

6

AN ASSOCIATIVE MEMORY BASED APPROACH FOR CMS

The key to a real-time trace reconstruction system is the pattern recognition. Before developing a complete demonstrator around associative memories, it must be ensured that these are usable in CMS. Our first contribution, between 2012 and 2014, was to develop a complete AM chip emulation to assess the potential of such an approach.

6.1/ ALGORITHM DESCRIPTION

6.1.1/ FOREWORD

The principle of a AM-based pattern recognition is sketched on the Fig. 6.1.

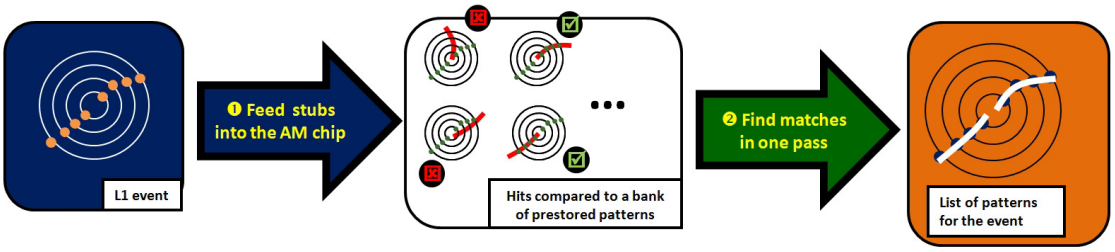


Figure 6.1: AM-based track reconstruction.

The stubs reconstructed for each event are injected into an AM chip where they are compared to a bank of patterns stored beforehand. The matched patterns are then transmitted to subsequent unit performing the fit.

In the context of the tracker these patterns can be seen as low granularity tracks. They are defined once for all using simulated events and they are supposed to provide a good coverage of the tracks reconstructible in the detector.

Each of the 48 towers of the system will have its own independent pattern bank. The name of the game is to create the 48 best performing banks for the future tracker.

6.1.2/ BANK, PATTERNS, AND SUPERSTRIPS

6.1.2.1/ WHAT IS A GOOD PATTERN BANK?

Before going further into AM pattern recognition technicalities, it is important to precise what we are aiming for exactly. Pattern recognition is a filtering stage, we are not fitting tracks here, but we are settling a clean environment for the track fitting. Therefore, the impact of this stage on track efficiency should stay as transparent as possible, but in the meantime we should increase significantly the stub purity. On the other hand we should consider the technical limitation of the AM chips and of the boards holding them. It means that pattern banks should have a reasonable size, and the number of roads matched in a given event should remain small.

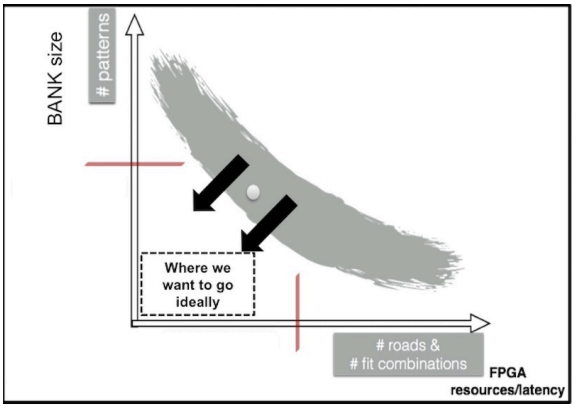


Figure 6.2: Pattern bank optimisation

As always in this type of problem the solution will consist in finding the best tradeoff. Figure 6.2 summarises the problem and defines the operating point that we will try to reach. The ordinate corresponds to the size of the pattern bank and on the abscissa is represented an indicator of the purity of the output data (the purity decreases when this indicator increases). The shaded area represents the operating points of the pattern banks. A small bank will have large patterns that will be activated more often and less pure. The extreme case would be a bank with only one pattern (the whole sector), which would be very effective but a bit useless! In the opposite case, one can imagine a bank of very large size containing patterns of minimum width (1 strip). Such a bank would obviously have optimum purity, but would require a too large number of chips.

If we get back to Figure 6.2, we will seek to reach a working point as close as possible to the origin. The ideal bank will be small, efficient and will generate a small number of routes with the greatest achievable purity. To determine this bank, we will have to look at its basic element: the pattern.

6.1.2.2/ PATTERN DEFINITION

If the stub is the key element for the track trigger, the pattern is the main object of the AM approach. A pattern can be defined as a road. Each road may contain one or more tracks, as shown on the Fig. 6.3.

On this picture one also get an idea of the pattern structure. Each pattern is made of

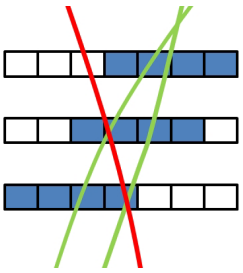


Figure 6.3: A pattern in a three-layers detector. Green tracks will match the pattern, red tracks won't.

one region per layer, called a superstrip. A superstrip is a group of strips/pixels, and is therefore heavily constrained by the detector itself. We can also easily understand that its width will have an impact on the bank filtering power and on the bank size, both parameter evolving in opposite directions. A pattern made of narrow superstrips will ensure a good filtering power, but will lead to a very large bank, whereas a pattern made of wide superstrips will lead to small bank, with a poor filtering power. We will see later that mixing narrow and wide superstrips can actually lead to very interesting results.

6.1.3/ SUPERSTRIP DEFINITION

A superstrip is defined as an address describing the position of the region within a tower. Current hardware limit for the superstrip size is 16 bits. We based our studies on this limit, but this size might be increased, if necessary, in the future generation of AM chips.

Once the superstrip size has been fixed (one can set different sizes for different layer/disks), corresponding position information is encoded within a N-bits word (in our case 16): the superstrip address. All the geometry information is contained into this address, the AM chip itself just know about the address and is therefore completely independent from the detector geometry.

Superstrips are transmitted to the AM independently for each layer, thus the layer information doesn't has to be encoded. In order to understand how a superstrip is encoded, Fig. 6.4 shows an example of superstrip definition for the phase II tracker geometry based on the 16-bit size constraint.

AM chips have 8 independent input lines. For most sectors each line is assigned to a specific detector layer. It is therefore not necessary to encode the number of the layer in the superstrip. Hybrid sectors containing 9 distinct layers, it is necessary to share a line between 2 layers. But here too the number of the layer is not necessary for the superstrip, because these areas are geometrically independent. So we encode in the superstrip only 2 coordinates in each case: Z/ϕ in the barrel, and R/ϕ in the endcaps.

A coding example is sketched on Fig. 6.5 for a barrel tower. In both coordinates, the number of bits necessary simply depends on the superstrip granularity. A first set of bits provides the module number along the corresponding coordinate: 5 bits for Z (one could have up to 24 modules in Z in one barrel tower) and 4 for ϕ (up to 15 modules per tower in the outermost layer). Then, for each coordinates, a second series of bits provides the superstrip position within the module. Fig. 6.4 shows a tracker strip module (in green), divided into 2 segments in Z and 1024 strip in ϕ. Therefore in order to describe all the

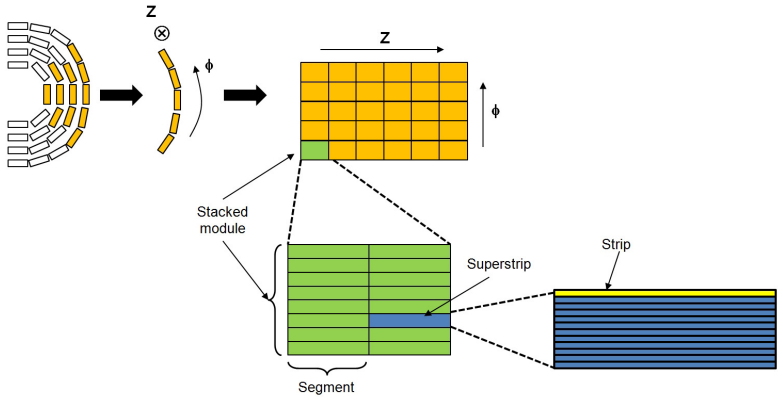


Figure 6.4: Geometric definition of a superstrip.

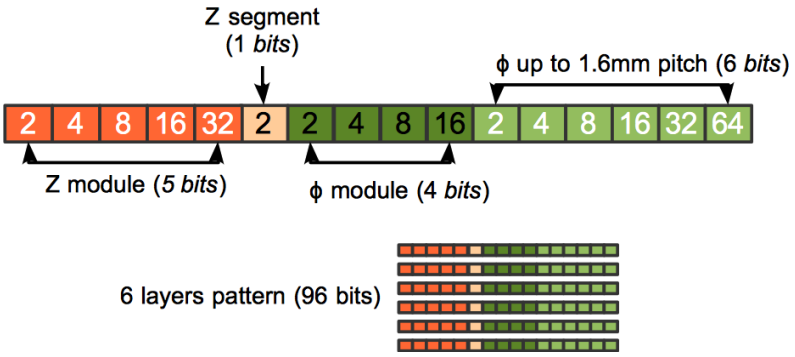


Figure 6.5: Address definition of a superstrip and representation of a 6 layers pattern.

positions, one would need only 1 bit for Z, and 10 bits for ϕ. At least 16 strips are grouped to form a superstrip, thus leading to 5 or 6 bits in the address.

The encoding of the superstrip can be quite different from one layer to the other for a given sector. We will indeed see later that it can be useful to increase the size of the superstrips with respect to distance to beam axis.

6.1.4/ Introducing ternary bits

As mentioned previously, pattern bank size is strongly related to the superstrip granularity. Rule of thumb is that dividing the superstrip size by a factor 2 leads to a factor 4 bank size reduction. In order to exploit the best possible resolution while keeping a reasonable number of patterns, one could fill the bank with variable resolution patterns [44].

To generate a bank with variable resolution pattern, we start by creating 2 banks: a high resolution bank, and a low resolution one. A low resolution pattern has a width greater than a factor of 2^N where N is between 1 and 3, relative to its high resolution counterpart. Then, as shown in the figure 6.6, the high resolution patterns belonging to the same low resolution pattern are merged. The variable resolution pattern is created via the procedure shown on Fig. 6.7. If both high-res superstrips are hit, we can neglect the least significant bit (LST) of the superstrip, we set it to DC (for Don't Care) in the pattern, otherwise we put the value of the high-res bit in the superstrip LST. The LST bit of the

Figure 6.6: Merging high-resolution patterns belonging to the same low-resolution pattern.

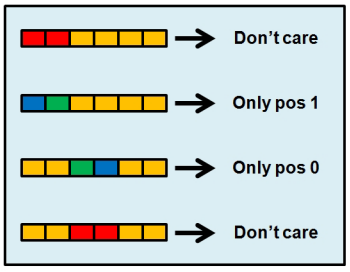


Figure 6.7: Associating the merging results to ternary bits values (1-bit case).

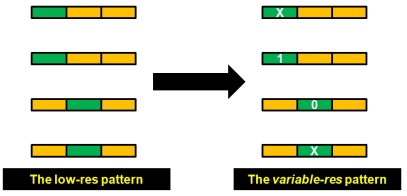


Figure 6.8: The variable resolution pattern compared to the low-res one.

superstrip is a ternary bit, it can therefore took three different values[1].

The main advantage of this technique is that instead of having two high-res patterns in the bank, one ends up with just one partially high-res pattern, as shown on Fig. 6.8. Of course one should expect a larger fake rate with the variable resolution bank than with the high-res one, but a much better one than with the low-res bank.

This procedure can of course be generalised by adding other ternary bits.There are indeed no limitations on the number of ternary bits per superstrip in the current AM chips. However, with more than one ternary bit, it becomes impossible to cover all the configurations. Fig. 6.9 shows the encoding of two ternary bits for all the configurations (15 configurations covered with 9 available codes). One sees that we don't gain anything w.r.t. the lowest resolution for 7 configurations, in particular for two configurations having just two superstrips hit (in the middle column).

Fig. 6.10 shows the mapping obtained using a mirror-encoding of the superstrip position (Gray encoding). The configurations leading to two DC bits are the same, except for the two configuration with only two superstrips hit, which are exchanged with two less track-like configurations.

With three DC bits the situation becomes even more permissive: 27 available mappings for 255 combinations. So this means that we will be able to use high resolution information only for 10% of the superstrips. A 3DC bits bank will therefore start to be very close to its low resolution counterpart. Going further clearly becomes irrelevant.

[1] In practice ternary bits are obtained using two classic bits. This means that each time one adds a DC bits, one looses one bit on superstrip size. For example with one DC bit, the superstrip address should fit within 15 bits.

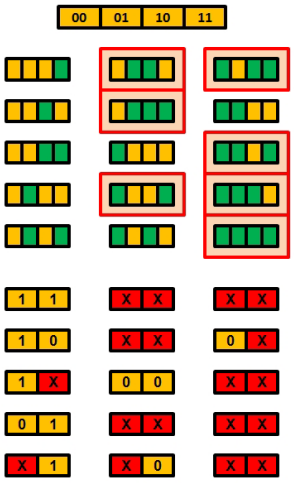


Figure 6.9: Coding 15 positions with 2 ternary bits (classic encoding).

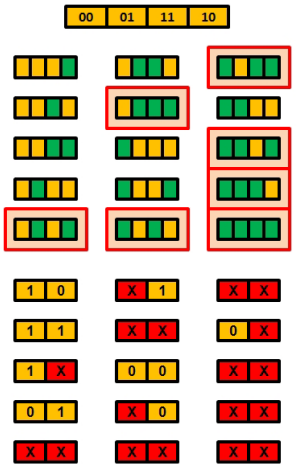


Figure 6.10: Coding 15 positions with 2 ternary bits (Gray encoding).

Since the AM chips are configurable, it is possible to use a different number of DC bits depending on the layer considered, this becomes particularly important when the basic size of the superstrip depends on the layer. We will come back to this in the next sections.

6.2/ AN AM CHIP FOR CMS

The generation of the optimal pattern of a bank involves a large number of parameters. In order to study the effect of each of these parameters, it is essential to have adapted simulation tools. Since these tools were not available in CMS when we joined this activity, our first contribution was to implement a complete software emulation of AM-based pattern recognition[2]. It consists of two components:

- **A bank generator**, which builds patterns from simulated data in the tracker (usually muons, more rarely electrons). The properties of the patterns (superstrips, ternary bits, number of layers, ...) are fully configurable when the generator is instantiated.
- **A pattern recognition emulator**, to get the list of patterns matched by an event. This component was first developed as a stand-alone C ++ code, then it was ported to the official CMS simulation framework.

