UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di
Scienze Matematiche
Fisiche e Naturali
Corso di Laurea magistrale in
Fisica Nucleare e Subnucleare

Development of calibration techniques and performance analysis of the CMS Inner Tracker for the High Luminosity phase of LHC

Sviluppo di tecniche di calibrazione e analisi delle prestazioni del Tracciatore Interno dell'esperimento CMS per la fase ad Alta Luminosità di LHC

Candidate:
Lorenzo Damenti

Advisor:
Dr. Giacomo Sguazzoni

Co-Advisor:
Prof. Piergiulio Lenzi

Academic year: 2021/2022

CONTENTS

Introduction **2**

Acronyms **5**

Glossary **7**

1 LHC and CMS experiment **10**
1.1 The Large Hadron Collider . . . 10
1.1.1 Particle Energy . . . 11
1.1.2 Luminosity . . . 12
1.2 Compact Muon Solenoid . . . 14
1.3 High Luminosity LHC . . . 18
1.3.1 CMS Upgrade . . . 19
1.3.2 Motivation for HL-LHC . . . 23

2 The CMS Inner Tracker for HL-LHC **25**
2.1 CMS IT Read Out Chip . . . 26
2.1.1 Front-End Registers . . . 30
2.2 Different types of Pixel Sensors . . . 31
2.3 Serial Powering . . . 32
2.4 Channel Threshold . . . 34
2.4.1 Calibration injection circuit . . . 35
2.4.2 Threshold adjust procedure . . . 36

3 Setup for Bench testing **37**
3.1 CROC Module and DAQ System . . . 37
3.1.1 The module test board . . . 37
3.1.2 DAQ System . . . 38
3.2 Chip properties and conversion factors . . . 40
3.2.1 Threshold stability as a function of the number of injected pixels 40
3.2.2 V_{cal} to electrons conversion . . . 41

3.2.3 GDAC to V_{cal} conversion . . . 43
3.2.4 TDAC · V_{LDAC} to V_{cal} conversion . . . 44
3.2.5 *KRUM_CURR_LIN* to I_{Krum} conversion . . . 46

4 Threshold Oscillations **48**
4.1 Synchronous and Asynchronous modes . . . 48
4.2 Tornado Plots . . . 50
4.2.1 Tornado Plots and Krummenacher current . . . 51
4.3 Fine Delay and Threshold Tuning . . . 55
4.3.1 Optimal Fine Delay and Noise performance . . . 57
4.3.2 How to find the Optimal Fine Delay . . . 58
4.4 Threshold Oscillation in Synchronous and Asynchronous mode . . . 60
4.5 Noise Performance and Threshold Oscillation . . . 60

5 Oscillation Analysis **65**
5.1 Oscillation changes within the pixel matrix . . . 65
5.2 Comparison with different sensors . . . 69
5.2.1 Preliminary Checks . . . 69
5.2.2 Oscillation Amplitude and Tuned Threshold . . . 70
5.2.3 Final Comparison . . . 71

6 Detailed description of oscillations **73**
6.1 Production of Synchronous mode . . . 74
6.1.1 Actual Threshold Oscillation and Krummenacher current . . . 75
6.2 Production of Asynchronous mode . . . 76
6.2.1 Comparator Threshold Oscillation . . . 77
6.3 Explanation of the differences between sensors . . . 79
6.4 Origin of Threshold oscillation . . . 82
6.4.1 Clock distribution within pixels matrix . . . 82
6.4.2 Clock Distribution and Synchronous mode . . . 83

Conclusions **87**

Bibliography **89**

Acknowledgements **92**

INTRODUCTION

The *Large Hadron Collider* (LHC) is the world's largest and highest-energy particle accelerator. It is able to reach a center of mass energy of $\sim$14 TeV and it has been designed to deliver a peak instantaneous luminosity of $1 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$ in proton-proton collisions (even if in 2018 it reached the value of $\sim 2 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$). The current LHC physics program, that has allowed for the discovery of the Higgs boson in 2012, will end in 2025. In 2029, LHC will enter a new era, named *High Luminosity LHC*, or HL-LHC. During this new phase, the machine will reach a peak instantaneous luminosity of $\sim 7.5 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$ which will result in an integrated luminosity of 3000-4000 $\mathrm{fb^{-1}}$ in about ten years of operation.

This enormous dataset will give access to many physics opportunities, including precise measurements of the Higgs boson couplings, and the search for rare Standard Model and Beyond Standard Model processes. However, the increased luminosity will result into extraordinary challenges for the experiments: an increased number of simultaneous proton-proton interactions per bunch crossing and unprecedented levels of the radiation that the detector parts will be exposed to. In order to fully exploit the physics potential during the HL-LHC era, the *Compact Muon Solenoid* (CMS) will undergo major upgrades, known as CMS Phase–II Upgrades.

In particular, a complete new tracking system will be built. As its predecessor, it will be divided in two main subsystems: the Outer Tracker (OT) and the Inner Tracker (IT). The latter will be based on silicon pixel sensors. In order to operate in the conditions of HL-LHC, both Planar and 3D sensors will be used.

The main purpose of this thesis is to develop calibration techniques and to analyze the performance of silicon pixel sensor modules for the CMS Phase–II Inner Tracker. A sensor module is the assembly of the sensor connected to one or more read-out chips and the appropriate high-density interconnect for powering and readout. Within the present study, test modules with 145 152 pixels are used, bonded to a Read-Out Chip called CROC-v1 (CMS Read-Out Chip version 1). The DAQ system used to read out the modules is called *Phase–II Acquisition and Control Framework* (Ph2ACF). The software, written in C++, features dedicated procedures

to configure and calibrate the CROC and runs the hardware, which consists of an FPGA board.

An important parameter which can be tuned using Ph2ACF is the threshold of each pixel cell. A part of this work is dedicated to the development of calibration techniques that allows to tune the threshold as accurately as possible. Within the scope of this work, periodic oscillations of the threshold have been observed. Since the threshold is a crucial parameter for the tracking system operation, a detailed investigation of this effect is extremely important. Threshold oscillations have been thoroughly studied and described to understand their origin.

In this work I describe how a deep understanding of a crucial component of the detector device, i.e. the CMS read-out chip, allows to optimize and improve the performance of the CMS Inner Tracker for the High Luminosity LHC.

ACRONYMS

- **AFE**: Analog Front End
- **ALICE**: A Large Ion Collider Experiment
- **ATLAS**: A Toroidal LHC ApparatuS
- **BX**: Bunch Crossing
- **CMS**: Compact Muon Solenoid
- **CROC**: CMS Read–Out Chip
- **CSA**: Charge Sensitive Amplifier
- **DAC**: Digital to Analog Converter
- **DAQ**: Data AcQuisition
- **ENC**: Equivalent Noise Charge
- **FBK**: Fondazione Bruno Kessler
- **FPGA**: Field Programmable Gate Array
- **HL-LHC**: High Luminosity LHC
- **HPK**: Hamamatsu Photonics K.K.
- **IP**: Interaction Point
- **IT**: Inner Tracker
- **ITμTDC**: Inner Tracker micro Data Trigger and Control
- **LHC**: Large Hadron Collider
- **LHCb:** LHC beauty
- **Ph2ACF**: Phase–II Acquisition and Control Framework
- **PU**: Pile–Up

- **QCD**: Quantum CromoDynamics
- **ROC**: Read–Out Chip
- **SdS**: Smallest detectable Signal
- **SLDO**: Shunt Linear Drop-Out voltage regulator
- **SNR**: Signal to Noise Ratio
- **ToT**: Time over Threshold
- **TW**: Time Walk

GLOSSARY

- **V_{cal} Unit**: this a CROC internal voltage unit. With a good approximation, one V_{cal} is the voltage seen between feedback capacitance's armors if they are charged with a charge Q equal to $5\,e^-$. Several CROC internal voltages are generated as $N \cdot V_{cal}$ where N is the integer number set in the corresponding configuration register.
- **Synchronous mode**: in this sampling mode, a hit is recorded if the comparator output is up at the leading 40 MHz clock edge. In Synchronous mode, the comparator output is up as long as the signal exceeds the threshold.
- **Asynchronous mode**: as for Synchronous mode, also in Asynchronous mode a hit is recorded if the comparator output is up at the leading 40 MHz clock edge. However, in this case, the comparator output goes down 25 ns after the signal exceeded the threshold.
- **Coarse Delay**: the value of the Coarse Delay is used to select the clock cycle when the injection must be performed. It is indicated with BX_{ID} because, in the standard detector operation on beam, the clock cycle eventually identifies a bunch crossing (BX). Its value is set in the *CalCommand* register.
- **Fine Delay**: the *Fine Delay* is the delay between the leading 40 MHz clock edge and the actual charge calibration injection. The value of the Fine Delay is set in units of $\frac{25\,ns}{32}$ in the register called *CalibrationConfig*.
- **S-Curve**: the *S-Curve* is a plot which shows the channel efficiency or occupancy as a function of the magnitude of the input signal (that can be a calibration signal injected on purpose or a particle signal collected by the pixel cell attached to the channel). The S-Curve 50% value corresponds to the channel threshold; the S-Curve slope at the threshold does depend on the channel noise.
- **Comparator Threshold**: this is the physical threshold of the comparator stage in the Analog Front End. The Comparator Threshold is referred to as V_{Thr} and can be configured by using the readout chip registers.

- **Actual Threshold**: this is the Smallest Detectable Signal for a given delay and Comparator Threshold. The value of the Actual Threshold can be obtained running a dedicated procedure called *ThresholdScan*. The Actual Threshold corresponds to the Comparator Thresholds for a specific delay value given by the correct clock cycle and the so called Optimal Fine Delay.
- **Target Threshold**: the Target Threshold is the value of the threshold to which the procedure *GlobalThresholdTuning* sets the chip working point by appropriately configuring the relevant registers. If the Fine Delay chosen for the procedure is the Optimal Fine Delay the Target Threshold corresponds to the Comparator Threshold, otherwise the Target Threshold is the Actual Threshold for the chosen generic Fine Delay.
- **Tornado Plot**: the Tornado Plot is a 2D scan on Q and Δt, where Q is the particle or injection signal induced charge and Δt is the time between the arrival of the signal and the sampling edge. In particular, Δt can be defined as:

$$\Delta t \;=\; (32 \times BX_{ID} \;-\; \text{Fine Delay}) \;\times \left[\frac{25\,\text{ns}}{32}\right] \tag{1}$$

  In general, the Tornado Plot does depend on the Comparator Threshold.

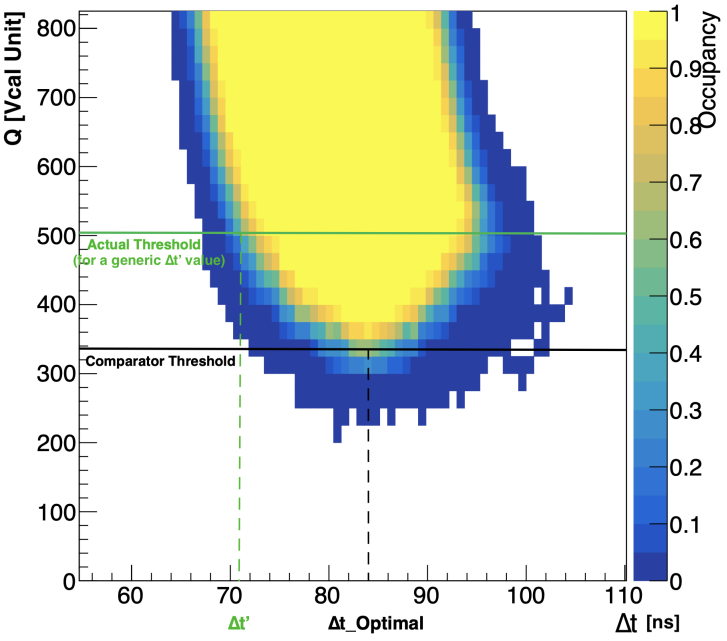


Figure 1: Example of Tornado Plot. The boundary between yellow and blue regions gives the threshold as a function of the delay (please note that the delay range spans across several 25 ns clock cycles). The Comparator Threshold corresponds to the horizontal black line and the dashed vertical black line corresponds to the Optimal Fine Delay within the correct clock cycle. If the *ThresholdScan* procedure is run with a different delay (for example the one corresponding to the dashed vertical green line) the resulting measured Actual Threshold, corresponds to the horizontal green line.

CHAPTER 1

LHC AND CMS EXPERIMENT

1.1 The Large Hadron Collider

The *Large Hadron Collider* (LHC) is the world's largest and highest-energy particle accelerator [1]. It is installed in an underground tunnel of 27.6 km circumference which was originally built for the *Large Electron and Positron* Collider (LEP). The LHC tunnel is located across the Franco-Swiss border at the *Conseil Européen pour la Recherche Nucléaire* (CERN), near Geneva.

The prime motivations of the LHC are the exploration of the Standard Model in the TeV energy range, the search for the Higgs boson (that was eventually discovered in 2012) and the search for potential new physics signatures. Furthermore, at this energy scale, there are high hopes for discoveries that could pave the way toward a unified theory.

The LHC is also able to provide heavy-ion beams at energies 30 times higher and more than previous accelerators. This allows to extend the study of the Quark Gluon Plasma (QGP) predicted by the quantum chromodynamics (QCD).

The Large Hadron Collider consists of two parallel pipes in which the accelerated particles flow in opposite directions. A magnetic field, orthogonal to the ring, is used to steer the charged particles into a quasi circular trajectory. Superconducting dipole magnets are used to generate the very strong magnetic field (up to $\sim$8 T) that is needed. As a consequence, most of the LHC ring is kept at cryogenic temperatures, which represented a true challenge during the LHC development. Moreover, the pressure inside the pipes is kept close to 10^{-13} atm, in order to minimise the spurious interactions between the accelerated particles and the residual gas molecules. The LHC pipes cross in four dedicated interaction points (IP), where as many experiments are placed to recollect the interaction products. These are CMS, ATLAS, ALICE and LHCb:

- *Compact Muon Solenoid* (CMS) and *A Toroidal LHC ApparatuS* (ATLAS):

these are multipurpose experiments designed to elucidate the nature of electroweak symmetry breaking and to study the Higgs mechanism. ATLAS [2] and CMS [3] are steered by independent collaborations in order to cross-check the scientific results. The Higgs Boson was discovered in 2012 both by CMS and ATLAS [4].

- *LHC beauty* (LHCb): LHCb experiment is mainly dedicated to flavor physics, i.e. to precision measurements of the CP violation and B–hadron decays [5].
- *A Large Ion Collider Experiment* (ALICE): the ALICE experiment is a detector optimized for heavy-ion physics, which mainly focus on QCD. In particular, ALICE is designed to study the properties of quark-gluon plasma [6].

The LHC tunnel is divided in octants: ATLAS is placed in Point 1 (first octant), ALICE in Point 2 (second octant), while CMS and LHCb are placed respectively in Point 5 (fifth octant) and Point 8 (eight octant). The schematic layout of LHC is shown in Fig. 1.1.

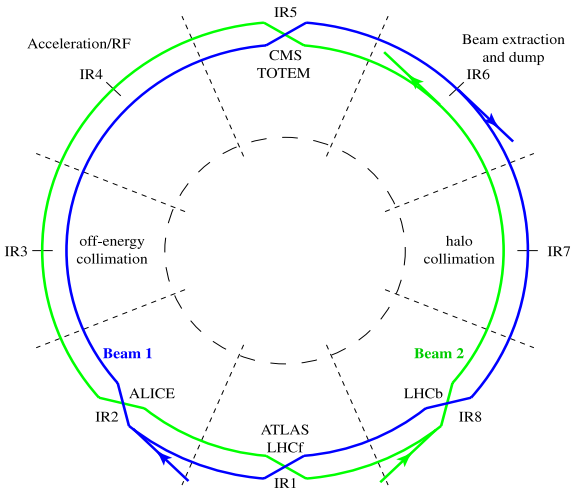


Figure 1.1: Schematic layout of the LHC pipes. The four main LHC experiments are placed at the IPs, where the pipes cross.

1.1.1 Particle Energy

One of the most important design parameter for an accelerator is the maximum energy that the particles can reach. Indeed, the higher is the center–of–mass energy of the collisions, the bigger is the mass of the particles that could be produced. Protons (or ions) reach their maximum energy in LHC where they are injected after a series of the pre-acceleration stages shown in Fig. 1.2.

Since 2020, the Linear accelerator 4 (Linac4) is the source of proton beams. It accelerates negative hydrogen ions to 160 MeV to prepare them to enter the Proton Synchrotron Booster (PSB). The ions are then stripped of their two electrons during injection from Linac4 into the PSB, leaving only protons. The PSB accelerates the protons to an energy of 2 GeV for the injection into the Proton Synchrotron (PS) which brings the beam to an energy of 25 GeV. The PS also groups the particles

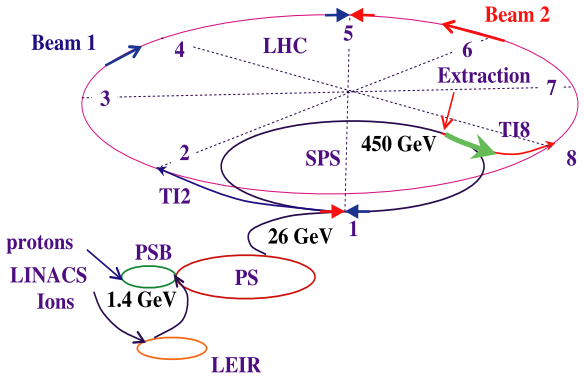


Figure 1.2: Schematic representation of the injection stages in LHC.

into bunches with a frequency of 40 MHz. Finally, protons get accelerated to an energy of 450 GeV by the SPS (Super Proton Synchrotron). After these stages, the particles are injected into the two pipes of LHC and then accelerated to an energy up to ∼7 TeV.

The pre-acceleration is similar in case of ions. For example, lead ions are obtained from a source of vaporised lead and enter Linear accelerator 4 (Linac4) before being collected and accelerated in the Low Energy Ion Ring (LEIR). They then follow the same route to maximum energy as the protons.

The magnetic field in which the pipes are inserted plays a key role during this final acceleration stage. Indeed, the momentum of the particle can be obtained as follows:

$$P[\mathrm{GeV/c}] \sim 0.3 \cdot B[\mathrm{T}] \cdot r[\mathrm{m}] \tag{1.1}$$

In Eq. 1.1, $r[m]$ is the curvature radius ad it's close to 4×10^3 m. In order to keep ∼7 TeV beams within the LHC circumference, a magnetic field of about 8 T is needed. This is achieved by using superconductive magnets, which need to be cooled to about 2 °K.

1.1.2 Luminosity

Beside the particle's energy, another important parameter for a particle accelerator is the luminosity. In general, luminosity is defined as the rate of a given process ($\dot{n} = \frac{dn}{dt}$) per unit of cross-section of that process (σ):

$$L = \frac{\dot{n}}{\sigma} \tag{1.2}$$

According to Eq. 1.2, the higher is the luminosity, the higher is the event rate for a given process. The instantaneous luminosity is measured in units of $\mathrm{cm}^{-2}\mathrm{s}^{-1}$. However, since the cross-section is normally given in units of barn[1] (b), luminosity can also be given in units of $\mathrm{b}^{-1}\mathrm{s}^{-1}$. If the collision between two particle bunches is central, the instantaneous luminosity can be obtained using the following formula:

$$L = \frac{N_1 N_2 k f_{rev}}{4\pi\sigma_x\sigma_y}. \tag{1.3}$$

[1] $1\,\mathrm{b} = 10^{-24}\,\mathrm{cm}^2$

In case of LHC: N_1 and N_2, i.e. the proton number in each bunch, are typically $1 - 2 \times 10^{11}$; k, the number of bunches, is between 2000 and 2800; f_{rev}, the revolution frequency, is about $c/27\,\text{km}$; σ_x (σ_y) is the standard deviation of the Gaussian distribution of the particle density in the bunches along x (y) (in LHC $\sigma_x \sim \sigma_y \sim$ µm).

The design value of the instantaneous luminosity is around $1.0 \times 10^{34}\,\text{cm}^{-2}\text{s}^{-1}$. However, as shown in Fig. 1.3 [7], the value of peak luminosity has grown constantly since 2010, and, in 2018, it reached the value of $\sim 2 \times 10^{34}\,\text{cm}^{-2}\text{s}^{-1}$.

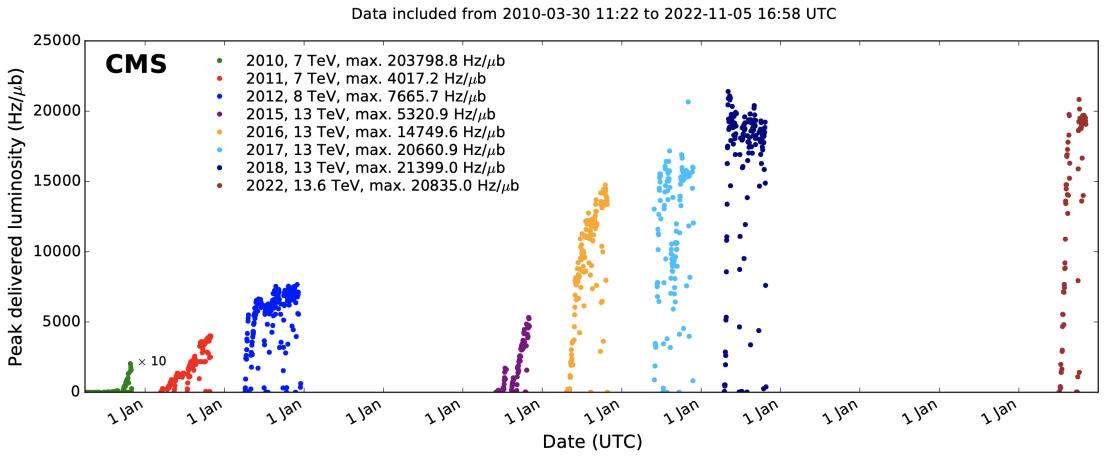


Figure 1.3: Instantaneous luminosity versus time delivered to CMS during stable beams and for pp collisions.

The one described in Eq. 1.2 is the so-called *Instantaneous Luminosity*. However, also the *Integrated Luminosity* ($\mathscr{L}$) can be defined. As the name suggests, the latter is the instantaneous luminosity integrated on a period of time (Eq. 1.4), and it is measured in units of b^{-1}.

$$\mathscr{L} = \int L dt. \tag{1.4}$$

The integrated luminosity as a function of time collected by CMS is shown in Fig. 1.4.

The *pile-up* is the number of simultaneous pp collisions within a single bunch crossing[2] (BX) and it is an important parameter from the experimental point of view (see paragraph 1.3.1 for more details). The pile-up depends on the instantaneous luminosity: the higher is the latter, the higher the former. Currently, the average pile-up figure for both ATLAS and CMS is close to 40/BX, but under certain condition it can also reach the peak value of 70/BX.

[2]The term *bunch crossing* is used to identify the moment when two bunches reach the interaction point of LHC and overlap each other giving place to several pp collisions.