6.2.1/ BANK GENERATOR

6.2.1.1/ PRINCIPLE

The generation of the pattern bank is performed using very large samples (several billions) of muons whose trajectory in the tracker is previously simulated. Only primary particles with transverse momentum larger than $2\ GeV/c$ are generated. This is slightly

[2]Developing this emulation is, moreover, the best way to understand in depth the operation of the AM chip

lower than the value of the specifications, and enables to evaluate more precisely the algorithm turn-on region.

For each trigger tower we create a sample of good particles that can be used for to grow the bank. A good particle induces at least one stub in the area in each layer it passes through.

In order to speed up the process, the generator starts from this skimmed sample of tens of millions of particles. The elaboration of the bank is then based on the following iterative procedure:

- We define the patterns for N particles and add them to the bank. The bank contains only unique patterns, so if the pattern computed for a given event is already present we do not add it a second time (the number of times it has been obtained, however, is conserved, thus defining the pattern popularity).
- At the end of the iteration one knows the number of patterns, denoted N_p effectively added to the bank during the iteration.
- We then calculate the coverage for this iteration :

$$coverage = 1 - \frac{N_p}{N} \tag{6.1}$$

- We iterate until the coverage reaches a pre-defined threshold, usually 90%.

Once the threshold is reached, the procedure is stopped and the program prepares the patterns with ternary bits from the high resolution patterns. The variable resolution patterns obtained are then saved and the new coverage rate is calculated. Just like the real AM chip, the generator can support up to 3 DC bits per superstrips for a given layer, and a variable number of DC bits per layers.

The use of DC bits mechanically increases the final coverage. Stopping the growth of the high-resolution bank at 90% yields a variable resolution bank with a rate well above 95%. The remaining inefficiency is compensated by the fact that patterns can be matched even if one stub is missing.

6.2.1.2/ TRAINING SAMPLES DEFINITION

Concerning the μ Monte Carlo samples used for the bank generation, it is important to use the correct p_T spectrum in order to ensure a sharp turn-on of the bank efficiency. The low p_T μ having larger bends than the high p_T μ, we will end up producing more patterns at low p_T. If we use a flat p_T spectrum, the coverage at high p_T will quickly rise and bias the average coverage value. The process will stop with a very poor coverage at low p_T (Fig. 6.11 a). Problem is solved by using a p_T spectrum flat in $\frac{1}{p_T}$. As we have more low p_T tracks, the corresponding patterns are created much more quickly. However, with this strategy, it will be difficult to populate the high p_T area (Fig. 6.11 b). We solve this point by creating 3 banks over 3 p_T ranges ([2,5], [5,20], [20,100]) and we finally merge them into a single bank: it ensures a good coverage over the whole p_T range (Fig. 6.11 c). Of course this merging is exclusive, and a pattern present in two sub-banks will be counted only once in the final one.

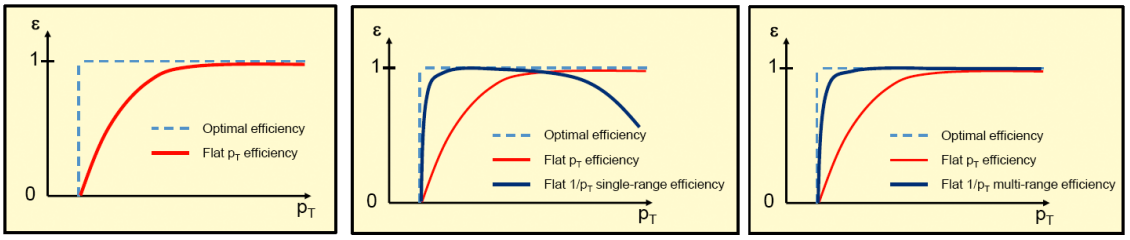


Figure 6.11: Turn-on curves using (a) a flat pT spectrum (b) a flat 1/pT spectrum (c) 3 ranges with a 1/pT spectrum

Once the bank is complete, it is possible to truncate it to a predetermined size, in order for example to rigorously emulate the capacity of one or more AM chips. The AM chip also has the ability to order the patterns. During our study we evaluated 2 ordering strategies: transverse momentum and popularity. Pattern momentum is the average impulsion of the tracks that were used to produce it, its popularity is the number of tracks that induced the pattern. These 2 methods each have their pros and cons but in the following, and unless otherwise stated, we used a popularity ordering because it allows to naturally maintain a high coverage rate even with a truncated bank.

6.2.2/ EMULATOR

The goal of the emulator is to get the list of matched roads from a given event. In input, we have the list of stubs coming from the detector and the pattern banks previously defined.

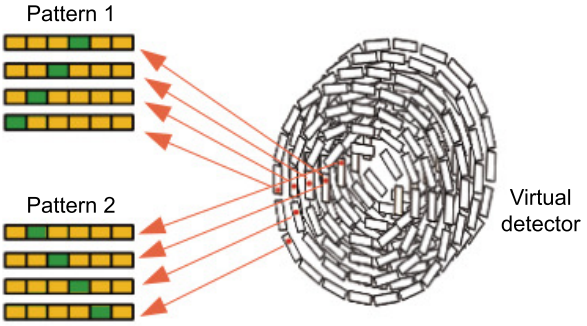


Figure 6.12: Each superstrip in the pattern bank points to a superstrip in the virtual detector.

A straightforward method would be to check for each stub whether it is contained in each pattern or not. This would lead to a total number of iterations of $N_{stubs} \cdot N_{patterns}$, which is clearly a problem for heavy events. Moreover, this very inefficient solution would not reflect the AM chip behaviour.

The solution implemented in our emulator is to use a virtual representation of the detector containing all the superstrips. We've seen previously that a pattern is made of one superstrip per layer. For each of these superstrips, we will create a pointer on the

corresponding virtual superstrip (see Fig. 6.12).

If several patterns are using the same superstrip, their pointers will be linked to the same virtual superstrip object. As a consequence, the activation of a single virtual superstrip is immediately seen by all patterns referencing it. The matching procedure is then performed as follows:

- For each stub we activate a flag on the corresponding superstrip in the virtual detector.
- For each pattern we count how many pointers are linked to activated superstrips in the virtual detector. If we are above a predefined threshold: the pattern is activated.

The number of steps therefore becomes $N_{stubs} + N_{patterns}$. As usually $N_{patterns}$ is much greater than N_{stubs} (millions of patterns versus thousands of stubs), the number of stubs in the event gets irrelevant for the computing time which becomes linear with the number of patterns. As in the real associative memory chip, it allows us to treat complex events in the same amount of time than simple PU-free particle gun events.

In order to test the potential of associative memories in CMS, this emulator has been ported to CMSSW. It has also been extensively documented to allow all groups involved in the project to carry out their own studies (see for example the following tutorial [45]). As we will see later, we were able to verify that this emulator is reproducing exactly the behaviour of the current AM chips.

6.3/ ALGORITHM PERFORMANCE IN CMS

6.3.1/ BANK OPTIMISATION

The goal of the AM pattern recognition step is to provide the cleanest possible environment for the subsequent track fitting stage. We have seen in the previous chapters that only a small fraction of the stubs are coming from the particles we are looking for. Therefore we will try to significantly increase this fraction.

The pattern bank optimisation procedure is a multi-variable process. First of all, the three different tower types are treated separately. Then, bank performances for the full tracker are evaluated.

The trigger sectors used are divided into 3 categories depending on the area covered in pseudo-speed: barrel, endcap, and hybrid. For each of these categories it is necessary to define the best possible bank. The parameters are obviously different for the 3 tower types, and the optimisation will lead us to very different banks. Since the patterns depend on several parameters (width, number of DC bits, etc.), the phase space to be tested is thus very large. The results presented here are the outcome of a long campaign made by several groups, but many improvements are still possible. For a nice overview of the latest developments since the study presented here, one can have a look at [46].

We present in this section the performances obtained with the flat geometry of the tracker (without tilted modules). The final geometry with tilted modules was introduced rather late at the end of 2016, just before the external review of the different approaches proposed for CMS. It is therefore with the flat configuration that most of the optimisations were

carried out until 2017, when we withdrew from the project. Nevertheless, we adapted the AM filtering to the final geometry and the preliminary results obtained were in agreement with those obtained for the flat geometry [47].

The first studies were carried out with banks containing constant width patterns (Figure 6.13 left). But this type of road induced very large banks, and was not accounting for the particle scattering effects. The greater the quantity of material traversed, the more the track is indeed likely to move away from the ideal trajectory. This phenomenon is better taken into account via the use of projective patterns (Figure 6.13 right), for which the width of the superstrip increases with the distance to the axis of the beam.

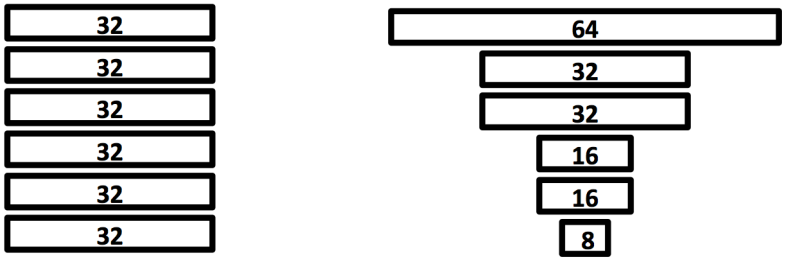


Figure 6.13: Fixed (left) and projective (right) patterns

In the same way, we observed that the performances of the banks were better when the number of ternary bits usable in each of the layers was variable.

The final banks from which the results presented in the following are derived are characterised by the width of the superstrips and the number of DC bits used in each layer. This information is summarised in Figs. 6.14, 6.15, and 6.16, for all three types of towers.

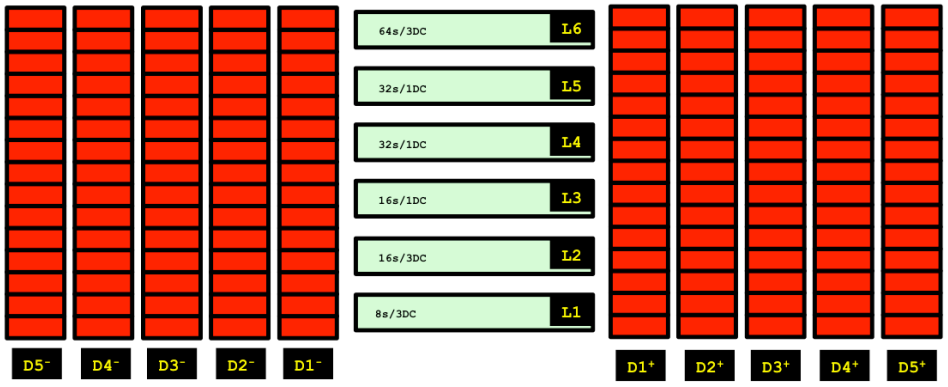


Figure 6.14: Barrel bank properties. Tracker parts involved in the bank are in green. Corresponding superstrips size (in strip units), and DC bit numbers are provided.

The bank size was limited to 1M patterns (2 AM chips) for the barrel and endcap sectors, and 1.5M patterns (3 AM chips) for the hybrid sectors.

In total, with this configuration, the AM approach would use 56M of patterns, or about 112 AM chips, for the full tracker. Taking into account the time multiplexing presented earlier, this represents a total of 4480 chips for the complete system, which is perfectly achievable.

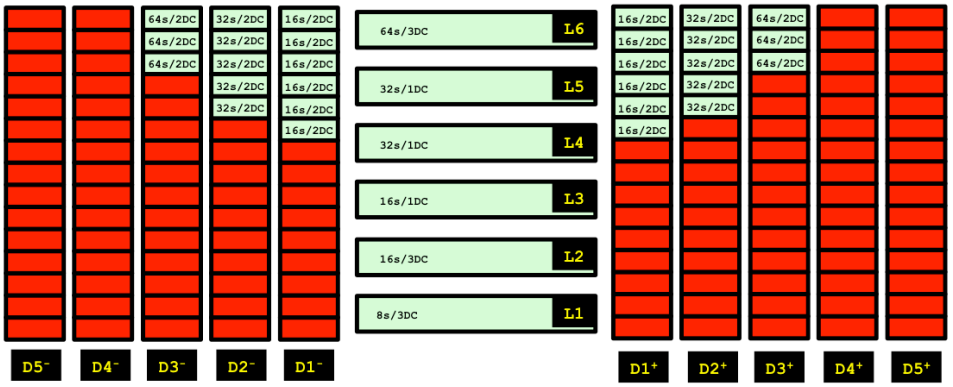


Figure 6.15: Hybrid bank properties. Tracker parts involved in the bank are in green. Corresponding superstrips size (in strip units), and DC bit numbers are provided.

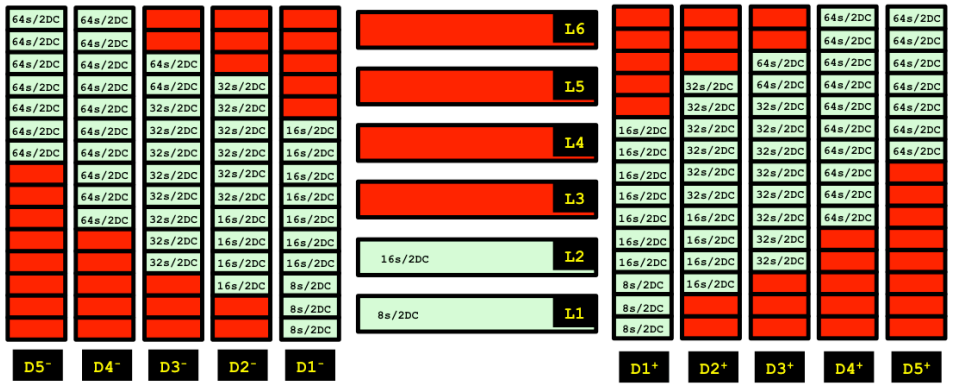


Figure 6.16: Endcap bank properties. Tracker parts involved in the bank are in green. Corresponding superstrips size (in strip units), and DC bit numbers are provided.

6.3.2/ EFFICIENCY AND FAKE RATE DEFINITION

In order to evaluate the performance of a pattern bank, several parameters must be studied: the efficiency, the fake rate, the purity of the output stubs, the average number of matched roads as well as the upper limit (value satisfied by 95% of events). The last parameter is related to the purely electronic nature of the procedure that we want to put in place. It is important to know the maximum size of the expected events, in order to estimate to what extent the system will be able to cope with them without losses.