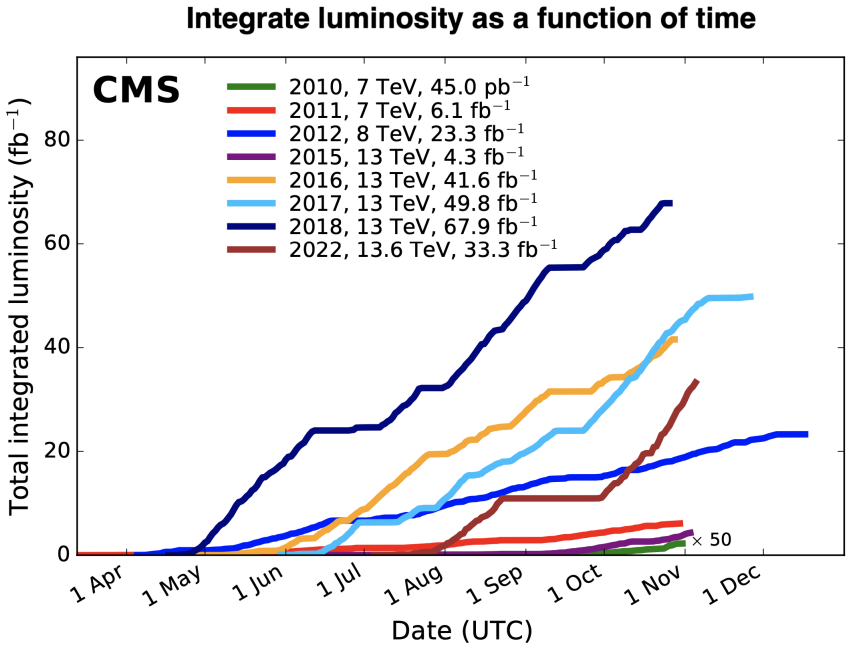


Figure 1.4: Integrated luminosity versus time delivered to CMS during stable beams for pp collisions at nominal center-of-mass energy.

1.2 Compact Muon Solenoid

The *Compact Muon Solenoid* (CMS) is one of the four LHC main experiments [3]. It is formed by a central cylindrical structure, called *barrel*, sealed on both side by structures called *endcaps*. The apparatus has a total length of 28.7 m, a diameter of 15.0 m, a total weight of 14 000 t and it is placed in a cavern $\sim$ 90 m underground. In order to identify all the different particles that are produced in a proton - proton collision, CMS is built up by different kinds of sub-detectors, which are stratified around the interaction point. The overall layout of CMS and of all its sub-detectors is shown in Fig. 1.5.

The main CMS sub-detectors are described below, with a particular attention to the tracking system.

Tracking System. At the heart of CMS sits the tracking volume, which is a cylinder of 5.8 m length and 2.6 m diameter. The structure is formed by a central barrel and two endcaps to allow for a tracking coverage up to $|\eta| < 2.5$, where η is the pseudorapidity[3]. The overall working temperature of the CMS tracker is between

[3]To study all the processes that can take place during a pp collision, it is useful to define the *rapidity* (y) and the *pseudorapidity* (η) as follows:

$$y = \frac{1}{2}\ln\left(\frac{E + p_z}{E - p_z}\right), \quad \eta = -\ln\left(tan\frac{\theta}{2}\right) \tag{1.5}$$

Usually, the interaction point corresponds to the origin of the right-handed xyz reference frame. The y-axis is orthogonal to the acceleration ring and points up, while the x-axis points to the center of LHC. In Eq. 1.5 E and p_z are the energy and the momentum along z of the produced particle, while θ is the polar angle of the trajectory measured from the z-axis. It is important to point out

$-10\,°C$ and $-20\,°C$. The tracking volume is formed by two main sub-detectors: the *Pixel Detector* and the *Strip Detector*.

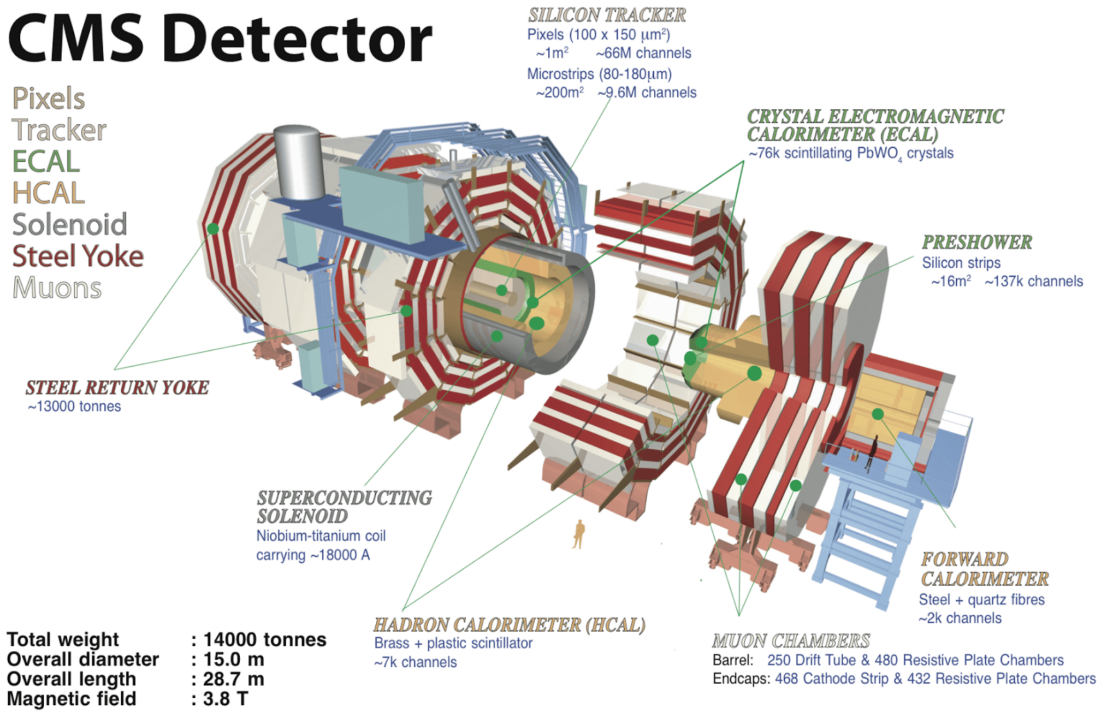


Figure 1.5: A perspective view of the CMS detector and of its sub-detectors.

A sketch of the actual tracking system is shown in Fig. 1.6

Pixel Detector. The Pixel Detector is the detector closest to the interaction point, and it is made by silicon pixel modules. It is crucial to improve the measurement of the impact parameter[4] of charged-particle tracks, as well as the position of secondary vertices. Until 2016, the *Phase–0* Pixel Detector was used in CMS. It was formed by three layers of pixel modules in the barrel region and by two disks in each endcap. In 2017 the *Phase–I* Pixel Detector was installed. It is formed by four layers in the barrel region and by three disks in each endcap.

Strip Detector: The CMS Strip Detector is made of silicon microstrips detectors. As shown in Fig. 1.6, the barrel region of the strip detector can be divided in an inner part, formed by four layers, called *Tracker Inner Barrel* (TIB), and in an outer part, which is formed by six microstrip layers, known as *Tracker Outer Barrel* (TOB). The TIB is sealed by two endcaps formed by three disks each, called *Tracker Inner Disk* (TID), while the TOB is sealed by two *Tracker End-Cap* (TEC), formed by nine disks each. The present tracker features both single-sided strip modules and double-

that Δy and $\Delta\eta$ are Lorentz's invariants.

[4]The *Impact Parameter* is defined as the perpendicular distance between the trajectory of a secondary particle and the interaction point. If the same distance is measured on the x-y plane, it is called *transverse impact parameter*, while the *longitudinal impact parameter* is the impact parameter measured along the z-axis.

sided modules. The latter are composed of two back-to-back silicon strip detectors with a stereo angle of 100 mrad. Double-sided modules are used to provide coarse measurements of the z and r coordinates in the barrel and endcaps, respectively.

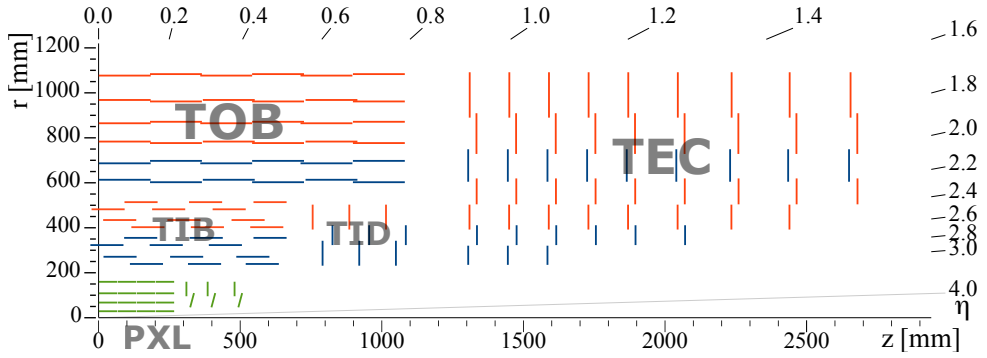


Figure 1.6: Sketch of one quarter of the Phase-I CMS tracking system in r-z view. The pixel detector is shown in green (PXL), while single-sided and double-sided strip modules are depicted as red and blue segments, respectively.

The resolution of position measurement in tracking system depends on physics processes and on external parameters [8]. In particular, the former are the diffusion of the charge carriers and the statistical fluctuation of the energy loss, while the latter are the strip pitch or pixel all size, the signal-to-noise ratio[5] (SNR) and the type of readout. In order to describe the latter parameter, suppose to focus on a single tracking layer. If a particle's track is so that most of the carriers are collected by a single pixel or strip, the position coordinates on the fixed layer comes from the measurements of the position of that strip or pixel. So, if the ionizing particles between two readout strips or on the pixel surface are *uniformly distributed*, the resolution of the measurement is:

$$\sigma \simeq \frac{d}{\sqrt{12}} \tag{1.6}$$

where d is the distance between two strips or the pixel size. Because d $\simeq 50\,\mu$m, the resolution is $\mathcal{O}(20\,\mu\text{m})$ if most of the carriers are collected by a single pixel or strip.

On the other hand, it is possible that a particle's track is associated to a cluster of cells, instead of a single one. In this case, the measurements of the pulse-height of the signal induced in each cell of the cluster can be exploited to improve the resolution of position measurement. As an example, let us denote $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ the vector position of two pixels that measure signals with pulse heights h_1 and h_2, respectively, in the reference layer. The best estimator for the particle position is the barycenter of charge:

$$\overrightarrow{v} = \frac{\overrightarrow{v_1}h_1 + \overrightarrow{v_2}h_2}{h_1 + h_2} \tag{1.7}$$

[5] In general, the signal-to-noise ratio (SNR) is defined as the amplitude of a signal divided by the root-mean-square value of its fluctuations [9]. Since noise fluctuations induce current pulses equivalent to those of traversing particles, they are usually expressed in numbers of equivalent electrons, or *Equivalent Noise Charge* (ENC).

In Eq. 1.7, the origin of all the vectors is the interaction point of proton beams. If $h_1 \gg h_2$ the resolution of position measurement in this case is:

$$\sigma \simeq \frac{d}{h_1/ENC} = \frac{d}{SNR} \tag{1.8}$$

where $d = |\overrightarrow{v_2} - \overrightarrow{v_1}|$. Because SNR $\simeq 15 \div 40$, the barycenter technique allows for the resolution to reach few microns.

In the CMS tracking system, the charge collected by a single pixel or strip is measured to improve the *local reconstruction*, i.e. the position measurement in case the induced charge is collected by a cluster of cells.

The tracking volume is surrounded by two calorimeters: the electromagnetic calorimeter (ECAL) and the hadron calorimeter (HCAL).

Electromagnetic Calorimeter. The ECAL [10] uses lead tungstate ($PbWO_4$) crystals to produce scintillation light, which is detected by silicon avalanche photodiodes (APDs) in the barrel region and vacuum phototriodes (VPTs) in the endcap region.

Hadron Calorimeter. The HCAL [11] is a brass/scintillator sampling hadron calorimeter, which uses hybrid photodiodes to detect the scintillation light.

Superconducting Magnet. Both tracking and calorimeters volumes are placed inside a superconducting solenoid, which is 13 m long with an internal diameter of 6 m. It produces a magnetic field of ~ 3.8 T that bends the trajectory of charged particles. This effect is crucial in CMS because from the curved trajectories it is possible to obtain the particle's momentum.

Muon Chambers: The outermost part of CMS is formed by a muon detector. It is organized in four layers of Drift Rubes (DT) in the barrel region and in four layers of Cathode Strip Chambers (CSC) in the endcap region. Resistive Plate Chambers (RPC) are placed both in the barrel and in the endcaps to complement DT and CSC detectors, by adding redundancy. RPCs are able to perform precise time measurements, useful for the trigger system. In the muon chambers, the residual magnetic field is close to 1.8 T.

Triggering System. In CMS, the triggering system is so important that it can be considered as a sub-detector itself. Typically, the size of an event observed in CMS is close to 2 MB. Since the bunch crossing frequency is 40 MHz, it is impossible for CMS to read all the events coming from every single pp collision, since the data flow would be close to 10^5 Gb/s. This bandwidth problem cannot be solved even using links of 1 Gb/s. On top of that, there is also a storage problem: even if it could be possible to read all the events, it would be impossible to store them. Indeed, it would mean to store data from more than 100 M read-out channels every 25 ns. So, in order to be sure to read and store only interesting physical events, the data flow is filtered using a *triggering system* made of two levels. The *Level 1 trigger* (L1) [12] receives data from fast detectors, which are calorimeters and muon chambers. Even if these data have low resolution and granularity, by using fast and simple criteria

L1 manages to decide if a certain event is worth to be sent to the next trigger level in few microseconds (*latency*). If an event passes the L1 cut, it is sent to the second trigger level, which is called *High Level Trigger* (HLT) [13]. This is made of many computers that analyze and filter the events using simplified algorithms. The parallelization helps the HLT to reduce its latency. All the events that pass the second trigger level are permanently stored, ready to be analyzed. After L1, the data flow has a frequency of some tens of kHz, which is reduced to some hundreds of Hz after HLT.

In Fig. 1.7 a schematic representation of how different particles interact with the CMS sub-detectors is shown. For example, an electron can be identified by associating to a shower in the ECAL a track in the tracking system. On the other hand, a charged hadron (i.e., a pion) can be identified by associating to a track showers both in ECAL and HCAL.

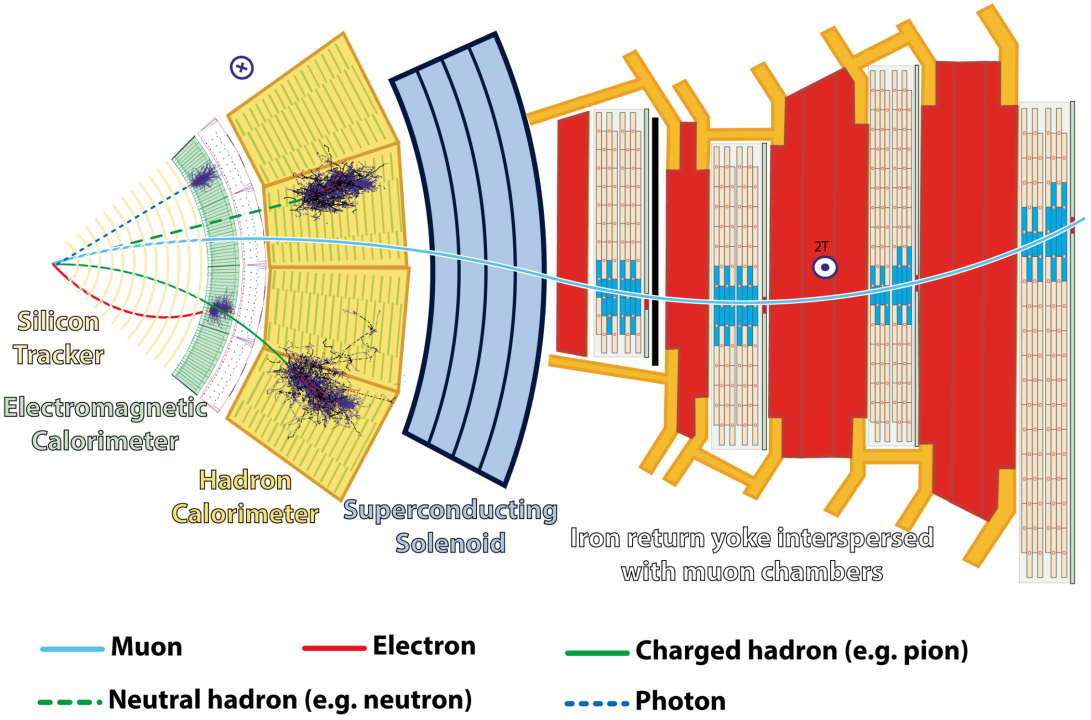


Figure 1.7: Sketch which shows how different particles interact with the CMS sub-detectors.

1.3 High Luminosity LHC

The current plan of LHC consists in a series of long periods of data taking, referred to as Run-I, Run-II, Run-III and so on, interleaved with Long Shutdowns, designated LS1, LS2, LS3 etc. During these shutdown periods, maintenance of the machines or the experiments can be done. As it is shown in Fig. 1.8, the next long shutdown is LS3, which will begin at the end of 2025 [14].

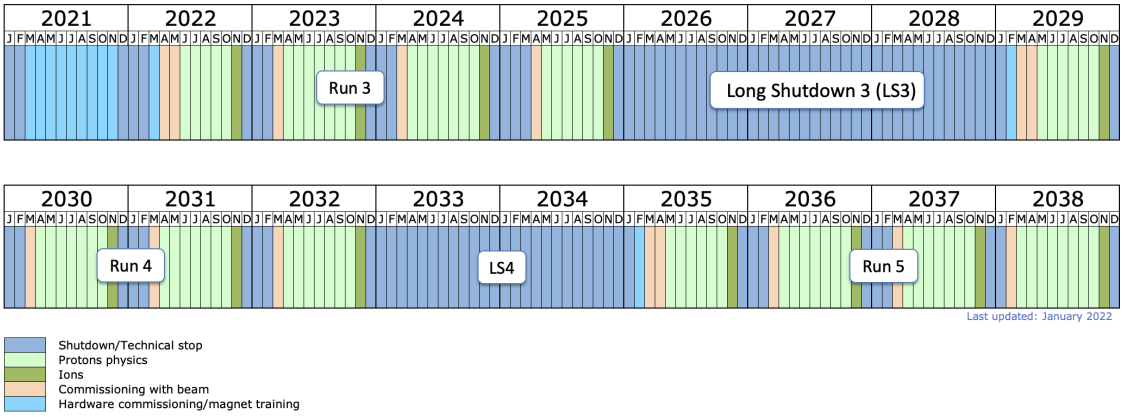


Figure 1.8: LHC long term schedule.

LHC will undergo a major upgrade during LS3. When the operation of the accelerator will resume at the beginning of 2029, the LHC will enter a new era, called *High Luminosity LHC* (or HL-LHC). The goals of this new phase are to reach a peak instantaneous luminosity of $7.5 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$ and to collect an integrated luminosity between $3000\,\mathrm{fb^{-1}}$ and $4000\,\mathrm{fb^{-1}}$ in about ten years of operation.

The increased luminosity will be reached through several improvements concerning beam optics. For example, new superconducting quadrupoles will be installed close to ATLAS and CMS interaction points. These are based on a new Nb_3Sn technology, and they will provide a high magnetic field of 12 T which will improve the bunch focussing. This will reduce both σ_x and σ_y and so luminosity will increase (see Eq. 1.3). In order to be able to fully exploit the physics potential during HL-LHC, all LHC experiments will undergo a major upgrade during Long Shutdown 3.

1.3.1 CMS Upgrade

During HL-LHC, the increased luminosity will be followed by a significant increase in the number of pile-up pp collisions per bunch crossing (BX). In CMS, this number will be on average close to 200/BX, which means that the pile-up will be five times higher than the current value of ∼40/BX. The increased hit rate[6] will lead to higher detector occupancy. In addiction to this, the radiation that the detector parts will be exposed to will reach unprecedented levels. Indeed, in the innermost layer of the CMS detector a fluence of $2.3 \times 10^{16}\,(1\mathrm{MeVn_{eq}})/\mathrm{cm^2}$ and a total ionizing dose of 1.2 Grad are foreseen for the baseline integrated luminosity of $3000\,\mathrm{fb^{-1}}$. As previously said, to cope with this extreme scenario, the CMS detector (as well as all other main experiments) will be substantially upgraded during LS3. This huge update is known as *CMS Phase–II Upgrade*.

A High Granularity Calorimeter (HGCAL) [15] will replace the existing endcap calorimeters. It will feature unprecedented transverse and longitudinal segmentation for both electromagnetic (ECAL) and hadronic (HCAL) compartments. Moreover,

[6] In general, a *hit* occurs when a detector elementary active cell has recorded a signal due to a traversing particle or to spurious effects (e.g. noise).

an innovative detector will be installed in CMS during LS3, that is a Minimum Ionizing Particle (MIP)[7] Timing Detector (MTD) [16]. This device will help in disentangling the approximately 200 nearly-simultaneous pileup interactions that will occur in each BX by precisely measuring the production time of the resulting outgoing particles. The MTD will also provide new capabilities for charged hadron identification and the search for long-lived particles.

In order to deal with higher occupancy, the CMS triggering system will be updated too [17]. It will always be based on a two level scheme, but the L1 trigger will include information from both high-granularity calorimeter and tracking system. The target maximum L1 rate is 750 kHz with a required latency of 12.5 µs.

Furthermore, since both the current Pixel Detector and Strip Detector will be unable to cope with the HL-LHC environment, the entire tracker will be replaced. This new tracker will be divided in two subsystems: the Outer Tracker (OT), based on silicon strip sensors, and the Inner Tracker (IT), based on silicon pixel sensors [18].

The main requirements for the tracker upgrade are the following:

- radiation tolerance and cold operation (up to $-20\,^{\circ}\mathrm{C}$);
- increased granularity;
- reduced passive material;
- provide the L1 trigger system with tracking information (this requirement eventually applies only to the Outer Tracker);
- large readout bandwidth and deep front-end buffers for higher rate and longer latency of the updated L1 trigger system;
- extended coverage up to $|\eta| \sim 4$ to improve the reconstruction in the forward region;
- IT fully accessible for maintenance and part replacement (tests under radiation have demonstrated that operating the innermost layer until the end of HL-LHC program without intermediate replacement is not realistic).

The *Outer Tracker* [19] detector for the HL-LHC consists of six barrel layers and five endcap disks per side. The basic unit of the detector is the so-called *pT module*. In particular, two different types of pT module exist: 2S and PS. Both module types have two closely spaced silicon sensors. The 2S modules has two sensors with micro-strips, while the PS modules is formed by one sensor with micro-strips and one sensor with macro-pixels. The Outer Tracker can be subdivided into four main large structures: the Tracker Barrel with 2S modules (TB2S), Tracker Barrel with PS modules (TBPS) and two Tracker Endcap Double Disks (TEDD).

The main feature of the OT modules is their ability to provide tracking information to the Level 1 trigger (L1). Thanks to the two closely spaced sensors, readout by

[7] A Minimum Ionizing Particle (MIP) is a particle whose mean energy loss rate through matter is close to $\sim 2\,\mathrm{MeV cm^2 g^{-1}}$. This value corresponds to the minimum of the Bethe-Block equation that describes the rate of energy lost by the charged particle through ionization.

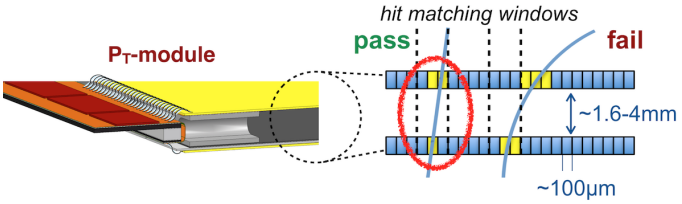


Figure 1.9: The concept of the low transverse-momentum track hit rejection in closely-spaced sensors; the "selection window" defines an accepted pair of high-momentum track hits.

the same electronics, hits from high transverse-momentum track can be identified thanks to their small curvature and routed to the L1 trigger system at each BX. This filtering makes the needed data bandwidth affordable. The concept is sketched in Fig. 1.9.

The *Inner Tracker* for the HL-LHC is based on silicon pixel modules. Those modules feature both two readout chips and four readout chips. The IT is organized into four barrel layers, referred to as Tracker Barrel Pixels (TBPX) plus eight small and four large disks, called respectively Tracker Forward Pixels (TFPX) and Tracker Endcap Pixels (TEPX). A detailed description of the Inner Tracker will be given in the next chapter.

A sketch of the upgraded tracking system is visible in Fig. 1.10.