The efficiency of trace reconstruction after associative memories is defined as follows:

$$\epsilon^{SR \to AM} = \epsilon^{SR} \cdot \epsilon^{ST} \cdot \epsilon^{TW} \cdot \epsilon^{AM} \tag{6.2}$$

where:

- ϵ^{SR}: stub reconstruction efficiency. This is the proportion of good tracks which are inducing at least 4 stubs in four distinct layers/disks.
- ϵ^{ST}: stub transmission efficiency. This is the proportion of tracks after SR which are keeping at least 4 stubs in four distinct layers/disks after the different frontend inefficiencies.

- ϵ^{TW}: trigger tower acceptance. Proportion of tracks after ST which are containing at least 4 stubs in four distinct layers/disks in at least one of the trigger towers.
- ϵ^{AM}: AM pattern recognition efficiency. Proportion of tracks after TW which are containing at least 4 stubs in four distinct layers/disks in at least one matched road.

This definition depends on what is considered as a good track. In the following, a good track will be defined as a track satisfying the following requirements:

- $|d_0| < 1cm$
- $|z_0| < 15cm$
- $|\eta| < 2.4$
- $p_T > 3GeV/c$
- At least one stub.

As already seen in Section 3.3.2, ϵ^{SR} and the denominator will strongly depend on the stub windows tuning used. With a cut of 2.5 strips in the innermost barrel layer, more tracking particles will induce at least one stub than with a cut at 1.5. On the other hand, as most of these particles will decay sooner, ϵ^{SR} might get lower.

Stub transmission inefficiencies are currently not fully taken into account in the simulation. As explained previously, the main source of data loss in the front-end chain is the transmission from the CIC chip to the lpGBT. At this level, data is transmitted in 8 bunch crossings block. In order to models the losses in CMSSW, one has to keep track of the previous events information. This breaks CMSSW coding rules. As there is currently no satisfying solution to address this problem, we will simply consider that $\epsilon^{ST} = 1$ in the following. We have indeed seen in Section **??** that these losses could be kept under control, in particular thanks to the stub windows tuning.

It should be noted that these two parameters are totally independent of the track reconstruction itself, eventually we speak of intrinsic detector efficiency for these terms.

The reconstruction algorithm will affect the last two terms, and more particularly ϵ^{AM}. It is indeed relatively simple to define the trigger towers in such a way as to obtain $\epsilon^{TW} \approx 1$[3].

For the fake rate, we base ourselves on the content of the matched road. This one is associated with a track if more than 4 stubs of the track (on 4 distinct layers) are contained in the pattern. Based on this definition, an activated pattern is considered as a fake if:

- It is not associated to any track
- It is associated to a track that does not fullfill the requirements defined previously.

We see that this definition does not take into account the purity of the stubs in the pattern. The patterns being relatively broad, they might contain stubs which are not belonging to the matched track. A true pattern will not necessarily be clean one. We will see in the following chapter how matched road purity can affect the rest of the algorithm.

[3]Define the cabling between the modules of these towers and the corresponding backend processing node is rather complex and comes out of this document.

6.3.3/ RESULTS

6.3.3.1/ EFFICIENCY

The efficiency was measured with both single particles (muons and electrons), and $PU+t\bar{t}$ samples (mainly pions and protons), for average PU values of 140 and 200.

On table 6.1 are summarised the overall efficiency results in the baseline configuration described above in two distinct p_T ranges: low and high.

	Muons		Electrons		$t\bar{t}$+PU140		$t\bar{t}$+PU200	
	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T$	$3 \leq p_T \leq 8$	$8 \leq p_T$
ϵ^{SR}	99.9 ± 0.05	$100.0_{-0.05}$	94.9 ± 0.1	98.3 ± 0.05	97.2 ± 0.05	97.8 ± 0.1	97.2 ± 0.05	97.5 ± 0.2
ϵ^{TW}	$100.0_{-0.05}$	$100.0_{-0.05}$	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05
ϵ^{AM}	97.0 ± 0.05	98.4 ± 0.05	76.5 ± 0.1	92.5 ± 0.1	95.5 ± 0.05	98.4 ± 0.1	95.7 ± 0.05	98.5 ± 0.1

Table 6.1: Efficiencies, up to the pattern recognition stage, for particles with $p_T \geq 3$GeV/c

Figures 6.17 and 6.18 show the evolution of the efficiency in the low p_T regime for muons and electrons respectively. The black dots represent ϵ^{SR} and the blue circles the total efficiency $\epsilon^{SR \to AM}$, which as we have $\epsilon^{ST} = \epsilon^{TW} \approx 1$ is equivalent to $\epsilon^{SR} \cdot \epsilon^{AM}$.

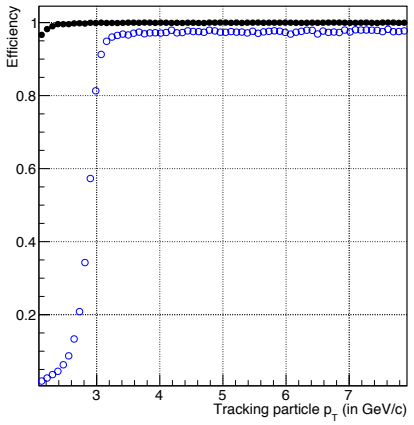


Figure 6.17: Efficiency for primary muons without PU

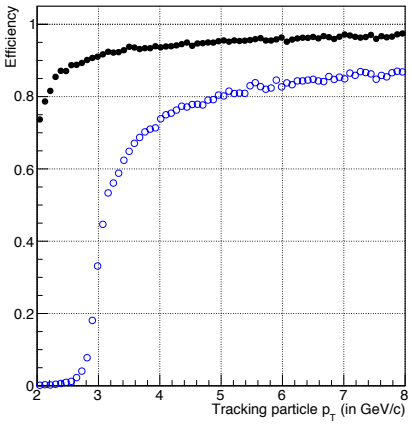


Figure 6.18: Efficiency for primary electrons without PU

As expected the muon efficiency is slightly better, especially at low p_T. Moreover the turn-on at 3 GeV/c is very sharp, with almost null efficiency at 2 GeV/c. This is an important point because otherwise a significant number of low p_T tracks would pass the identification step, dangerously increasing the system occupancy for the next steps.

As they were used to build the banks, the muon performance without pileup is the optimal result. We see that for events containing high PU efficiency is comparable to that of muons. This confirms the fact that pile up has little impact on this parameter. A pattern activated by a single track will also be activated if PU effects are superimposed.

The fact that the efficiency plateau for muons is below 99% is due to the definition of banks in the endcap sectors. Historically flat geometry banks were based on trigger towers extended up to $|\eta| = 2.3$, instead of 2.4. This problem has been corrected later in the tilted geometry, but in the results presented here we still observe a significant drop in

efficiency beyond $|\eta| = 2.3$ which impacts the general level of the plateau.

Performance is less good for the electrons, in particular at low p_T. This is not really a surprise, the trajectories of electrons being much more prone to interact with the detectors. We see that even the acceptance efficiency is affected. To overcome this problem we have seen in section 3.3.2 that we can intervene at the level of ϵ^{SR}, for example by widening the stub windows (at the cost of an increase of stubs multiplicity). It is also possible to improve ϵ^{AM} by using pattern banks generated using electrons.

6.3.3.2/ FAKE RATES AND MATCHED ROAD MULTIPLICITIES

The average road rates for $PU + t\bar{t}$ events are presented on Table 6.2, along with the rates satisfied by 95% of the events. Fake proportion are also provided. Results are presented for the 3 different tower types.

	Average road rate		Fake proportion (in %)	
Average PU	140	200	140	200
Barrel tower	11.0 (41)	19.6 (62)	24.4	52.4
Hybrid tower	18.4 (63)	39.2 (109)	54.6	76.4
Endcap tower	16.0 (54)	29.2 (84)	46.6	67.8

Table 6.2: Matched road rates and fake road proportion after AM for $t\bar{t}$ events. 95% rate limits are given in brackets

As we use an activation threshold of 4 stubs in certain areas of the hybrid towers and, to a lesser extent, in the towers of the endcaps, the fake rate in these areas is larger than in the barrel where the threshold is always at 5.

Pileup dependency is also stronger in sectors containing disk modules. This is because the resolution of low p_T stubs is much worse in these areas (we saw that in chapter 3). It is therefore much more likely to activate a pattern for a low p_T track in these towers. We will see later that these low-p_T primary tracks actually contribute to a significant proportion of the fake roads.

Regarding the output rate, we see that it is less than 110 roads per sector per event for 95% of the events, regardless of the pileup value. This is consistent with the goal of 100 that was set at the beginning of this optimisation work. Dynamic losses due to too many activated patterns would be under control with such a configuration.

6.3.3.3/ STUB PURITY AFTER THE AM STAGE

The impact of the AM filtering on the stub content is significant, as shown on Table 6.3. Once again, the filtering power is better in the tower containing barrel layers, due to the poor stub resolution in the disks. However, in all the cases, the purity of the stub content after AM is reaching a slightly better level than before the pattern recognition stage, where it was below 5 %.

The number of stubs to be processed later is reduced by an order of magnitude for an average pileup of 140. The reduction is less important in the case of 200, but still significant.

	Stub multiplicity		Good stub proportion (in %)	
Average PU	140	200	140	200
Barrel tower	27.7	65.0	43.6	23.3
Hybrid tower	33.6	87.0	32.0	15.6
Endcap tower	28.2	63.8	32.7	18.8

Table 6.3: Average stub rates per trigger tower after the AM stage. A good stub is a stub induced by a particle with $|d_0| < 1cm$ and $p_T > 3GeV/c$

However, if the efficiency of the AM method is not pileup dependent, it is not surprising that the output multiplicities seem slightly more affected by this parameter.

6.3.4/ SUMMARY

The potential of associative memories in CMS has been evaluated thanks to the emulation presented in this chapter. This study allowed us to confirm the interest of such an approach, and served as a basis for the development of a more complete procedure.

However, these results reflect only a small part of the totality of the studies carried out, and only a fraction of the enormous potential of the associative memories has been exploited. The phase space is very broad, and the room for improvement still very large.

Some ideas that have been explored and not mentioned in this chapter:

- **Banks sorted by** p_T: one of the big advantages of AM chips is the ability to extract activated patterns in a predefined order. In the previous study, roads were ranked by popularity, that is, by the number of particles corresponding to the road when the bank was created. With such an order, it is possible to considerably reduce the size of the bank by removing the less frequent roads. The impact on efficiency is relatively low. However, one could also imagine ordering the patterns by p_T in order to first extract the patterns compatible with a interesting high transverse momentum track, or to define a hybrid scenario optimising the efficiency with a reduced bank size. The first studies in this direction were presented [48, 49].

- **Banks for Electrons**: the idea here is to build dedicated banks using high p_T electrons in order to reduce inefficiency due to multiple scattering. The patterns activated via these banks can be marked as electron candidates and can thus be directed to an adapted fitting unit, with more flexible selection criteria. This solution has not been thoroughly tested yet.

- **Using the stub width**: the use of this information for reconstruction has been the subject of much debate. In the AM approach, it was decided not to rely on this parameter, whose medium-term robustness has not been fully demonstrated (eg how to deal with a dead strip). The other approaches made a different choice. In order to fairly compare the results of the AM approach with others, it was decided to test the impact of this information on AM filtering. These studies showed that if it was not necessarily very interesting to use the width in the chip itself [50, 51], its use to filter the stubs in the matched roads could prove very efficient [52, 53]).

Again, these examples are just the tip of the iceberg. However, they clearly demonstrate the richness and flexibility of this approach.

7

TOWARDS A FULL DEMONSTRATOR: THE FASTTRACK PROJECT

We have just demonstrated the feasibility of an approach based on associative memories in CMS. The logical continuation of this work is the implementation of a complete algorithm, and its deployment in a demonstrator. These are the steps presented in this last chapter.

7.1/ INTRODUCTION

The results of the previous chapter have allowed us to initiate in 2013 a collaboration with the ATLAS group of LPNHE, a group involved in the FTK project. This association gained momentum in 2014 with the start of the FastTrack project, which was granted between 2014 and 2018 by the National Research Agency (Project ANR-13-BS05-0011 [54]) .

This project was divided into two complementary parts: development of a new generation of AM chips in a 28 nm CMOS technology (LPNHE), and creation of a demonstrator using these chips in the framework of CMS (IP2I).

In CMS, FastTrack was part of a larger collaboration, particularly with INFN and Fermilab, whose goal was to develop a complete demonstrator of the entire process. We will discuss this point at the end of the chapter.

In parallel with the development of these demonstrators, a complete track reconstruction algorithm has been developed in order to complement the AM-based filtering step. We will first focus on this work, and more particularly on an intermediate filtering module developed at IP2I.

7.2/ FILTER THE MATCHED ROADS: THE TRACK CANDIDATE BUILDER

7.2.1/ PRINCIPLE

One of the most promising methods for adjusting track parameters is Principal Component Analysis (PCA) [55]. This technique provides a linear relationship between the track parameters and its coordinates. Besides the fact that it leads to the optimal resolution on these parameters, this method has the immense advantage of being very simple to implement on an FPGA. Indeed, the parameters are obtained from the coordinates of the stubs via a single scalar product. It is extremely fast and efficient. Added benefit, this is a proven technique, which has been successfully implemented in CDF [56]

However, this technique is very sensitive to impurities. It is important to provide as input the cleanest possible content, and in particular exactly 1 stub per layer. This condition was satisfied by construction in the case of CDF, in particular because the patterns were very fine. The situation of CMS is rather different.

The roads being relatively wide, they can contain up to several stubs per layer. For a PCA processor, it means testing all possible stub combinations. For patterns matched in high p_T jets, the number of combinations to be tested can quickly explode.

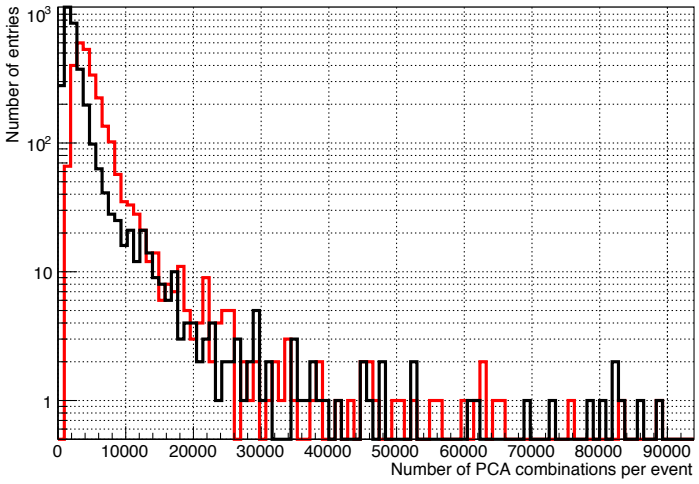


Figure 7.1: Number of stub combinations per event to feed to the PCA processor, for $PU140 + t\bar{t}$ events (black line) and $PU200 + t\bar{t}$ events (red line)

As an example, Figure 7.1 shows the total number of combinations after AM filtering, for standard HL-LHC events. This number is very important, and if all combinations must go through a PCA processor, even with multiple processors running in parallel, this can become problematic for total latency.