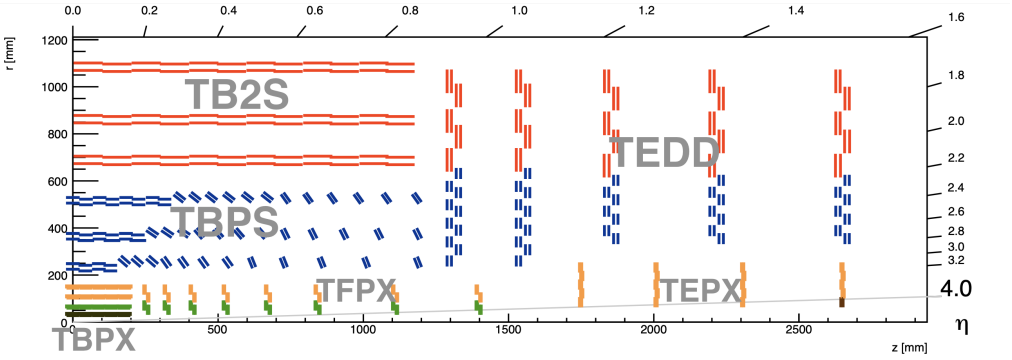


Figure 1.10: Sketch of one quarter of the upgraded tracker layout in r-z view. In the Outer Tracker, the blue lines corresponds to pixel-strip sensors (PS modules), while the red lines corresponds to strip-strip sensors (2S modules). In the Inner Tracker, the green and yellow lines represent the two types of modules: the one with two readout chips and the one with four readout chips, respectively. The black lines identify 3D pixel modules.

The Phase–II tracking system will have a much better granularity and a reduced passive material with respect to the current tracking system resulting in a huge boost on physics performance as demonstrated by simulation.

The track reconstruction efficiency for the Phase–II detector is shown in Fig. 1.11 (left) for simulated $t\bar{t}$ events and two different pile-up scenarios. As another example

of the very good tracking performance, the resolution on the transverse impact parameter as a function of η is shown in Fig. 1.11 (center) for both Phase-I and Phase-II tracker. As can be observed, this performance figure improves by a factor ~ 2 going from the current to the upgraded detector.

One of the most important indicator of the quality of a tracking device at a pp collider is the ability to perform track reconstruction in a dense environment such as the core of a jet[8]. The resolutions of jet-related observables depend on the possibility to distinguish the tracks belonging to the jet with respect to pile-up tracks accidentally close-by. The tracking efficiency as a function of the distance from the jet axis is shown for the Phase–I and Phase–II tracker detectors in Fig. 1.11 (right). In this case, the distance is defined as follows:

$$\Delta R \;=\; \sqrt{(\Delta\eta)^2 \;+\; (\Delta\phi)^2} \tag{1.9}$$

where ϕ is the trajectory angle in the x-y plane, measured from the y axis. The Phase–II tracking efficiency at low ΔR is much bigger with respect to the efficiency of the current detector.

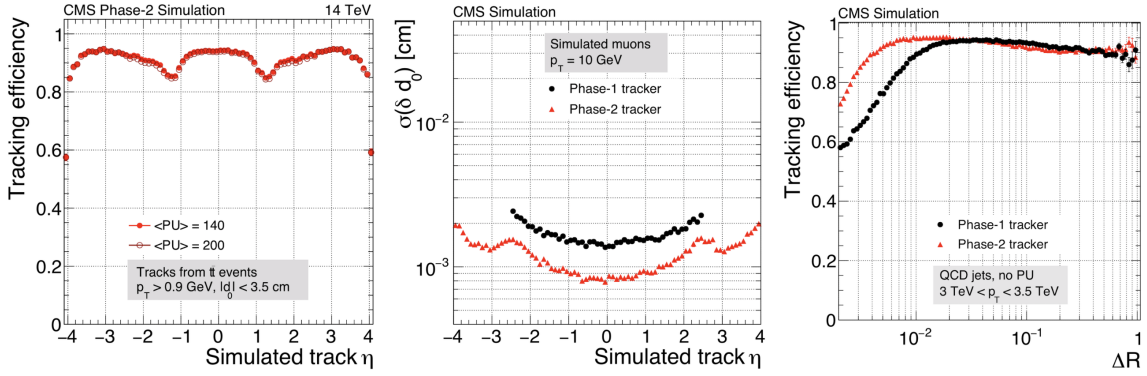


Figure 1.11: Left: Phase-II tracking efficiency on simulated $t\bar{t}$ events at pile-up 140 and 200 as a function of $|\eta|$; Center: muon transverse impact parameter resolution as a function of $|\eta|$ for Phase–I and Phase–II trackers; Right: track reconstruction efficiency as a function of the distance ΔR from the jet axis for Phase–I and Phase–II trackers.

[8]In particle physics, a jet identifies the particles originating from the hadronization of a parton that are tipically produced in a narrow cone.

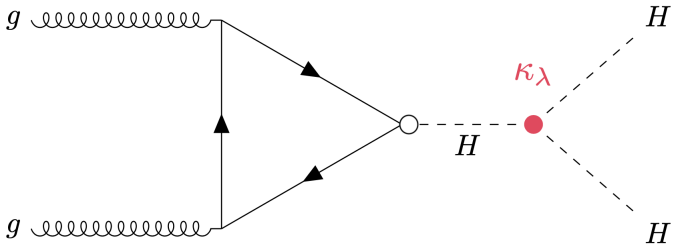


Figure 1.13: Feynman diagram of the Higgs trilinear coupling.

where $\lambda_{\mathrm{HHH}}^{\mathrm{SM}}$ represent the Standard Model prediction.

The main production channel of the double Higgs final states is the gluon fusion and its Feynman diagram is shown in Fig. 1.13. Since the cross-section of this process is only 35 fb at 14 TeV, the increased luminosity at HL-LHC will largely help to study the Higgs trilinear coupling so to be able to access this very important sector of the Standard Model where deviations due to new phenomena are to be expected.

The physics opportunities of the HL-LHC do not concern just Higgs boson. Indeed, the increased luminosity will offer new possibility to test BSM scenarios, motivated by the long-standing problems such as the Electroweak naturalness, dark matter (DM) and neutrino masses. All these hint to the existence of new particles to be searched for at HL-LHC thanks to the much larger statistics and the upgraded detectors. For example, a very interesting scenario is the search for a *dark photon* (A'), that could be one of the "portals" between the visible and the dark sectors.

CHAPTER 2

THE CMS INNER TRACKER FOR HL-LHC

As it has been said in the previous chapter, during the long shutdown in 2026-2029 (LS3) the entire CMS Tracker will be replaced in order to manage the new HL-LHC environment. Since it is relevant for this work, a detailed description of the CMS Phase-II **Inner Tracker** will be given.

A quarter of the mechanical implementation of the inner tracker is shown in Fig. 2.1. The IT that will be used during HL-LHC is formed by ∼4200 hybrid pixel modules with either 1×2 or 2×2 readout chips (ROC), for a total of $\sim 4.9\,\mathrm{m}^2$ silicon pixel detectors. The total number of readout channels is close to 2 billion.

The view of the typical IT module is shown in Fig. 2.2. Each module consists of a sensor bump-bonded to the ROC chips and a High-Density Interconnect (HDI) to route signals and power.

The elementary detector cell consists of a single pixel of the sensor attached to the corresponding readout channel. For this reason, in the following, *cell*, *pixel* and *channel* are used as synonyms.

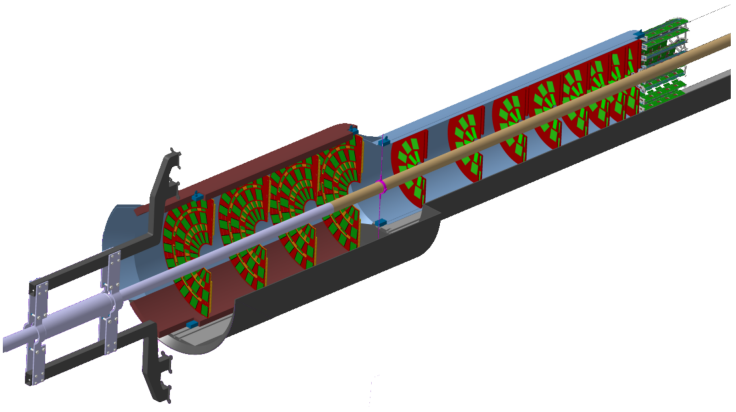


Figure 2.1: Sketch of the inner tracker structure, showing the various subparts.

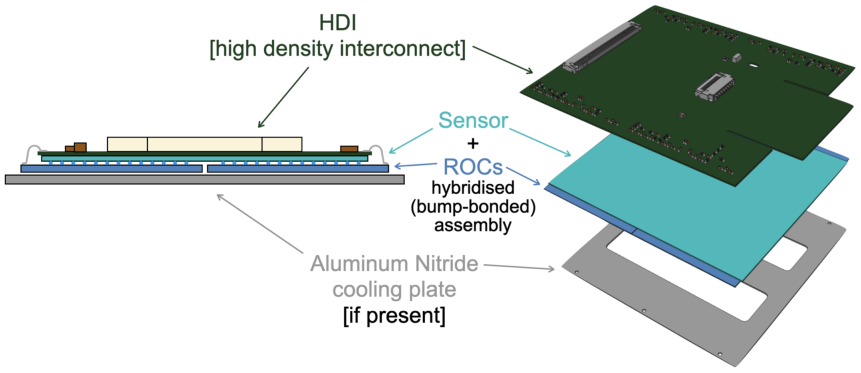


Figure 2.2: Transverse (left) and exploded (right) view of a typical 2×2 module (the transverse view is not in scale).

2.1 CMS IT Read Out Chip

The readout chip (ROC) required for the Inner Tracker is a technological frontier device that needs to withstand 1.2 Grad of total ionizing dose and a maximum hit rate of 3.2 GHz/cm^2. It has been developed by the RD53 Collaboration, which is a joint ATLAS-CMS effort. The first complete pixel chip was called RD53A and was submitted in 2017. It has been used extensively over several years within the R&D program. In particular, RD53A has been used to test and qualify numerous different pixel sensors and in a large set of irradiation campaigns, to get a good understanding of the long-term function and reliability of such a complex chip covering. RD53A had three different analog front ends (AFE): a Synchronous AFE, a Linear AFE, and a Differential AFE [21] (Fig. 2.3). This choice has been made in order to determine the most appropriate implementation of the final production chips.

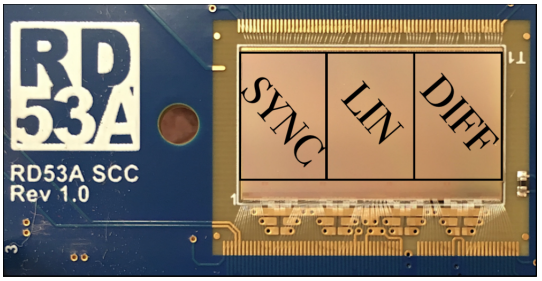


Figure 2.3: Photograph of the RD53A chip with indication of the three AFEs regions.

The next generation RD53 chip, named RD53B, are full size chips made specifically for each experiment: RD53B-ATLAS (2020), also known as ITK-v1, and RD53B-CMS (2021), also known as CMS Read out Chip, version 1 (CROC-v1 or CROC). The main parameters of CROC-v1 are given in Table 2.1 [22].

The chip is designed to operate fully bound to and in sync with the 40 MHz bunch crossings of the LHC. This clock is encoded within the 160 MHz stream that is provided to the chip together with commands and trigger.

In general, the RD53B chip consists of a pixel matrix and a chip bottom [23]. The

Parameter	Value
Max. hit rate	3 GHz/cm^2
Trigger Rate	750 kHZ
Trigger Latency	12.5 µs
Bump bonding pitch	50 x 50 µm^2
Pixel Array	432 x 336 pixels
Chip Dimensions	21.6 x 18.6 mm^2
Min. Threshold	600 e$^-$
Radiation Tollerance	1 Grad over 10 years
Power delivery	Serial Powering
Power	<1 W/cm^2
Clock & Data Recovery	Clock and Command/Trigger stream: 160 MHz; Internal Clocks: 160 MHz, 64 MHz, 640 MHz and 1.28 GHz

Table 2.1: List of design requirements and chip specifications of the CROC-v1.

pixel matrix is the region that will be physically coupled with the sensor, and it is formed by identical 8 by 8 pixel cores repeated in columns and rows. A core is physically $400 \times 400\,\mu m^2$. The number of core columns and rows determine the size of the chip, that is different from CMS and ATLAS.

Each core contains 64 cells, each operating as an independent read-out channel with its own front end. In a core, we can define 16 so-called analog islands with four analog inputs each, which are embedded in a "digital sea". The conceptual depiction of RD53B framework is shown in Fig. 2.4.

The chip bottom (also known as *periphery* and shown in Fig 2.5) is placed in the bottom of the chip and contains all global and digital circuitry needed to bias, configure, monitor and readout the chip. In the chip periphery, all the analog building blocks are grouped in a macro-block called Analog Chip Bottom (ACB). The ACB is surrounded by a second block, called Digital Chip Bottom (DCB), which implements the Input, Output and Configuration digital logic.

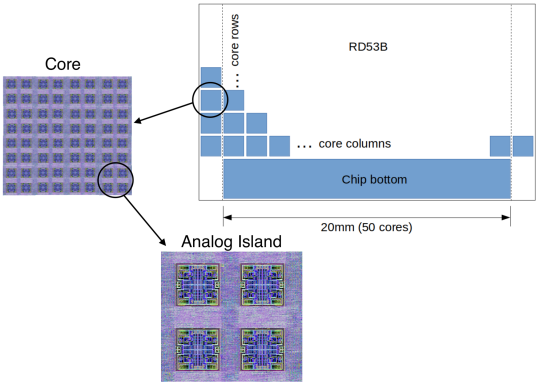


Figure 2.4: Conceptual depiction of RD53B framework.

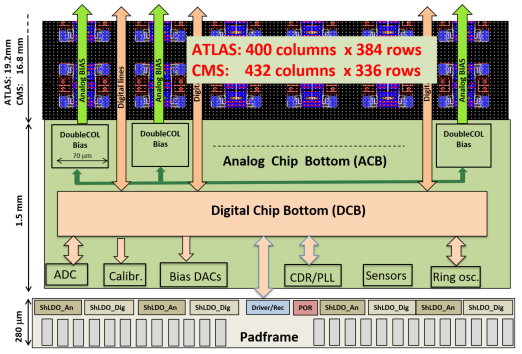


Figure 2.5: RD53B floor-plan, functional view.

Apart from the dimension and some minor features, the main difference between

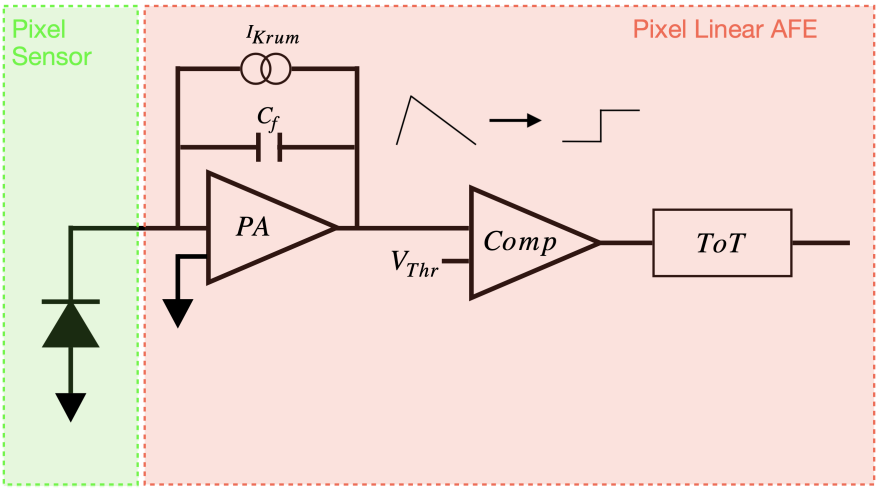


Figure 2.6: Simplified schematic of CROC-V1 linear analog front-end.

ITK-v1 and CROC-v1 is that the former has a differential analog front end, while the latter uses a linear analog front end. In Fig. 2.6 a simplified scheme of the CROC linear analog front end is shown.

The charge induced by a traversing particle in a pixel sensor (here represented as a reverse biased diode) is transferred to the pixel chip and then integrated by a pre–amplifier stage (PA). The PA stage of the Linear Front End is formed by a charge sensitive amplifier (CSA) with Krummenacher feedback. The Krummenacher current (I_{Krum}) generated in the feedback plays a crucial role in the linear AFE since it provides a linear discharge of the CSA feedback capacitance (C_f) and reduces the contribution of the leakage current coming from the DC coupled sensor. Downstream the preamplifier stage, the signal is characterized by a fast-rising edge and by a slowly decreasing falling edge. The pulse width is proportional to deposited charge, while the return to baseline time depends on Krummenacher current: the higher is I_{Krum}, the faster is the return to baseline [24]. The amplified signal is then transformed into a digital hit signal by a comparator with a programmable threshold, V_{Thr}. The value of V_{Thr} is determined by a Global Threshold bias together with a four – bit threshold adjust per channel, to compensate for threshold dispersion among channels. The dynamic range of the threshold adjust is also configurable by a global programmable bias. A more detailed description of the Front-End parameters is given in paragraph 2.1.1.

A key feature of the Inner Tracker is the rather low detection threshold that will be set around 1000 e^- thanks to the exceptional specification of the CROC. In fact, since the typical charge released by a MIP is $\mathcal{O}(10\,000\,e^-)$ and the typical noise is $\mathcal{O}(100\,e^-)$, a threshold of $\sim$1000 e^- ensure high detection efficiency and a good local reconstruction with negligible channel occupancy due to noise.

In order to improve the resolution of position measurements if the charge induced by a traversing particle is collected by a cluster of cells (see paragraph 1.2) the charge Q induced in a single pixel needs to be measured. In particular, Q can be obtained measuring the time that the signal stays above the threshold, by counting how many

times the comparator output is up at a clock edge. This parameter is known as *Time over Threshold* (ToT) and can be measured using the block named *ToT* in Fig. 2.6. The ToT is measured in units of 25 ns, so if a signal has a ToT of 4 it means that the signal was above the threshold for 100 ns.

The actual ToT depends on Krummenacher current. Indeed, the higher is I_{Krum} the faster is the return to baseline of the signal and so the smaller is the time during which the signal itself stays above the threshold. The effects of Krummenacher current on signal shape and ToT are shown in Fig. 2.7.

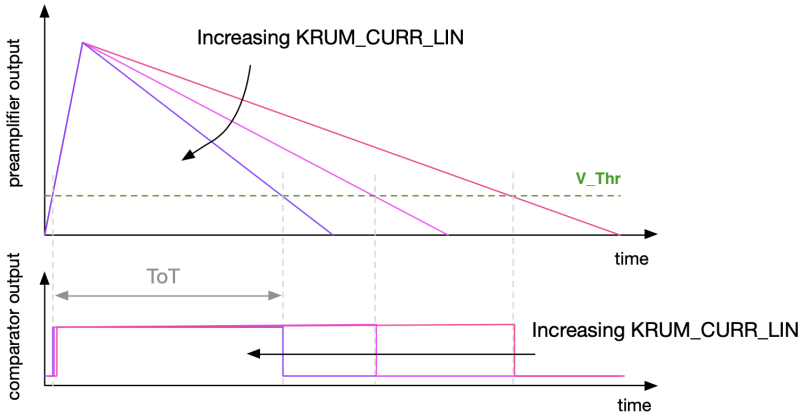


Figure 2.7: Effects of Krummenacher current on signal waveform and ToT.

A more detailed schematic of the CMS analog front-end (up to the comparator) is shown in Fig. 2.8.

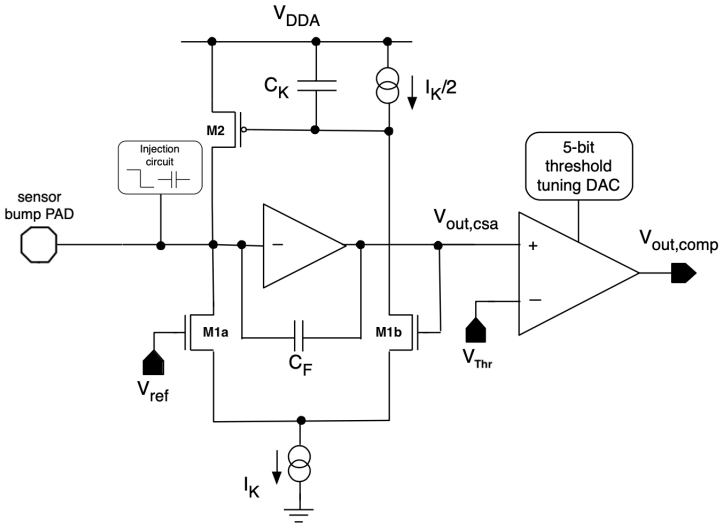


Figure 2.8: Detailed schematic of CMS Analog Front End.

2.1.1 Front-End Registers

The registers are the DAC[1] settings for the CMS AFE, which are used to configure the linear front end. For example, the comparator threshold V_{Thr} can be set as follows:

$$V_{Thr} = V_{GDAC} - V_{LDAC} \times TDAC \tag{2.1}$$

where V_{GDAC}, V_{LDAC} and TDAC are defined using programmable bias.

The AFE main registers are:

- *GDAC_LIN*
- *LDAC_LIN*
- *KRUM_CURR_LIN*
- *PIX_MODE*
- *CalibrationConfig*
- *VCAL_MED* and *VCAL_HIGH*
- *TRIM_VREFA* and *TRIM_VREFD*

A more detailed description of all these registers is given below.

GDAC_LIN: This register sets the Global Threshold of the Linear AFE, named V_{GDAC} in Eq. 2.1. It is generated in a particular kind of unit known as V_{cal} unit. With a good approximation, one V_{cal} is the voltage seen between feedback capacitance's armors if they are charged with a charge Q equal to $5\,e^-$. Please note that this conversion factor can slightly change by changing the sensor type (more details in paragraph 3.2.2).

LDAC_LIN: The value of V_{LDAC} (see Eq. 2.1) is set using the global register *LDAC_LIN*. Even if the value of *GDAC_LIN* is the same for all the channels, the actual threshold can have a dispersion due to the fact that channels do not share the same ground. In order to reduce this dispersion, the factor $V_{LDAC} \times TDAC$ is added in Eq. 2.1. In particular, TDAC is a number from 0 to 32 which changes cell by cell, and it is determined during the calibration procedure (see paragraph 2.4.2), while V_{LDAC} (generated in V_{cal} units) determines the dynamic range of the threshold adjust and the minimum step you can take.

KRUM_CURR_LIN This register sets the current in the Krummenacher feedback used to linearly discharge the preamplifier feedback capacitance. Increasing *KRUM_CURR_LIN* results in a faster preamplifier return to baseline (seen Fig. 2.7). According to the value of *KRUM_CURR_LIN*, three types of discharge can be distinguished:

1. *KRUM_CURR_LIN* $= 70$ *(Slow Discharge)* $\rightarrow$ *ToT* $= 133\,ns$ *if* $Q = 6000\,e^-$
2. *KRUM_CURR_LIN* $= 80$ *(Normal Discharge)* $\rightarrow$ *ToT* $= 25\,ns$ *if* $Q = 1500\,e^-$

[1] Digital to Analog Converter

3. *KRUM_CURR_LIN = 190 (Fast Discharge)* $\rightarrow$ *ToT* = 25 ns *if* $Q = 3000\,e^-$

PIX_MODE: This register sets the Hit Sample Mode. There are two modes that can be chosen: Synchronous mode (*PIX_MODE* = 10) and Asynchronous mode (*PIX_MODE* = 2). A detailed description of these modes is given in Chapter 4.