It is therefore important to limit the number of combinations to be tested at the output of associative memories. This is the role of the track candidate (TC) builder.

7.2.2/ ALGORITHM PRINCIPLE

The TC builder is an algorithm that must be implemented on an FPGA. It should therefore be as simple as possible. It consists of a predefined number of filter modules operating in parallel: the TC modules.

The operating principle of any of these modules is explained in Figure 7.2.

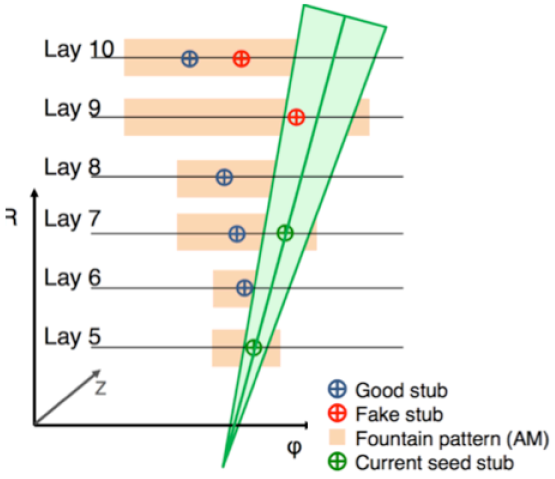


Figure 7.2: TC builder principle

Each TC module processes the stubs of one pattern at a time. The idea is to start from a seed of 2 stubs, and to project this seed to the next layer. If we find a stub in the projected area we continue the projection, and so on. In order to simplify the algorithm as much as possible, the projection is always done from the seed. If the algorithm ends successfully, we obtain a clean track candidate containing exactly one stub per layer, which can be sent to the PCA module.

We will now briefly describe each step of this algorithm.

7.2.2.1/ SEED GENERATION

A seed is a pair of stubs belonging to two distinct PS layer/rings. All the seeds are treated sequentially. The choice of the seeding layers is driven by two main constraints: resolution and multiplicity.

In order to get the best possible accuracy for the 3D projection, only the layers containing PS modules are used to build seeds. In the disks, this means that only the innermost rings are used. This choice also allows to satisfy the second constraint, as using less seeding layers will naturally reduce the number of seeds.

Seeding layers are also depending on the trigger tower type. The current repartition is the following:

- **Barrel:** the 3 innermost barrel layers.
- **Hybrid:** the 3 innermost barrel layers.

- **Endcap:** the 2 innermost barrel layers and the 9 innermost rings of the two lowest $|z|$ disks.

This repartition ensures that every track crosses at least three seeding layers, and consequently induces at least one seed, even if one stub is missed.

This also reduces the number of seed pairs to be tested for each roads, and therefore enables to keep the algorithm within a reasonable overall latency. The total number of seed pairs to test is fixed beforehand using dedicated calibration procedure described later in this section.

This definition can be extended to the tilted geometry by dividing the three inner barrel layers in flat and tilted parts, which will be considered as distinct seeding layers. There is therefore 3 more seeding layers in the tilted geometry.

7.2.2.2/ 3D PROJECTION AND FILTERING

This step is a key element of the algorithm. The projections are done in the (r, ϕ) and (r, z) planes, where a primary track trajectory can be considered as linear in first approximation. One can indeed linearise the trajectory in the (r, ϕ) plane. Starting from primary particle circular trajectory, it's straightforward to derive the following relation:

$$r = 2R_c \sin(\phi - \phi_0), \tag{7.1}$$

leading to

$$\phi = \phi_0 + \arcsin \frac{r}{2R_c}, \tag{7.2}$$

where R_c is the curvature radius, equal to 1.75m for 2GeV/c and 2.63m for 3GeV/c track. If $\frac{r}{2R_c}$ is small enough, one can use the following approximation: leading to

$$\phi = \phi_0 + \frac{r}{2R_c}, \tag{7.3}$$

The error never exceeds 9mrad at 2GeV/c, and 2mrad at 3GeV/c, which is acceptable for the extrapolation windows. In addition, the radial distance between the seed and the extrapolation point will always be below 1m, error made will therefore be slightly smaller.

The method used to select track candidates is sketched on Fig. 7.3 for the (r, ϕ) plane. The principle is the same in the (r, z) plane. The seeding stubs are denoted A and B on the figure. If the stub contained on the projection layer, denoted as C belongs to the particle trajectory, it can be easily shown that surfaces S_1 and S_2 are identical. Therefore, the parameter Δ_ϕ given by $S_1 - S_2$ should be compatible with 0 for potentially good stubs. In the same way, a parameter Δ_z can also be computed.

The calculation of Δ_ϕ and Δ_z, as it does not involve any division, is particularly interesting for an FPGA implementation.

In order to be selected, a stub has to be within both windows (windows are precomputed using monte-carlo single particle events). Then, for a given layer, if there is more than one stub within the windows, the one with the lowest $|\Delta_\phi|$ value is selected (Δ_z resolution being much worse, it is just used as a control).

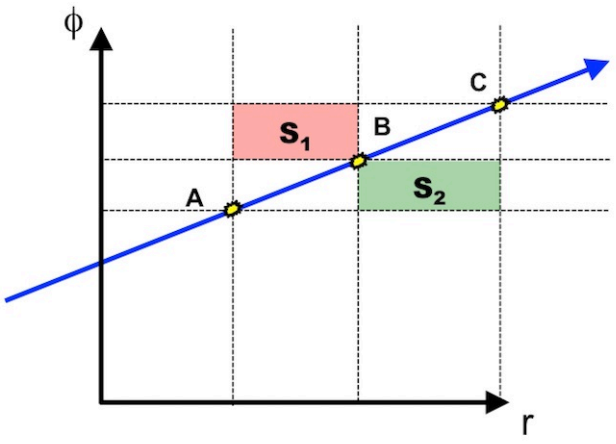


Figure 7.3: Stubs selection at projection stage in the TC module

7.2.2.3/ TRACK CANDIDATE RECORDING

At the end of the extrapolation procedure, only the track candidates with no more than one missing stub are kept. This corresponds to 5 stubs in the barrel and endcap towers, and 4 stubs in the hybrid tower. For a given matched road, only the best TC is kept. Priority is given to the TC without any stubs missing. Then, if one stub is missing, the TC with at least 3 stubs in the PS modules are kept first.

The process stop as soon as a good TC (no stubs missed) has been found.

7.2.2.4/ EXTRACTING THE EXTRAPOLATION WINDOWS

The calibration step is done using clean single particle events (muons or electrons). For each pair of seeding layers, a dedicated macro computes, for each extrapolated layer, the optimal window width. Fig. 7.4 shows for example the $|\Delta_\phi|$ values for barrel 2S layer 1, based on seeds built with the 2 innermost barrel layers. For muons and electrons.

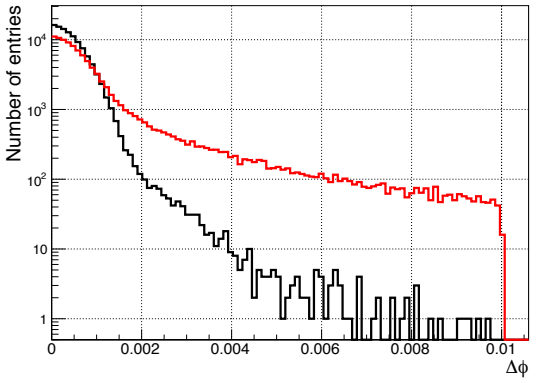


Figure 7.4: Δ_ϕ values for particle gun muons (black) and electrons (red). Seeding layers are the two barrel innermost rings, projection layer is barrel fourth ring.

As expected the residuals are roughly gaussian for the muons, with slightly larger residual

tails for the electrons. For the muons, one can easily tune a tight extrapolation window as the value for which only 1% of the residuals are rejected. For electrons one can see that a slightly looser window should be defined. As for the stub windows tuning, the TC windows tuning is a trade-off between efficiency and purity.

7.2.3/ PERFORMANCE OF THE COMPLETE ALGORITHM

7.2.3.1/ FOREWORD

The study of which we present here is a continuation of the one introduced in the previous chapter. The same samples were used, but this time they were passed trough a complete track reconstruction algorithm. In addition to the TC filtering described above, we added algorithms for adjusting parameters (based on PCA) and deleting duplicates (a track is usually contained in several patterns). These algorithms have been developed by other groups, and a detailed description can be found in the following documents [57, 58].

7.2.3.2/ DEFINITIONS

As explained earlier, the TC builder windows were determined using a dedicated sample of particle gun muons. The values of Δ_ϕ and Δ_z for which 98% of the stubs induced per muons with $p_T \geq 3GeV/c$ are kept were defined as windows for the TC builder implementation in this setup. No particular optimisation was done for electron in the flat geometry setup, therefore only one set of TC windows was used.

For the definition of efficiency, we add three additional terms to the formula defined in the previous chapter:

$$\epsilon = \epsilon^{SR \to AM} \cdot \epsilon^{CB} \cdot \epsilon^{TF} \cdot \epsilon^{DR} \tag{7.4}$$

where:

- $\epsilon^{SR \to AM}$: efficiency up to the AM filtering stage, introduced in the previous chapter.
- ϵ^{CB}: combination builder efficiency. Proportion of tracks after AM which are containing at least 4 stubs in four distinct layers/disks in at least one track combination.
- ϵ^{TF}: track fitting efficiency. Proportion of tracks after CB which are containing at least 4 stubs in four distinct layers/disks in at least one fitted track.
- ϵ^{DR}: duplicate removal efficiency. Proportion of tracks after TF which are containing at least 4 stubs in four distinct layers/disks in at least one L1 track.

Fake rate definition is slightly different from that used for the AM part. Indeed, if an activated pattern can contain a certain number of stubs not belonging to the reconstructed track, this is not the case for a TC or a final L1 track, whose purity must be much better.

A TC/L1 track containing N stubs is matched to a particle if more than $N - 1$ stubs of the TC/L1 track belongs to this particle. Based on that assumption, a TC/L1 track is considered as a fake if:

- It is not matched to any particle.
- It is matched to a particle which is not fulfilling our requirements ($p_T < 3GeV/c, ...$).
- It contains more than one fake stub.

It should be noted that this fake definition is slightly more restrictive than a definition based on track parameters comparison (which are not accessible after the TC builder). Indeed, with this definition, a TC containing a real track not fulfilling our track requirements will be considered as a fake. The values provided in this note should therefore be considered as pessimistic estimates.

7.2.3.3/ RESULTS

The table 7.1 presents the results obtained, on the same basis as the table in the previous chapter, including this time the value of the total efficiency.

	Muons		Electrons		$t\bar{t}$+PU140		$t\bar{t}$+PU200	
	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T$	$3 \leq p_T \leq 8$	$8 \leq p_T$
ϵ^{CB}	99.0 ± 0.05	99.1 ± 0.05	93.6 ± 0.1	94.6 ± 0.1	97.9 ± 0.05	98.8 ± 0.1	97.4 ± 0.05	98.8 ± 0.1
ϵ^{TF+DR}	99.9 ± 0.05	99.9 ± 0.05	99.8 ± 0.1	99.3 ± 0.1	99.5 ± 0.05	99.5 ± 0.1	99.3 ± 0.05	99.4 ± 0.1
ϵ	96.0 ± 0.05	97.4 ± 0.05	67.8 ± 0.1	85.4 ± 0.1	90.5 ± 0.1	94.5 ± 0.2	89.9 ± 0.1	94.3 ± 0.2

Table 7.1: Efficiencies up to the final L1 track, for particles with $p_T \geq 3$GeV/c

We see that the efficiencies obtained with the filtering algorithm are very good, even for electrons. This is not totally surprising because the electrons whose trajectories are strongly deviated are already removed at the level of associative memories. We also see that this efficiency is relatively insensitive to pileup.

The Figure 7.5 illustrates this point in a remarkable way. It shows the proportion of stubs rejected by the different filtering steps, as a function of the multiplicity in stubs of the event, a parameter directly related to the average pileup multiplicity.

It is clear that the filtering power of associative memories decreases with PU. On the contrary, the impact of PU on the TC module is much less important, and it keeps a very good rejection power even in extreme cases.

As expected also, the efficiency of the fit and duplicate removal is very close to 1. This is due to the operating principle of these algorithms, but also to the fact that the quality of the information provided by the filter module is very good.

This point is illustrated in Fig.7.6, where the ratio between the number of combinations and the number of track candidates per event is represented for the 2 PU values considered. It can be seen that the number of track candidates is approximately one order of magnitude smaller than the number of combinations and is less sensitive to PU.

The limitation to a maximum of one TC per matched road clearly helps there, but the very good filtering power of the TC builder plays also a role. Indeed, as shown on Fig.7.7, the TC builder often discards more than 50% of the matched roads, and this proportion improves with PU, without any impact on the efficiency, as we have already seen previously.

TC rates, summarised in Table 7.2, tends to confirm this observation. The rates are significantly lower than in Table 6.2, and the fake proportions confirms that most of the

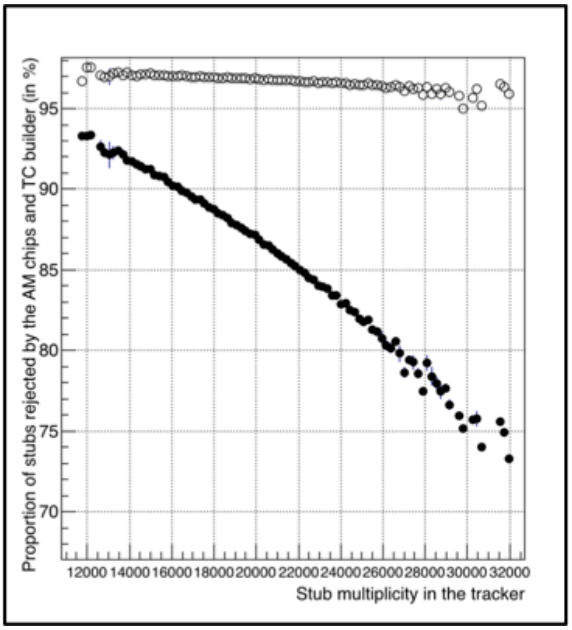


Figure 7.5: Proportion of stubs filtered according to the total number of stubs in the event after associative memories (black circles) and TC builder (white circles)

.