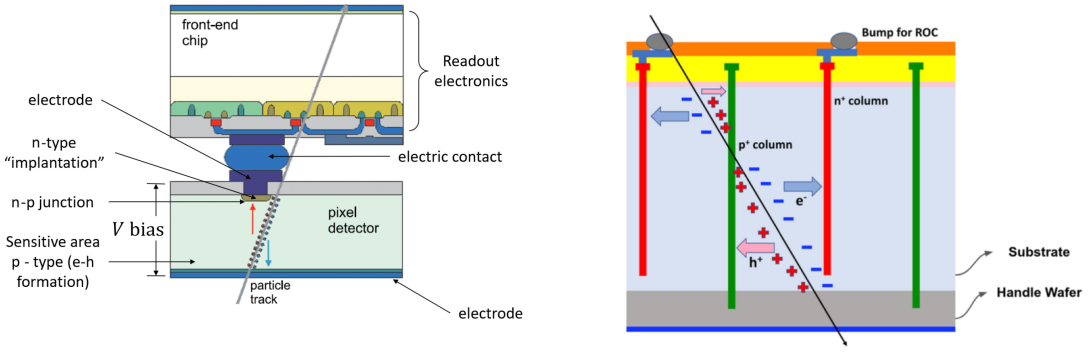
CalibrationConfig: This register is a 8 bits register: the firsts 6 bits are used to set the Fine Delay, which is the relative delay between the 40 MHz clock and the charge calibration injection, the seventh bit is used to select the mode of analog injection, which can be Uniform or Alternating and finally, the last bit can be used to select a digital (1) or an analog (0) injection. More details concerning charge injection are given in paragraph 2.4.

VCAL_MED and VCAL_HIGH: These registers are used to set the value of the voltages used for calibration injection. *VCAL_MED* and *VCAL_HIGH* are generated in V_{cal} unit and their difference is defined as ΔV_{cal}. Please note that ΔV_{cal} is the difference between two registers, and it is measured in V_{cal} unit (the magnitude must not be confused with its unit).

TRIM_VREFA and TRIM_VREFD: These two registers are used to set the supply voltage of both analog and digital part of the read-out chip of each channel.

2.2 Different types of Pixel Sensors

The great majority of the CMS Phase-II inner tracker will be equipped with thin *planar* n-in-p sensors. However, for the innermost barrel layer ("barrel layer 1") a different sensor technology will be used. In particular, 3D pixel sensors have been chosen, thanks to their intrinsically lower bias voltage also after irradiation. Moreover, the 3D technology is more suitable than planar technology to deal with the increased radiation damage that will characterize the layer 1 environment during HL-LHC. For both sensor technology, the bulk thickness is 150 µm and the pixel cell size is 2500 μm^2. The latter is six times smaller with respect to the current Phase-I pixel sensors. The schematic views of a planar sensor and of a 3D sensor are shown, respectively, in Fig. 2.9a and 2.9b.



(a) (b)

Figure 2.9: Schematic view of a planar sensor (a) and of a 3D sensor (b).

As can be noted by looking at Fig. 2.9a, in a planar sensor the electrodes are thin superficial implants on the surface of the substrate. The main difference between a planar and a 3D sensor is that in the latter the electrodes are cylindrical and penetrate deeply into the substrate (see Fig. 2.9b). This new configuration allows for many advantages. Since the distance between electrodes is smaller in 3D sensors than in planar sensors, the first advantage is linked to power consumption: the voltage needed to deplete 3D sensors is smaller with respect to planar ones while the leakage current, essentially dependent on the depleted volume, remains the same. The short distance between electrodes can also be exploited to reduce the effects of the radiation damage. In fact the distance that the particle ionization charges must travel is shorter in 3D sensors (with a consequential reduced drift time). Hence the probability that these charges are being absorbed by the radiation-damage induced defects in the silicon lattice is lower.

However, 3D sensors are a relatively new technology and the production process is more complicated, with consequences on costs and production yield. This is the reason why 3D sensors will be used only in IT barrel layer 1. All the 3D and planar modules used in this work are manufactured by the *Fondazione Bruno Kessler* (FBK) [25] and the *Hamamatsu Photonics K.K.* (HPK) [26] respectively.

2.3 Serial Powering

Serial powering is the only possible powering scheme compatible with the physics requirements of HL-LHC. Indeed, it provides the 50 kW necessary for the Inner Tracker, while minimizing the material budget, given the reduced number of powering cables. This is crucial in a tracking system because multiple scattering and particle conversions in passive material could adversely affect tracking performance. Thanks also to the serial power concept, the estimated material budget of the upgraded tracker is significantly reduced with respect to the current Phase-I tracker (in particular around $|\eta| = 1.5$). The comparison Phase–II vs Phase–I material budget is shown in Fig. 2.10.

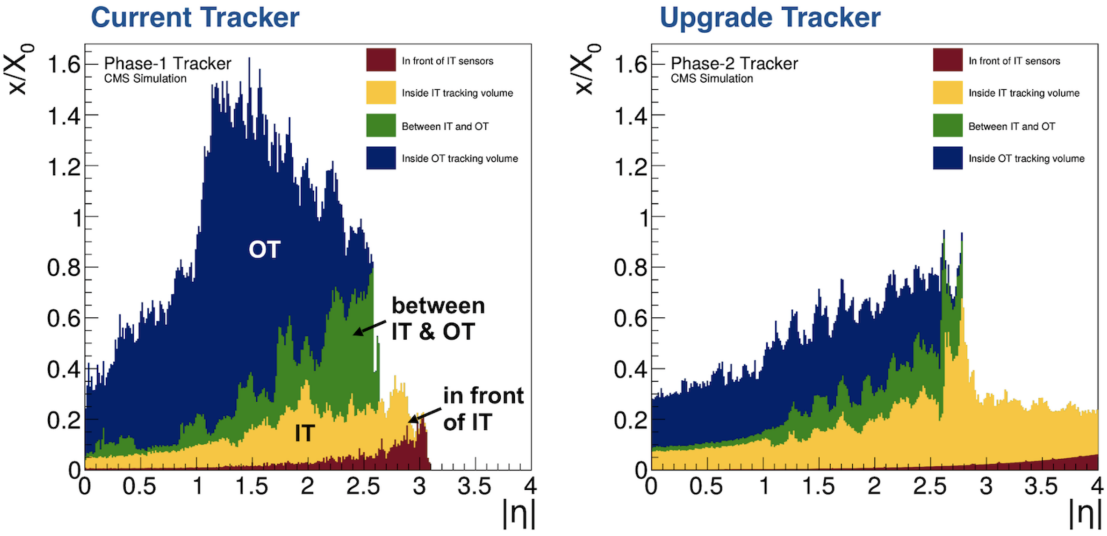


Figure 2.10: Material budget vs. $|\eta|$ profiles for the Phase-I (left) and Phase-II (right) tracker detectors; the material is given in units of radiation length.

The CMS ROCs have been made specifically to be powered within serial power chains, made of 8 to 12 modules. A functional sketch of such a chain is visible in Fig. 2.11. All the modules in the chain receive the same current (by construction) and the voltage is equally shared if all elements represent the very same and constant load. This goal can be reached thanks to the *ShuntLDO* (SLDO), which is an integrated circuit of the ROC that implements the combination of a Linear Drop-Out voltage regulator (LDO) and a shunt element. Each CROC have independent SLDOs for analog and digital domains to prevent digital logic activity to interfere with the sensitive analog parts.

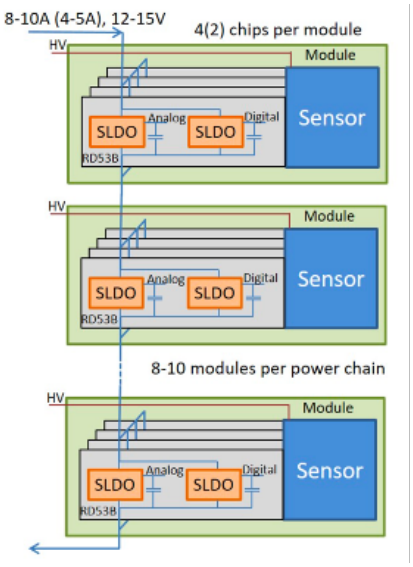
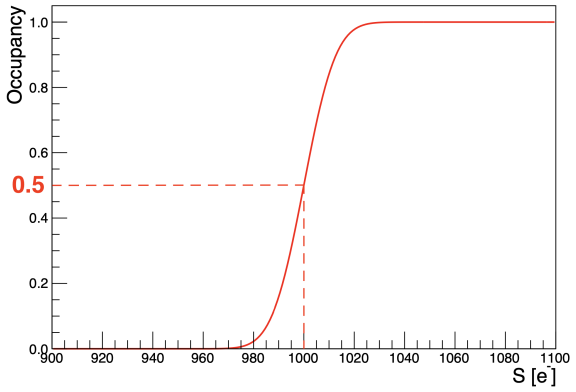


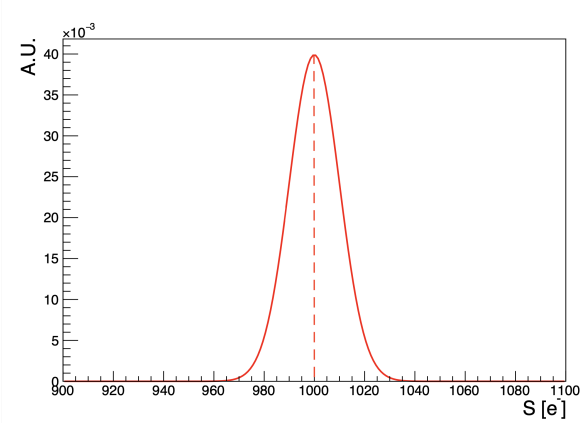
Figure 2.11: Functional sketch of IT modules within a serial power chain.

2.4 Channel Threshold

In order to better understand the results described in my thesis, a detailed description of the generalized concept of "threshold" is needed. To do so, suppose to focus on a single channel. The channel *Occupancy* as a function of the particle induced charge of magnitude S can be defined as the number of times the signal is recorded over the number of times it is induced in the channel. In ideal conditions, the occupancy as a function of S is expected to be a step function that gets to 1 when the resulting voltage at the comparator output is larger than the comparator threshold. However, since the signal is characterized by noise fluctuations, the occupancy profile has a smoother behavior around the threshold, i.e. a *S-Curve*. An example of this kind of curve is given in Fig. 2.12a. In general, the actual threshold can be defined as the signal value corresponding to the 50 % occupancy value. Moreover, the noise can be estimated as the standard deviation of the first derivative of the S-Curve (as shown in Fig. 2.12b).



(a)



(b)

Figure 2.12: (a) S-curve for a channel: the value of S which has 50 % occupancy can be defined as the actual threshold of the channel; (b) first derivative of the S-Curve: the standard deviation of this distribution is the channel noise.

The procedure previously described can be used in CROC modules to measure the actual threshold of each channel. In general this actual threshold does depend on the operating conditions of the CROC. To perform this tasks in a easy and controlled way and for other calibration purposes, the CROC is equipped with an injection circuit which induces a known signal S in the channels emulating the signal that would be induced by particles in the sensor.

2.4.1 Calibration injection circuit

The calibration injection circuit provides in each channel an injection of a charge Q obtained by charging a reference capacitance, called C_{inj}, with a voltage V_{inj}. The value of V_{inj} is obtained considering the difference between two voltages: $V_{cal-High}$ and $V_{cal-Med}$, whose value can be set using the registers *VCAL_MED* and *VCAL_HIGH* (as described in paragraph 2.1.1). Having two voltages allows a precise differential voltage which is independent of local ground drops that each channel can have. The nominal value of C_{inj} is 8.02 fF. The injection circuit is implemented in each front-end. Injections are performed in all enabled pixels, which location and number can be configured.

In general, the calibration circuit can be configured by using two registers: *CalibrationConfig* and *CalCommand*. In particular, the latter is used to set the value of a *Coarse* Delay that selects the clock edge, i.e. the clock cycle, after which the injection must be performed. To be noted that in the operating experiment a 40 MHz clock cycle will correspond to a bunch crossing, identified with a BX ID. For this reason the Coarse Delay or bunch crossing ID are often used as synonyms. On the other hand, as described in paragraph 2.1.1, the register CalibrationConfig can be used to set a *Fine* Delay which determines when the injection is performed within the selected clock cycle. In fact the Fine Delay ranges from 0 to 25 ns. The difference between *Coarse* and *Fine* Delay is sketched in Fig. 2.13.