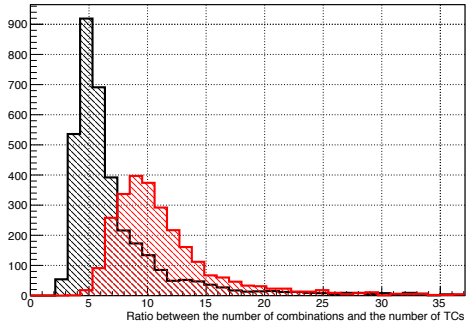


Figure 7.6: Number of stub combinations over the number of TC per event for $PU140 + t\bar{t}$ (black) and $PU200 + t\bar{t}$ (red)

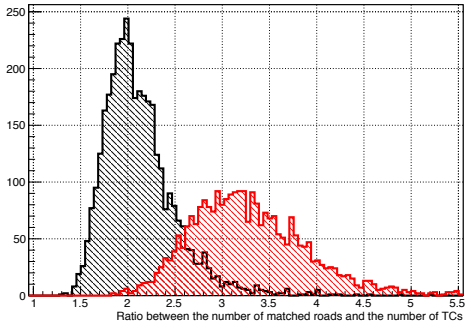


Figure 7.7: Number of matched roads over the number of TC per event for $PU140 + t\bar{t}$ (black) and $PU200 + t\bar{t}$ (red)

removed roads were effectively fakes. The 95% limits are not very different among the different towers. In all the cases, the PU dependence observed after AM gets canceled out after the TC stage.

	TC rate		Fake proportion (in %)		Fake proportion with 1.5 GeV/c cut (in %)	
Average PU	140	200	140	200	140	200
Barrel tower	6.0 (24)	7.4 (26)	5.2	8.6	1.8	4.7
Hybrid tower	8.2 (33)	10.3 (37)	9.9	16.9	3.2	9.5
Endcap tower	6.9 (28)	8.9 (31)	16.1	21.1	3.1	6.3

Table 7.2: TC rates and fake TC proportion. 95% limits are provided into brackets.

The two last column of the Table show that most of the remaining fake TCs are indeed primary tracks with a p_T between 1.5 and 3 GeV/c. The remainder, which can be consid-

ered as the real fake content, is below 4% at PU140, and below 9% at PU200. As we will see later this is very close to the final purity of the L1 tracks.

The 95% limits for the TC multiplicities per tower are recalled in Table 7.3, and compared with the corresponding combinations multiplicities, is the PCA was run right after the AM. Those values confirms the advantage of using the TC builder for the PCA processing time and input quality. This also confirm the lower PU dependence observed on Fig. 7.6.

	TC		Raw combinations	
Average PU	140	200	140	200
Barrel tower	24	26	129	261
Hybrid tower	33	37	195	389
Endcap tower	28	31	151	267

Table 7.3: Comparison of the 95% limits for track candidates and raw number of combinations without TC builder

Finally, track rates and purity after the duplicate removal and PCA stages are provided in Table 7.4. As one can see, fake rates are relatively stable with respect to the PCA outcome, main difference is due to the fact that most of the duplicate tracks are removed. Final L1 track rates are relatively small, as are the tails, as one can see on Fig. 7.8. As for the number of combinations, the tails are due to high-pT jets more than PU average. The main advantage of using the TC builder clearly appears on this plots. Indeed, by ensuring a very low fake rate, one controls the quality and size to the data stream sent to the L1 trigger system.

	L1 track rate		Fake proportion (in %)		Fake proportion with 2GeV/c cut (in %)	
Average PU	140	200	140	200	140	200
Barrel tower	1.2 (5)	1.5 (5)	5.7	9.1	1.6	4.3
Hybrid tower	1.2 (4)	1.6 (5)	10.5	17.9	3.2	9.4
Endcap tower	1.0 (4)	1.3 (4)	11.6	16.2	2.9	5.8

Table 7.4: L1 track rates and fake proportion. 95% limits are provided into brackets.

To conclude, Fig.7.9 illustrates the performance of the procedure on a complex PU200+4tops events. The left picture shows the stubs contained in the disks for this event. Middle figure shows the stubs remaining after the AM stage. Tracks start to appear clearly, but some noise remains in the busiest areas. The right figure shows the remaining content after the TC building stage. Noisy areas are now clean.

The complete algorithm works perfectly in simulation, remains to prove that it also works on the demonstrator.

7.3/ DEMONSTRATOR

7.3.1/ GENERAL DESCRIPTION

The pocket demonstrator that was set up as part of the FastTrack project is sketched on the Figure 7.10.

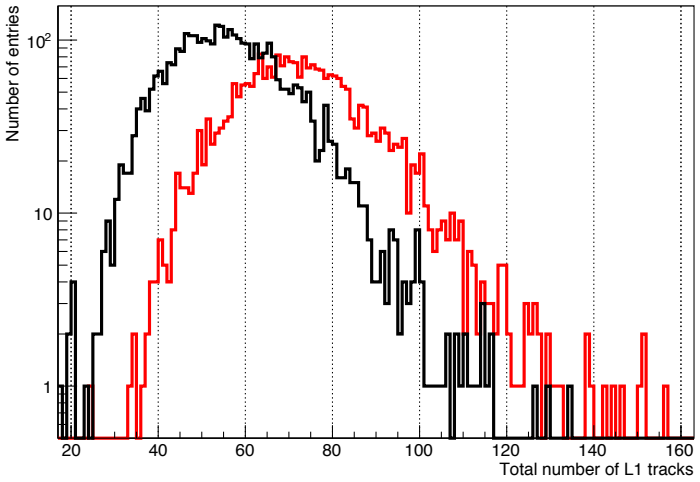


Figure 7.8: Final number of L1 tracks per events, for PU140+ttbar events (black line) and PU200+4ttbar events (red line)

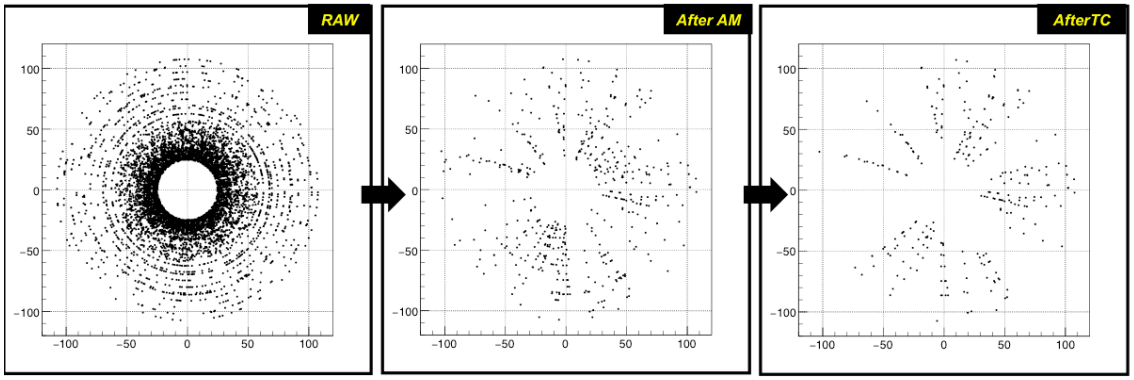


Figure 7.9: AM filtering and TC builder in action on a complex PU200+4tops event.

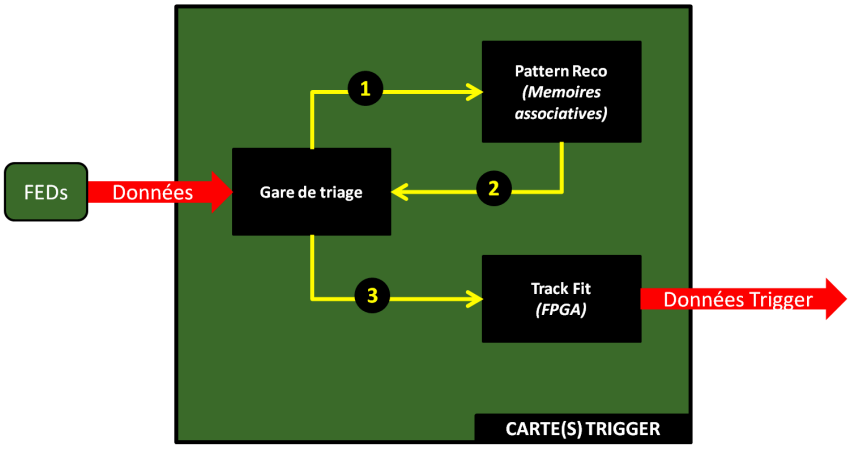


Figure 7.10: The FastTrack demonstrator principle

It is not a complete system because only the track reconstruction was considered. We therefore start from a system in which time and space multiplexing have already been done. This means that the data that enters the demonstrator (box named FEDs in Figure 7.10) corresponds to the data of a trigger tower, received every microsecond.

Incoming stubs are received in a Data Organizer (Gare de Triage) unit where they are stored and converted into superstrips. This is also the point where the stub addresses are converted to global coordinates in the tracker reference frame. It is a complex and very important step, which must be carried out as quickly and precisely as possible. We will come back to this in more detail in the section dedicated to the description of the test bench firmware.

Superstrips, once created, are sent to the AM chip for identification. As soon as the last superstrip has entered the chip, the activated patterns return to the Data Organizer where the coordinates of the stubs contained in these patterns are retrieved and transmitted to the next step. The track candidates thus created are then extracted from the demonstrator to be analised.

With this demonstrator, our goal was to confirm the feasibility of the method developed and tested by simulation. Then, we were able to provide a first estimate of the total latency of the procedure, which corresponds to the time between the arrival of the first stub and the output of the last track, knowing that this measure would have to be amended.

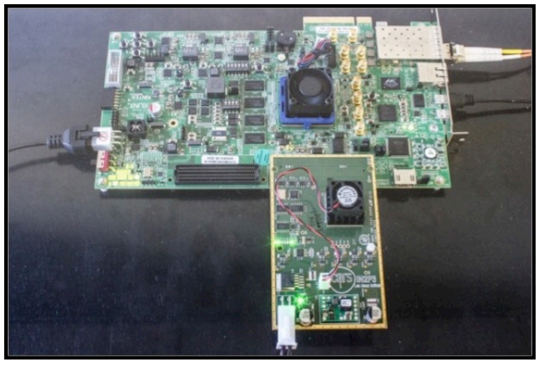


Figure 7.11: FastTrack test-bench

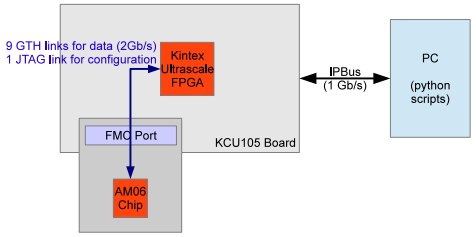


Figure 7.12: Test-bench working principle

The test-bench that has actually been developed is pictured in Figure 7.11, its principle being described in Figure 7.12. It is based on a Xilinx KCU105 evaluation board hosting a Kintex UltraScale FPGA. One of the FMC connectors of the board is occupied by a FMC mezzanine developed at the IP2I and hosting an AM06 chip. A PC is connected to the KCU105 via an optical Ethernet connection at 1 Gb/s. Communications are established using Python scripts and the IPBus [59] protocol.

The AM06 chip [60] used was designed for ATLAS FTK system. This is an ASIC designed in 65nm CMOS technology containing 2^{17} patterns ($\tilde{1}30000$), consisting of 8 superstrips each. It integrates 8 input lines at 2 Gb/s and a line at 2 Gb/s for the output. The nominal frequency of the chip is 100 MHz but it can be overclocked to 125 MHz. Minimising latency was clearly not the main goal for this chip. It was therefore not expected to meet our needs in this sector but remained very interesting for our tests because it was including all the necessary features and an already high pattern density.

7.3.2/ ALGORITHM IMPLEMENTATION

The block diagram of the firmware implemented on the FPGA is shown in Figure 7.13. Just like the CIC model, it is divided into sub-blocks (surrounded by dashes) each with a specific role.

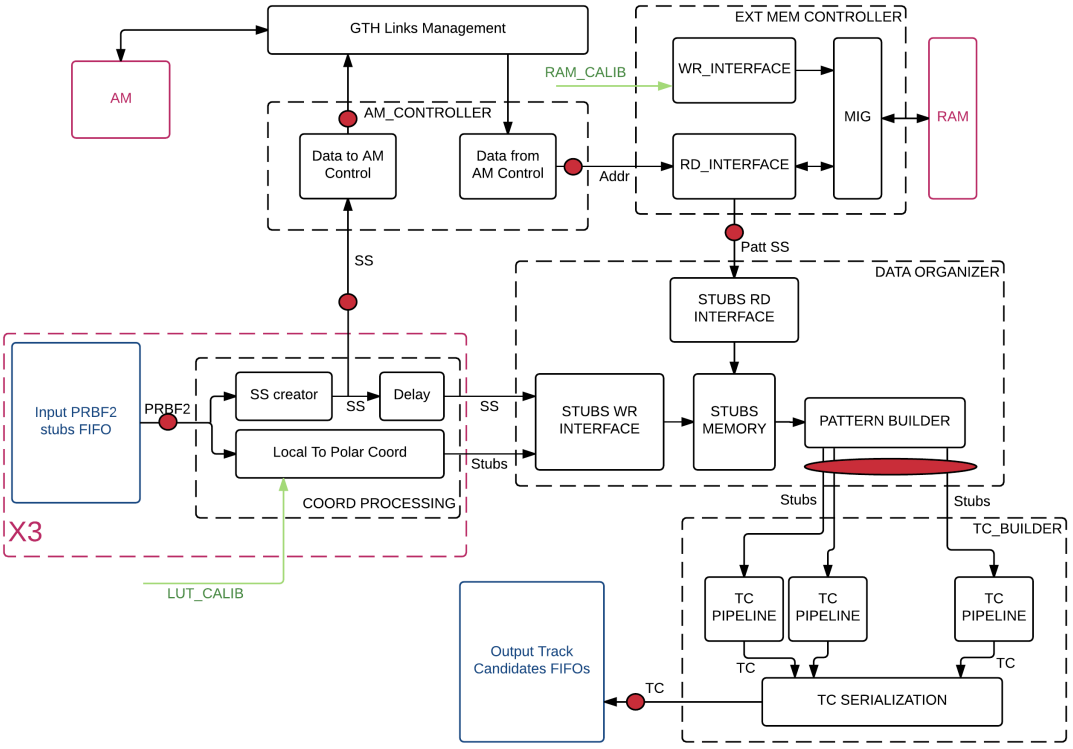


Figure 7.13: Demonstrator firmware block diagram

The local coordinates of the stubs (position in the module, output format from the CIC) arrive first in the block *COORD PROCESSING* where they are converted simultaneously into 2 different formats: superstrips and global coordinates in the tracker reference system. Superstrips are sent immediately to the *AM CONTROLLER* block that interfaces the firmware and the AM chip. The global coordinates are sent later to the *DATA ORGANIZER* block.

From the moment the superstrips start to enter the AM chip we can start to recover activated patterns at the output. These addresses are immediately transmitted to the *EXT MEM CONTROLLER* block, which retrieves the list of pattern superstrips (via a hash table pre-stored in the RAM of the evaluation board). The patterns are then passed to *DATA ORGANIZER* as superstrips lists.

The *DATA ORGANIZER* has a fundamental role because it allows to match the stubs and activated patterns. For each pattern, the global coordinates of the stubs contained in the superstrips are retrieved. Eventually, for each pattern, we end up with the list of corresponding stubs, in the form of global (r, z, ϕ) coordinates. We can send them to the *TC BUILDER*.

The latter corresponds to the physical implementation of the algorithm described above. A more detailed description of this implementation can be found in [61]. The number of TC modules operating in parallel depends on the FPGA used. Our test bench contained

for example 12 such modules.

The reconstructed track candidates are then sent to the analysis program in order to be compared with the expected results obtained from the simulation.

7.3.3/ PERFORMANCE AND DATA/SIMULATION COMPARISONS

In order to compare the results obtained with the TC builder, a bitwise implementation of the algorithm has been developed. Instead of using floating numbers, we use, as in the FPGA, integers encoded on a given number of bits. Preliminary studies have shown us that the best compromise to minimise the false TC rate was to encode the coordinates on 18 bits: 1 bit for the sign, 7 bits for the integer part, and 10 bits for the fractional part. With this encoding, the fake rate for complex events goes from 1.8 % for the floating point version, to 2.3 %, which remains acceptable.

Of course, encoding the numbers is not sufficient to emulate the behaviour of the FPGA implementation, it is also necessary to perform the operations (additions and multiplications) in the same way. By doing so, the developed code must reproduce the TC builder behaviour to the nearest bit. This is actually what was observed for our system.

We had previously been able to verify that AM emulation also has bitwise accuracy. So we had a bitwise emulation of the complete chain. In practice, this means that for simple events without PU, the emulator and the test bench provided exactly the same matched patterns and corresponding track candidates. For more complex events small differences were observed, but these were expected and came from limitations of the system (truncation effects).

In the same way we measured the total latency of the procedure, starting from a latency model of the TC Builder which we validated beforehand (the latency of the other blocks is simple to estimate). For this exercise, we used a sample of 934 $t\bar{t}$ events. On average, we measured 420 clock cycles between reading the first stub and reading the last track candidate. Figure 7.14 gives details of the average time spent at each step. We note that out of 420 clock cycles, 228 are linked to the AM06 chip and could therefore be greatly improved with a new generation of chips. The remaining clock cycles (outside the AM chip) are less than 200. Fig. 7.15 gives the distribution of latency for all events. Since the frequency of the FPGA is 125 MHz for our tests (to match the frequency of the AM chip), we see that the total latency is less than 4.2μ s for 95% of the events.

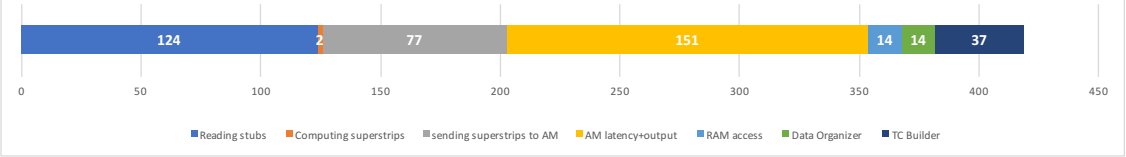


Figure 7.14: Average clock cycles spent in the various steps of the firmware for PU140 $t\bar{t}$ events

If we consider that a microsecond is necessary at the adjustment stage, we can extrapolate the total latency of our demonstrator to about 5μ s. This is a quite honourable result, especially considering that the AM06 chip used here was not optimised to work in reduced latencies.

This important result enables to demonstrate that with a faster ASIC, an approach based

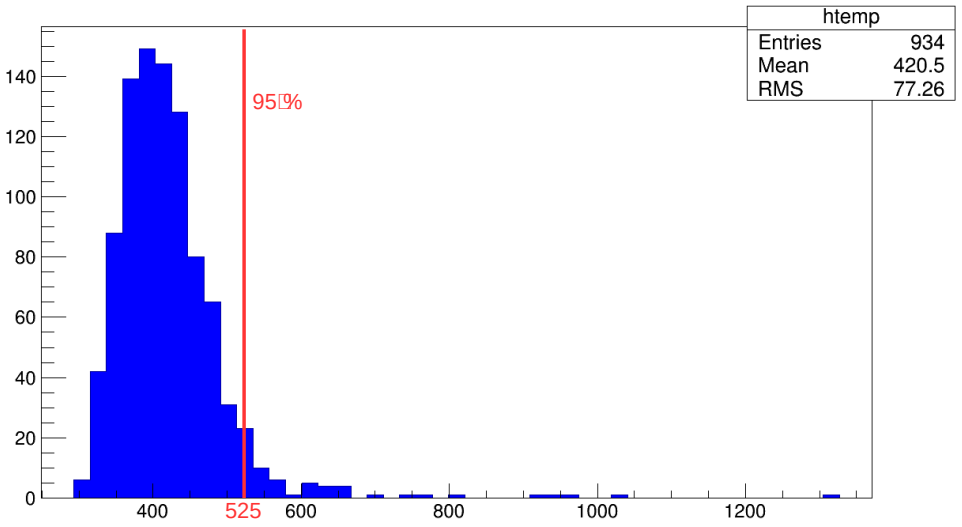


Figure 7.15: Distribution of clock cycles spent to treat PU140 $t\bar{t}$ events, 95% limit in red.

on associative memories in CMS was within reach. This observation was reinforced with the development of a more complete system, including the data distribution stage.

7.4/ TOWARDS A COMPLETE SYSTEM

Our test bench, which was using only a single AM chip, served as a basis for a more complete system developed by INFN. This setup included a firmware almost identical to that of our demonstrator, but it used a mezzanine containing 12 chips, allowing to model an entire trigger sector. This system was successfully completed and tested in 2017 [62], thus demonstrating that our test bench could be extended to a complete trigger sector.

The critical point of these systems was clearly the fact that ASIC AM06 was not suited to our problem. However, the development of faster ASICs was one of the deliverables of the FastTrack project. This milestone was however reached in 2017 with the characterisation of the AM07 chip, which was one of the first ASICs developed in the field of high energy physics in the 28 nm CMOS technology node [63]. This success paved the way for the development of more complete versions (AM08 then AM09 [64]), whose development is currently in progress for ATLAS HL-LHC track trigger upgrade.

Concerning the data distribution, the operation of the system described in chapter 5 was set up successfully in a dedicated demonstrator in 2016 [65].

7.5/ CONCLUDING REMARKS

From 2012 to 2017 a huge amount of work was conducted by the AM collaboration. In 2017, all the puzzle bits were gathered to develop a complete system adapted to CMS. Given the complexity of the project, it was a major result. Through our emulator, then via the FastTrack project, we have contributed significantly to these developments and to their success. The feasibility of the AM approach was demonstrated, it is a fact, but the approach did not survive the conclusions of the end-of-2016 project review, which concluded in favour of a approach purely based on FPGAs.

It was difficult, if not impossible, to continue without the support of CMS. Some of the groups went on to join the ranks of the FPGA approaches, others, like us, decided to stop there.

From a personal point of view, I did not wish to continue the adventure, for several reasons. First, the CIC project required a full-time investment in 2017. As a project manager, it would have been difficult for me to tackle these two major projects together. The concentrator being on the critical path, it became de facto a priority. The other reason is more personal; when you have invested several years in a project and you are convinced that it was the best, it is difficult to go on with persons who invested several years telling the opposite...

In particular, I found it questionable that the data distribution system we were proposing was not retained. It was indeed fully compatible with other approaches, and it was the only one whose operation had been demonstrated in 2016. Instead of that, it was decided to develop another system, nonexistent at the time. I am not naive and I know that other factors must be taken into account in this type of situation, but the arguments used were far from convincing me.

Locally, even if the project did not reach the expected outcome, the good results obtained had a rather positive impact. The engineer hired on a fixed-term contract via the ANR grant got a permanent position at IP2I. As for the engineer (IE) with whom I started this activity in 2012, his role as technical project manager, which he has insured with great skill, allowed him to benefit from a promotion to research engineer grade in 2017.

As a project manager, these points are important and show that such entreprise can bring a lot even if it does not end as one would have wished.

V

CONCLUSION

CONCLUSION

I have tried to trace through this document most of my activities over the last fifteen years. It all started with the alignment of a VELO in LHCb, and ends today with a stub extractor in CMS. Beyond those activities, it's also my job that has evolved. From a simple postdoctoral fellow in LHCb, I gradually became a project manager, first in ATLAS, and more concretely in CMS, where I enjoyed regular reviews and other critical paths.

I hope I have been able to transcribe the beauty and complexity of the future CMS tracker. If I do this job, it's to take part in those projects that seem to have been created for Mark Twain's famous sentence: *"They did not know it was impossible, so they did it"* .

I could personally verify this with the concentrator. If I had in 2013 the experience I have today in the field, I would never have taken over this project. And finally, at the time of writing, the first version of CIC is a success. So obviously, when you embark on this kind of adventure, there are not only pleasant moments, but when things finally succeed, we forget a lot of the difficulties encountered.

Even if the project we defend is not finally chosen, as was the case with the associative memories based track reconstruction, useful lessons can be learned. There is always something to gain, it is the message that must be remembered in the end.

BIBLIOGRAPHY

[1] L. Evans. *The Large Hadron Collider: a marvel of technology*. Fundamental Sciences. EPFL Press, Lausanne, 2009. https://cds.cern.ch/record/1193034.

[2] Georges Aad et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett.*, B716:1–29, 2012. https://arxiv.org/abs/1207.7214.

[3] Serguei Chatrchyan et al. Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett.*, B716:30–61, 2012. https://arxiv.org/abs/1207.7235.

[4] S. Viret. *Etude des désintégrations radiatives des mésons B dans le détecteur ATLAS*. PhD thesis, Université Joseph-Fourier - Grenoble I, 2004. https://tel.archives-ouvertes.fr/tel-00007066.

[5] The ATLAS Collaboration. ATLAS Level-1 Calorimeter Trigger: Timing Calibration with 2009 LHC Beam Splashes. Technical Report ATL-DAQ-PUB-2010-001, 2010. http://cds.cern.ch/record/1262858.

[6] The CMS Collaboration. Technical Proposal for a MIP Timing Detector in the CMS Experiment Phase 2 Upgrade. Technical Report CERN-LHCC-2017-027, 2017. https://cds.cern.ch/record/2296612.

[7] A. Augusto Alves, Jr. et al. The LHCb Detector at the LHC. *JINST*, 3:S08005, 2008. https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08005/pdf.

[8] R. Aaij et al. Performance of the LHCb Vertex Locator. *JINST*, 9:P09007, 2014. https://iopscience.iop.org/article/10.1088/1748-0221/9/09/P09007/pdf.

[9] S. Blusk, O. Buchmuller, A. Jacholkowski, T. Ruf, J. Schieck, and S. Viret, editors. *Proceedings, first LHC Detector Alignment Workshop*, number CERN-2007-004, 2007. http://cdsweb.cern.ch/record/970621.

[10] S. Viret, C. Parkes, and M. Gersabeck. Alignment procedure of the LHCb Vertex Detector. *Nucl. Instrum. Meth.*, A596:157–163, 2008. https://arxiv.org/abs/0807.5067 et Annexe A.1.

[11] M. Gersabeck, S. Viret, and C. Parkes. Performance of the LHCb Vertex Detector Alignment Algorithm determined with Beam Test Data. *Nucl. Instrum. Meth.*, A596:164–171, 2008. https://arxiv.org/abs/0807.5069 et AnnexeA.2.

[12] J M Amoraal. Alignment of the LHCb detector with Kalman filter fitted tracks. Technical Report LHCb-PROC-2009-026, 2009. http://cds.cern.ch/record/1205066.

[13] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008. https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003/pdf.

[14] G. Aad et al. Readiness of the ATLAS Tile Calorimeter for LHC collisions. *Eur. Phys. J.*, C70:1193–1236, 2010. https://arxiv.org/abs/1007.5423.

[15] E Starchenko et al. Cesium monitoring system for ATLAS Tile Hadron Calorimeter. In *Proceedings, 8th International Conference on Instrumentation for colliding beam physics (INSTR02): Novosibirsk, Russia*, 2002. https://cds.cern.ch/record/685349.

[16] J. Abdallah et al. The Laser calibration of the Atlas Tile Calorimeter during the LHC run 1. *JINST*, 11(10):T10005, 2016. https://arxiv.org/abs/1608.02791.

[17] K. Anderson et al. Design of the front-end analog electronics for the ATLAS tile calorimeter. *Nucl. Instrum. Meth.*, A551:469–476, 2005. http://inspirehep.net/record/699395.

[18] Ziad J Ajaltouni et al. The Tilecal Laser Monitoring. Technical Report ATL-TILECAL-94-039, 1994. https://cds.cern.ch/record/683494.

[19] M. Anfreville et al. Laser monitoring system for the CMS lead tungstate crystal calorimeter. *Nucl. Instrum. Meth.*, A594:292–320, 2008. https://cds.cern.ch/record/1073694.

[20] P. Gris. Upgrade of the Laser Calibration System of the Atlas Hadron Calorimeter. In *Proceedings, 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC 2015): San Diego, California, United States*, number ATL-TILECAL-PROC-2015-019. http://cds.cern.ch/record/2103219.

[21] L. Rossi and O. Brüning. Introduction to the HL-LHC Project. *Adv. Ser. Direct. High Energy Phys.*, 24:1–17, 2015. https://cds.cern.ch/record/2130736.

[22] M. Prydderch et al. CBC3: a CMS microstrip readout ASIC with logic for track-trigger modules at HL-LHC. Technical Report CMS-CR-2017-383, 2017. https://cds.cern.ch/record/2293142.

[23] D. Ceresa et al. Characterization of the MPA prototype, a 65 nm pixel readout ASIC with on-chip quick transverse momentum discrimination capabilities. Technical Report CMS-CR-2018-279, 2018. https://cds.cern.ch/record/2658203.

[24] A. Caratelli et al. Characterization of the first prototype of the Silicon-Strip readout ASIC (SSA) for the CMS Outer Tracker phase-2 upgrade. Technical Report CMS-CR-2018-286, 2018. https://cds.cern.ch/record/2650963.

[25] B. Nodari et al. A 65 nm Data Concentration ASIC for the CMS Outer Tracker Detector Upgrade at HL-LHC. Technical Report CMS-CR-2018-278, 2018. http://cds.cern.ch/record/2650712.

[26] P. Moreira. The LpGBT Project, Status and Overview. In *ACES 2016 - Fifth Common ATLAS CMS Electronics Workshop for LHC Upgrades*, 2016. https://cds.cern.ch/record/2137809.

[27] C. Soós and others. Versatile Link PLUS transceiver development. In *Proceedings, Topical Workshop on Electronics for Particle Physics (TWEPP 2016): Karlsruhe, Germany*, volume 12, page C03068, 2017. https:/inspirehep.net/record/1518507.

[28] S. Viret. Stubs. Technical Report CMS DN-2018/003, 2018. http://cms.cern.ch/iCMS/jsp/openfile.jsp?tp=draft&files=DN2018_003_v2.pdf et Annexe C.1.

[29] S. Viret. Stub windows tuning: a tutorial. https://www.dropbox.com/s/sc1ifhfbcv5pts3/Stub%20windows%20tuning-%20a%20tutorial.pdf, 2017.

[30] R. Turchetta. Spatial resolution of silicon microstrip detectors. *Nucl. Instrum. Meth.*, A335:44–58, 1993. https://doi.org/10.1016/0168-9002(93)90255-G.

[31] B.Nodari S.Scarfi S. Viret L. Caponetto, G. Galbit. CIC1 technical specification. https://espace.cern.ch/Tracker-Upgrade/Electronics/CIC/Shared%20Documents/Specifications/CIC_specs_v3.pdf et Annexe C.3, 2018.

[32] S. Viret. Data transmission efficiency of the phase II tracker front-end system for the tilted geometry. https://espace.cern.ch/Tracker-Upgrade/Electronics/CIC/Shared%20Documents/Simulation%20studies/FE_inneff_tilted.pdf et Annexe C.4, 2018.

[33] F. De Canio et al. Characterization of SLVS Driver and Receiver in a 65 nm CMOS Technology for High Energy Physics Applications. In *Proceedings, Topical Workshop on Electronics for Particle Physics (TWEPP17): Santa Cruz, CA, USA*, 2017. http://cds.cern.ch/record/2673784.

[34] K. E. Batcher. Sorting Networks and Their Applications. In *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, 1968. https://doi.org/10.1145/1468075.1468121.

[35] M. Smith. *Application-Specific Integrated Circuits*. Addison Wesley, 1997. Lien pdf.

[36] A. Baird et al. A Fast high resolution track trigger for the H1 experiment. *IEEE Trans. Nucl. Sci.*, 48:1276–1285, 2001. https://arxiv.org/abs/hep-ex/0104010.

[37] M. Dell'Orso and L. Ristori. VLSI structures for track finding. volume A278, pages 436–440, 1989. http://inspirehep.net/record/276421.

[38] J. Adelman et al. Real time secondary vertexing at CDF. *Nucl. Instrum. Meth.*, A569:111–114, 2006. https://doi.org/10.1016/j.nima.2006.09.022.

[39] Evelyn J. Thomson et al. Online Track Processor for the CDF Upgrade. *IEEE Trans. Nucl. Sci.*, 49:1063–1070, 2002. https://doi.org/10.1109/TNS.2002.1039615.

[40] M. Shochet, L. Tompkins, V. Cavaliere, P. Giannetti, A. Annovi, and G. Volpi. Fast TracKer (FTK) Technical Design Report. Technical Report CERN-LHCC-2013-007, 2013. https://cds.cern.ch/record/1552953.

[41] Fabrizio Palla and Giuliano Parrini. Tracking in the trigger: From the CDF experience to CMS upgrade. *PoS*, VERTEX2007:034, 2007. https://pos.sissa.it/057/034/pdf.

[42] Edward Bartz et al. FPGA-Based Tracklet Approach to Level-1 Track Finding at CMS for the HL-LHC. *EPJ Web Conf.*, 150:00016, 2017. https://arxiv.org/abs/1706.09225.

[43] I. Tomalin et al. An FPGA based track finder for the L1 trigger of the CMS experiment at the High Luminosity LHC. *JINST*, 12:P12019, 2017. http://cds.cern.ch/record/2287640.

[44] A. Annovi et al. A new "Variable Resolution Associative Memory" for High Energy Physics. In *Proceedings, 2nd International Conference on Advancements in Nuclear Instrumentation, Measurement Methods and their Applications (ANIMMA 2011): Ghent, Belgium*, 2011. https://cdsweb.cern.ch/record/1352152.

[45] G. Baulieu and S. Viret. Associative memory l1 track trigger camp. https://indico.cern.ch/event/325924/, 2014.

[46] G. Vidale. A study of l1 track trigger algorithms for the cms detector at hl-lhc. http://tesi.cab.unipd.it/57489/1/VidaleGiorgio_tesi.pdf, 2017.

[47] G. Baulieu & S. Viret. System performance, from default geometry to titled geometry. https://indico.cern.ch/event/591143/contributions/2385618/, 2016.

[48] G. Baulieu & S. Viret. Killing tails and combinatorics:an AM update. https://indico.cern.ch/event/354998/contributions/835298/, 2015.

[49] G. Baulieu & S. Viret. Latest Simulation results from Lyon. https://indico.cern.ch/event/521286/contributions/2134756/, 2016.

[50] G. Baulieu & S. Viret. AM Simulation Status Overview. https://indico.cern.ch/event/348920/contributions/1748967/, 2014.

[51] G. Baulieu & S. Viret. Bank optimization studies. https://indico.cern.ch/event/354336/contributions/1759770/, 2014.

[52] G. Baulieu & S. Viret. Recent ideas for AM optimization. https://indico.cern.ch/event/612363/contributions/2468787/, 2017.

[53] G. Baulieu & S. Viret. Stub bend in AM approach. https://indico.cern.ch/event/625269/contributions/2524812/, 2017.

[54] Développement d'un système électronique de reconstruction de traces pour les expériences du grand collisionneur de hadrons. https://anr.fr/Projet-ANR-13-BS05-0011, 2016.

[55] H Wind. Principal component analysis, Pattern recognition for track finding, Interpolation and functional representation. Technical Report CERN-EP-INT-81-12-REV, 1982. https://cds.cern.ch/record/139228.

[56] S. Amerio et al. The GigaFitter: Performance at CDF and Perspectives for future Applications. *J. Phys. Conf. Ser.*, 219:022001, 2010. https://iopscience.iop.org/article/10.1088/1742-6596/219/2/022001/pdf.

[57] W. Ashmanskas et al. Performance of the CDF Online Silicon Vertex Tracker. *IEEE Trans. Nucl. Sci.*, 49:1177–1184, 2002. https://inspirehep.net/record/584840.

[58] E. Clement et al. A High-performance Track Fitter for Use in Ultra-fast Electronics. *Nucl. Instrum. Meth.*, A935:95–102, 2019. https://arxiv.org/abs/1809.01467.

[59] Thomas Stephen Williams. IPbus A flexible Ethernet-based control system for xTCA hardware. Technical Report CMS-CR-2014-334, 2014. https://cds.cern.ch/record/2020872.

[60] Alberto Annovi et al. AM06: the Associative Memory chip for the Fast TracKer in the upgraded ATLAS detector. Technical Report ATL-DAQ-PROC-2016-017, 2016. https://cds.cern.ch/record/2228284.

[61] G. Galbit. Clustering et Filtrage rapide de traces pour le L1 track trigger de CMS. https://indico.in2p3.fr/event/12860/contributions/11805/, 2016.

[62] G. Fedi et al. A Real-Time Demonstrator for Track Reconstruction in the CMS L1 Track-Trigger System Based on Custom Associative Memories and High-Performance FPGAs. *PoS*, TWEPP-17, 2018. https://cds.cern.ch/record/2673659.

[63] A. Annovi et al. Characterization of an Associative Memory Chip in 28 nm CMOS Technology. In *Proceedings, 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018. https://doi.org/10.1109/ISCAS.2018.8351801.

[64] A. Stabile et al. Phase-II Associative Memory ASIC Specifications. Technical Report CERN-OPEN-2018-003, 2018. http://cds.cern.ch/record/2320701.

[65] M. Vaz et al. A Framework for Development and Test of xTCA Modules with FPGA Based Systems for Particle Detectors. In *Proceedings, 11th International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC 2016): Campinas, Brasil*, 2017. http://inspirehep.net/record/1649458/.

List of Figures

1.1 LHC . . . 7
1.2 The LHCb detector . . . 10
1.3 LHCb VELO . . . 10
1.4 VELO working principle . . . 10
1.5 Detailed view of the strips arrangement of a VELO module . . . 11
1.6 Alignment and track reconstruction . . . 12
1.7 Vertices found before alignment . . . 13
1.8 Vertex found after alignment . . . 13
1.9 The ATLAS detector . . . 14
1.10 View of fibres and photomultipliers from a tile calorimeter sector . . . 15
1.11 TileCal low level calibration scheme . . . 16
1.12 LASER box description . . . 17
1.13 LASER system light distribution scheme . . . 17
1.14 LASER event principle . . . 18

2.1 LHC and HL-LHC project phases . . . 20
2.2 Occupancy vs granularity . . . 20
2.3 Data flow of the upgraded tracker . . . 21
2.4 CMS phase II tracker . . . 22
2.5 Tracker module repartition in the R-Z plane . . . 22
2.6 2S module views . . . 23
2.7 PS module views . . . 23
2.8 Stub: basic description . . . 23

3.1 Readout: basic principle . . . 27
3.2 Detailed readout principle of a 2S module . . . 28
3.3 Detailed readout principle of a PS module . . . 29
3.4 Stub: main principle . . . 30
3.5 Stub definition . . . 31

3.6 Correction applied to the stub width in order to account for the module planarity . . . 31
3.7 Parameters defining the stub in the case of a module tilted with respect to the beam axis. . . . 32
3.8 Bend resolution for stubs induced by primary muons in the barrel . . . 33
3.9 Bend resolution for stubs induced by primary muons in the endcaps . . . 33
3.10 The different types of stubs associated with a primary particle . . . 33
3.11 Tracker regions requiring a specific stud window tuning. . . . 35
3.12 Wide stub windows for an average pileup of 140. . . . 35
3.13 Wide stub windows for an average pileup of 200. . . . 36
3.14 Resolution of poor quality stubs coming from primary muons in the barrel . 37
3.15 Resolution of good quality stubs coming from primary electrons in the barrel 37
3.16 Width of good quality stubs induced by primary muons as a function of the transverse momentum of the muon. . . . 38
3.17 Width of poor quality stubs induced by primary muons as a function of the transverse momentum of the muon. . . . 38

4.1 CIC basic concept . . . 42
4.2 CIC I/Os . . . 44
4.3 FE_ana code working principle . . . 47
4.4 Proportion of good stubs lost after the CIC chips (in %), for PU250 events sequences, using 320MHz output speed. . . . 48
4.5 Number of L1 events stored in a CIC as a function of time. . . . 49
4.6 CIC block diagram . . . 50
4.7 Word alignment in the 2S case . . . 52
4.8 L1 path data flow . . . 53
4.9 ASIC design steps [35] . . . 54
4.10 CIC2 floorplan . . . 55
4.11 CIC2 power grid . . . 55
4.12 Routed CIC2 . . . 56
4.13 CIC2 IR drop analysis. Scale of the largest IR drop (in red) is 10mV. . . . 57
4.14 Standalone CIC test bench . . . 58
4.15 Wire-bonded CIC1 on the carrier board. . . . 59
4.16 CIC1 power consumption during different operating stages. . . . 59

5.1 AM chip working principle. . . . 66
5.2 Data distribution, space multiplexing . . . 68

5.3 Data distribution, time multiplexing 68

6.1 AM-based track reconstruction. 71
6.2 Pattern bank optimisation 72
6.3 A pattern in a three-layers detector. Green tracks will match the pattern, red tracks won't. 73
6.4 Geometric definition of a superstrip. 74
6.5 Address definition of a superstrip and representation of a 6 layers pattern. . 74
6.6 Merging high-resolution patterns belonging to the same low-resolution pattern. 75
6.7 Associating the merging results to ternary bits values (1-bit case). 75
6.8 The variable resolution pattern compared to the low-res one. 75
6.9 Coding 15 positions with 2 ternary bits (classic encoding). 76
6.10 Coding 15 positions with 2 ternary bits (Gray encoding). 76
6.11 Turn-on curves using (a) a flat pT spectrum (b) a flat 1/pT spectrum (c) 3 ranges with a 1/pT spectrum 78
6.12 Each superstrip in the pattern bank points to a superstrip in the virtual detector. 78
6.13 Fixed (left) and projective (right) patterns 80
6.14 Barrel bank properties. Tracker parts involved in the bank are in green. Corresponding superstrips size (in strip units), and DC bit numbers are provided. 80
6.15 Hybrid bank properties. Tracker parts involved in the bank are in green. Corresponding superstrips size (in strip units), and DC bit numbers are provided. 81
6.16 Endcap bank properties. Tracker parts involved in the bank are in green. Corresponding superstrips size (in strip units), and DC bit numbers are provided. 81
6.17 Efficiency for primary muons without PU 83
6.18 Efficiency for primary electrons without PU 83

7.1 Number of stub combinations per event to feed to the PCA processor, for $PU140 + t\bar{t}$ events (black line) and $PU200 + t\bar{t}$ events (red line) 88
7.2 TC builder principle 89
7.3 Stubs selection at projection stage in the TC module 91
7.4 Δ_ϕ values for particle gun muons (black) and electrons (red). Seeding layers are the two barrel innermost rings, projection layer is barrel fourth ring. 91
7.5 Proportion of stubs filtered according to the total number of stubs in the event after associative memories (black circles) and TC builder (white circles) 94

7.6 Number of stub combinations over the number of TC per event for $PU140+t\bar{t}$ (black) and $PU200+t\bar{t}$ (red) 94
7.7 Number of matched roads over the number of TC per event for $PU140+t\bar{t}$ (black) and $PU200+t\bar{t}$ (red) 94
7.8 Final number of L1 tracks per events, for PU140+ttbar events (black line) and PU200+4ttbar events (red line) 96
7.9 AM filtering and TC builder in action on a complex PU200+4tops event. . . 96
7.10 The FastTrack demonstrator principle 96
7.11 FastTrack test-bench 97
7.12 Test-bench working principle 97
7.13 Demonstrator firmware block diagram 98
7.14 Average clock cycles spent in the various steps of the firmware for PU140 $t\bar{t}$ events 99
7.15 Distribution of clock cycles spent to treat PU140 $t\bar{t}$ events, 95% limit in red. 100

List of Tables

3.1 Frontend extraction limits, in average number of stubs per module and per collision. . . . 34

3.2 Proportions of reconstructible tracks measured with different types of wide stub windows sets. . . . 36

3.3 Proportions of reconstructible electrons measured with wide and tight stub windows tunings define at PU140. . . . 36

3.4 Proportion of reconstructible tracks with or without bend-based selection. . . . 38

4.1 CIC technical specifications. . . . 50

4.2 Comparing power consumption estimated for CIC1 and CIC2. . . . 61

6.1 Efficiencies, up to the pattern recognition stage, for particles with $p_T \geq 3$GeV/c 83

6.2 Matched road rates and fake road proportion after AM for $t\bar{t}$ events. 95% rate limits are given in brackets 84

6.3 Average stub rates per trigger tower after the AM stage. A good stub is a stub induced by a particle with $|d_0| < 1cm$ and $p_T > 3GeV/c$ 85

7.1 Efficiencies up to the final L1 track, for particles with $p_T \geq 3$GeV/c 93

7.2 TC rates and fake TC proportion. 95% limits are provided into brackets. . . . 94

7.3 Comparison of the 95% limits for track candidates and raw number of combinations without TC builder 95

7.4 L1 track rates and fake proportion. 95% limits are provided into brackets. . . . 95

ACKNOWLEDGMENTS

I realize as I write those lines that it is much more complicated to write HDR acknowledgments than thesis ones. This is probably why we rarely find this section in HDRs (we rarely find HDRs in general anyway...).

Of course, it's more complicated to forget nobody after 20 years than after 3. There were a lot of great encounters during this period, that's what I remember.

Many of these good times have an important correlation with the members of my examination panel, thus a big thank you to them for agreeing to be part of it. You have all, at one time or another in my career, exerted a positive influence through your human and scientific qualities. I am happy to have been able to meet researchers like you, and I hope that one day the same can be said about me.

There are many other people that I should thank, starting with all the colleagues from the different laboratories that I have worked with in 20 years. The list would be too long and I would risk forgetting people, so I am not going to embark on this perilous exercise.

However, much of the work described in this document would not have seen the light of day without the fundamental contributions of some IP2I engineers. A big thank you to Guillaume, Geoffrey, Luigi, and Benedetta, because without you I would have had a lot less chapters to write.

Last but not least, thank you to my family who support me on a daily basis. Living with a researcher is not easy, and the main suspect does not always realize it... With age, it starts to come, but I understand that it can seem long to you sometimes.

VI

APPENDIX

A

LHCb NOTES

A.1/ ALIGNMENT PROCEDURE

University of Glasgow | Experimental Particle Physics

GLAS-PPE/2008-06
10th July 2008

Department of Physics and Astronomy
Experimental Particle Physics Group
Kelvin Building, University of Glasgow,
Glasgow, G12 8QQ, Scotland
Telephone: +44 (0)141 330 2000 Fax: +44 (0)141 330 5881

Alignment procedure of the LHCb Vertex Detector

S. Viret, C. Parkes, M. Gersabeck

Department of Physics and Astronomy, University of Glasgow
University Avenue, Glasgow, G12 8QQ, United Kingdom

Abstract

LHCb is one of the four main experiments of the Large Hadron Collider (LHC) project, which will start at CERN in 2008. The experiment is primarily dedicated to B-Physics and hence requires precise vertex reconstruction. The silicon vertex locator (VELO) has a single hit precision of better than 10 μm and is used both off-line and in the trigger. These requirements place strict constraints on its alignment. Additional challenges for the alignment arise from the detector being retracted between each fill of the LHC and from its unique circular disc r/ϕ strip geometry. This paper describes the track based software alignment procedure developed for the VELO. The procedure is primarily based on a non-iterative method using a matrix inversion technique. The procedure is demonstrated with simulated events to be fast, robust and to achieve a suitable alignment precision.

A.2/ ALIGNMENT TESTS

University of Glasgow | Experimental Particle Physics

GLAS-PPE/2008-07
10th July 2008

Department of Physics and Astronomy
Experimental Particle Physics Group
Kelvin Building, University of Glasgow,
Glasgow, G12 8QQ, Scotland
Telephone: +44 (0)141 330 2000 Fax: +44 (0)141 330 5881

Performance of the LHCb Vertex Detector Alignment Algorithm determined with Beam Test Data

M. Gersabeck*[a],S. Viret*[a],C. Parkes*[a] *et al.*
(the LHCb VELO group)
[a] University of Glasgow, Glasgow, United Kingdom
* *corresponding authors*

Abstract

LHCb is the dedicated heavy flavour experiment at the Large Hadron Collider at CERN. The partially assembled silicon vertex locator (VELO) of the LHCb experiment has been tested in a beam test. The data from this beam test have been used to determine the performance of the VELO alignment algorithm. The relative alignment of the two silicon sensors in a module and the relative alignment of the modules has been extracted. This alignment is shown to be accurate at a level of approximately 2 μm and 0.1 mrad for translations and rotations, respectively in the plane of the sensors. A single hit precision at normal track incidence of about 10 μm is obtained for the sensors. The alignment of the system is shown to be stable at better than the 10 μm level under air to vacuum pressure changes and mechanical movements of the assembled system.

B

ATLAS NOTES

B.1/ LASER DESCRIPTION

ATLAS NOTE

ATL-TILECAL-INT-2010-001

January 25, 2010

ATLAS tile calorimeter LASER calibration system

R. Alves[b], D. Calvet[a], J. Carvalho[b], R. Chadelas[a], M. Crouau[a], C. Crozatier[a], R. Febbraro[a], D. Lambert[a], R. Marchionini[a], J. Oliveira[b], A. Pereira[b],C. Santoni[a], L.P. Says[a], F. Vazeille[a], S. Viret[a]

[a] *Laboratoire de Physique Corpusculaire, Clermont-Ferrand, France*
[b] *Dept of Physics, Coimbra, Portugal*

Abstract

This note provides a detailed description of an important part of the ATLAS tile calorimeter (TILECAL) calibration framework: the LASER monitoring. The main role of this system is to monitor the stability and the linearity of the 9852 photomultiplier tubes (PMTs) and associated electronics used in the detector. This will be of particular importance for physics analysis, as (TILECAL) performance will be closely related to our understanding of the PMT response.

B.2/ LASER COMMISSIONING

ATLAS NOTE

ATL-COM-TILECAL-2010-014

June 7, 2011

Commissioning of the ATLAS tile calorimeter LASER calibration system

D. Calvet[a], J. Carvalho[b], R. Febbraro[a],
B. Galhardo[b], C. Santoni[a], F. Vazeille[a],
S. Viret[a]

[a] *Laboratoire de Physique Corpusculaire, Clermont-Ferrand, France*
[b] *Dept of Physics, Coimbra, Portugal*

Abstract

This note provides details on the performance of the ATLAS tile hadronic calorimeter (TILECAL) LASER monitoring system. The main role of this system is to monitor the stability and the linearity of the 9852 photomultiplier tubes (PMTs) and associated electronics used in the detector.

C

CMS NOTES

C.1/ STUBS STUDY

Available on CMS information server **CMS Internal Note DN-2018/003**

CMS Internal Note

January 17, 2020

Stub definition and properties

S. Viret [a)]

Abstract

This document describes in details an important component of the future CMS Phase-2 tracker: the stub. The stub is the main building block of the future level-1 track trigger, and understanding its properties is therefore of particular importance.

[a)] Institut de Physique des 2 Infinis, UCBL, CNRS-IN2P3, Lyon, France

C.2/ FRONT END DYNAMIC INEFFICIENCIES

Available on CMS information server **CMS Internal Note IN-2017/XXX**

CMS Internal Note

January 4, 2017

Data transmission efficiency of the phase II tracker front-end system for the tilted geometry.

S. Viret [a)]

Abstract

Phase II outer tracker of the CMS experiment, due to the inclusion of the tracking at the first level of the trigger, will be a key component of CMS future triggering strategy. L1 tracking will be a very challenging task requiring the fast treatment of a huge data troughput. In order to meet this challenge, data rate reduction will be performed directly at the frontend level. The trigger signal will be compressed in two stages before being extracted from the detection module. The second compression step, performed by a concentrator chip (CIC), will heavily rely on average occupancy in the tracker. If this occupancy becomes to large, one has to evaluate how large the risk of data losses becomes. This document is an update of the previous losses studies, based on the future baseline tilted geometry.

[a)] Institut de Physique Nucléaire, UCBL, CNRS-IN2P3, Lyon, France

C.3/ CIC1 SPECIFICATIONS



CIC1 technical specification

L. Caponetto *(1)*, ***G. Galbit*** *(1)*, ***B.Nodari*** *(1)*, ***S.Scarfi*** *(2)*, ***S. Viret*** *(1)*

(1)-Institut de Physique Nucléaire de Lyon (FR)

(2)-CERN

Version 3.1

10/12/2018



C.4/ TRACK TRIGGER DEVELOPMENT

Available on CMS information server

CMS Internal Note DN-2015/025

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

May 18, 2020

Emulation of a track reconstruction system based on associative memories

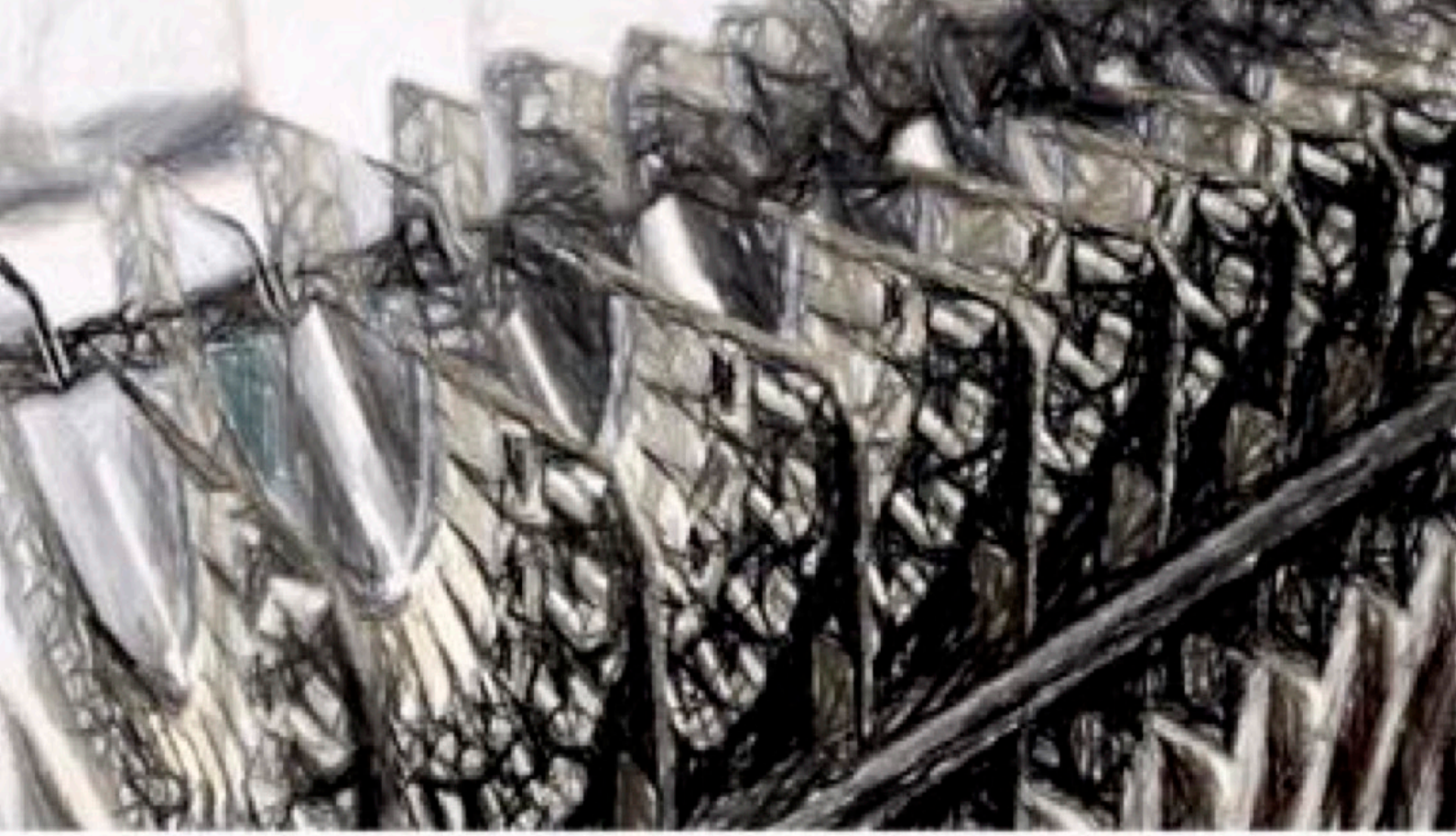
G. Baulieu [a)], G. Galbit [a)], W. Tromeur [a)], S. Viret [a)]

Abstract

The role of this note is to study the feasibility of an L1 tracking system based on associative memories (AM) for the CMS experiment Phase-2 upgrade. This work is based on simulations performed using a complete software emulation of AM-based tracking. A detailed description of this emulation, along with an estimation of the system parameters, is provided. A small hardware demonstrator is also presented.

[a)] Univ Lyon, Univ Claude Bernard Lyon 1, CNRS/IN2P3, IP2I Lyon, UMR 5822, F-69622, Villeurbanne, France

Document generated with LaTeX and:
the LaTeX style for HDR created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/



So long, and thanks for all the fish