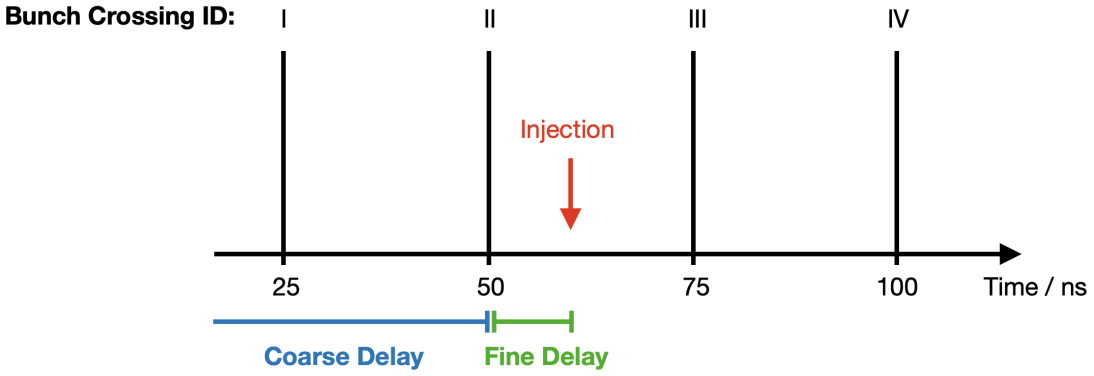


Figure 2.13: Sketch that shows the difference between *Coarse* and *Fine* Delay. In this case, the register CalCommand is set to select the bunch crossing with ID = 2, while the Fine Delay determines when to perform the injection between BX with ID = 2 and BX with ID = 3.

The register CalibrationConfig can also be used to select between *Uniform* and *Alter-*

nating mode. In the former mode, pulses of the same size are injected in all enabled channels, while in the latter, pulses of different size are injected simultaneously in adjacent channels.

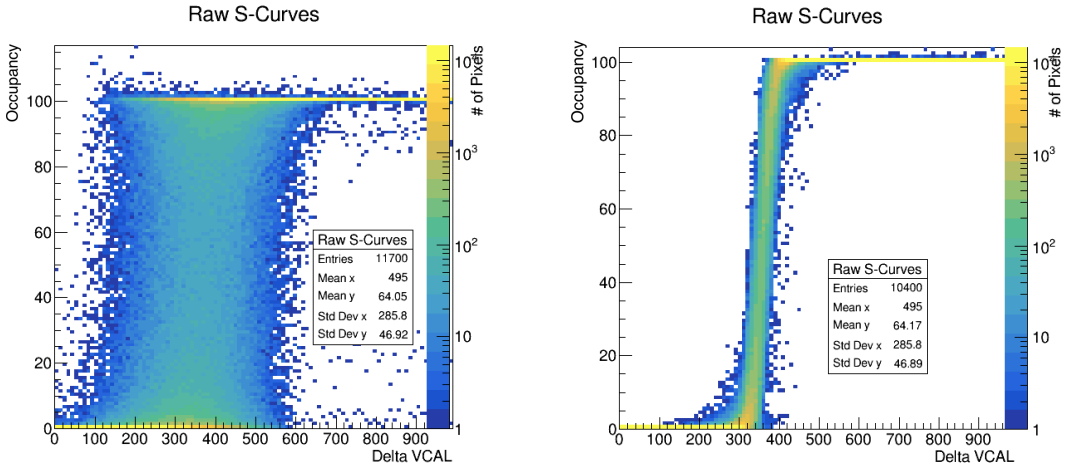
Please note also that we can distinguish between *Analog* and *Digital* injections. In particular, an Analog injection involves the preamplifier stage, while a Digital injection bypass the preamplifier and triggers the comparator. The register CalibrationConfig can be used to select between those two types of injection. In this work, only Uniform Analog injections have been used.

2.4.2 Threshold adjust procedure

In paragraph 2.1.1, the threshold of each pixel has been defined using Eq. 2.1. Suppose to take into account only the first part of that equation:

$$V_{Thr} = V_{GDAC} \tag{2.2}$$

where V_{GDAC} is set using the global register named *GDAC_LIN*. If the calibration circuits are used to obtain the S-Curves of each channel, a plot as in Fig. 2.14a is obtained, where a large threshold dispersion is visible. This is due to channel-to-channel ground differences. In order to have a uniform sensitivity among channels, the second part of Eq. 2.1 must be taken into account. As it can be seen in Fig. 2.14b, the trimming procedure reduces the spread between S-Curves [27]. In the latter procedure, the TDAC value is scanned, and its proper value is selected for each channel. The proper TDAC value is the one for which the target threshold corresponds to the 50% occupancy value.



(a) (b)

Figure 2.14: (a) S-Curves without trimming the DACs; (b) S-Curves after tuning trimming the DACs.

CHAPTER 3

SETUP FOR BENCH TESTING

3.1 CROC Module and DAQ System

3.1.1 The module test board

In my thesis, the performance of the CMS ROC are studied using test modules formed by a sensor made of 145 152 pixels bonded to a single read-out chip on a board equipped with high-density interconnect for the powering, readout and control.

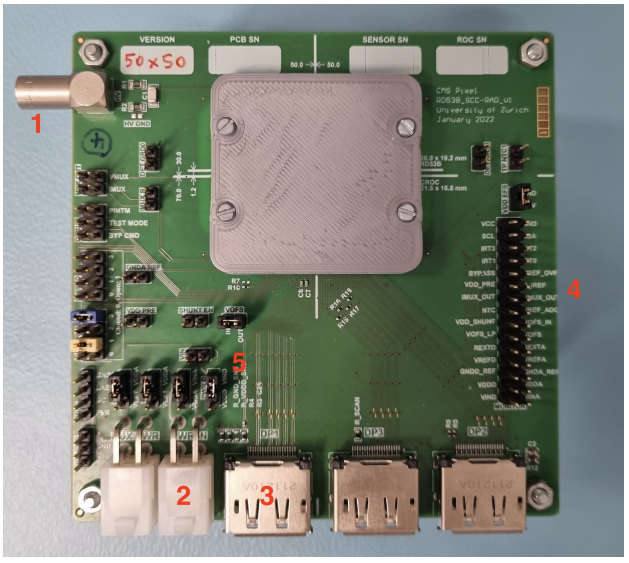


Figure 3.1: Layout of a typical test module. In this figure, the chip (hidden by the protection cover) is bonded with a 50×50 µm^2 planar sensor.

The general layout of a typical module is shown in Fig. 3.1, where several parts of the chip are highlighted with different numbers:

1. HV Lemo used to bias the sensor bonded to the read-out chip.

2. LV Molex used to power the readout chip with low voltage (typical range: $1.6 - 1.7$ V).
3. Display port used for read-out (for more details, see paragraph 3.1.2).
4. Terminal header, which can be used to probe and verify several chip operating voltages.
5. Jumpers used to configure the powering mode.

The modules are powered using a CAEN Universal Multichannel Power Supply, both for High Voltage and Low Voltage. The value of the HV, in the range of 30 V to 100 V, depends on the sensor bonded to the chip (see paragraph 2.2), while the value of the LV is $1.6 - 1.7$ V for all the chips. It is worth to be noted that in this work, "bare" chips have also been tested. A bare chip is just a chip with no connected sensor.

The modules (with both planar and 3D sensors) used in this work are listed in Table 3.1.

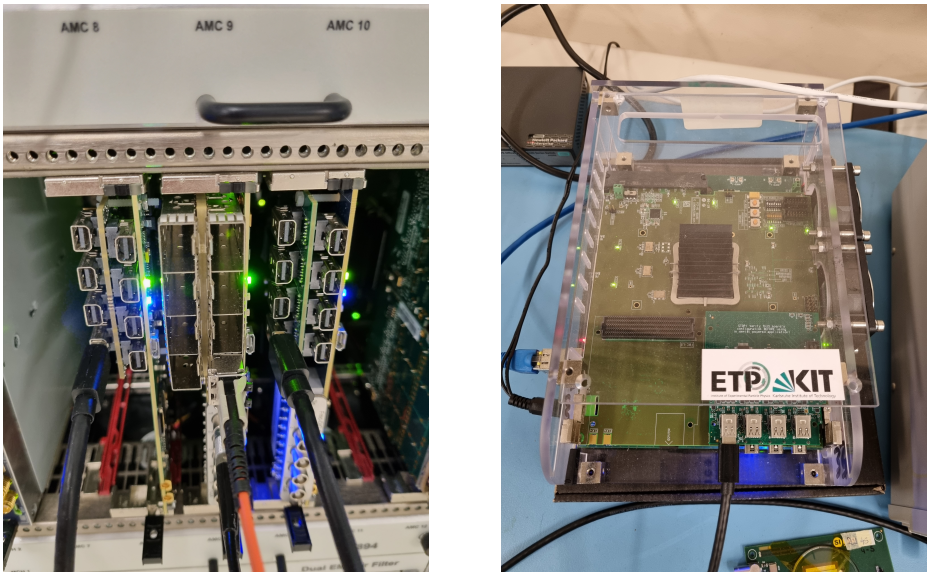
Sensor Type	**Name**	**Pitch / μm**	**Thickness / μm**
3D	FBK – w6_5-1	25x100	150
	FBK – w13_5-1	25x100	150
Bare Chip	201.45	-	-
	201.65	-	-
Planar	HPK – 719	25x100	150
	HPK – 704	50x50	150
	FBK – w5_1-6	25x100	150

Table 3.1: Table with all the modules used in this work. Each module is described by a name and, when possible, by the pitch and the thickness of the sensor.

3.1.2 DAQ System

The DAQ system used in this work to read out the CROCs is called Ph2ACF, which stands for "Phase–II Acquisition and Control Framework". Ph2ACF is a system developed by the CMS collaboration. The hardware consists of a custom board, named FC7, based on a Kintex7 FPGA (Field Programmable Gate Array). Usually, the FC7 is placed in a crate, but for bench testing, it can also be placed in a specifically designed nano-crate (Fig. 3.2). The communication between FC7 and PC is granted by an Ethernet link. The FC7 can host two FMC[1] cards, which allow electrical communication with the CROC through a Display Port Cable. The firmware developed for the FC7 is called $IT - \mu DTC$ (Inner Tracker micro Data Trigger and Control) and it is entirely written in VHDL.

[1]The acronym FMC stands for FPGA Mezzanine Card, and it is an industrial standard that defines I/O mezzanine modules with connections to an FPGA.



(a) (b)

Figure 3.2: (a) FC7s in Crate; (b) Single FC7 in the so-called nano-crate.

The Ph2ACF software is written in C++ and uses ROOT [28] for data analysis. It allows to configure the CROC DAC registers and provides a full set of procedures which can be used to calibrate and characterize the chip. The main procedures and their purpose are listed below:

- **RegReader** : This tool will load on the chip the registers value chosen in the configuration file.
- **ThresholdScan** : The ThresholdScan tool injects in each pixel signals configurable from 0 to 1000 V_{cal}, measuring for each signal the occupancy. So, the ThresholdScan tool produces the S-Curve plots and the distribution of the threshold among pixel matrix. The value of the threshold obtained using this procedure is referred to as the *Actual Threshold*.
- **ThresholdEqualization** : This tool reduces threshold dispersion by searching the optimal TDAC value for each channel.
- **GlobalThresholdTuning** : The GlobalThresholdTuning tool finds and then sets in the configuration file the value of GDAC which corresponds to a given value of the threshold in V_{cal} units. The threshold set using this procedure is referred to as *Target Threshold*.
- **NoiseScan** : This scan allows to identify, and eventually mask, the noisy pixels. A pixel is defined as "noisy" if, without injection, it sees more than 100 hits over 10^6 triggers, that is, if its noise occupancy is bigger than 10^{-4}. By definition, the number of noisy pixels is a threshold dependent parameter: the lower is the threshold, the higher is the number of noise hits, and so the higher is the probability that a pixel is marked as noisy.
- **StuckPixelScan** : StuckPixelScan identifies, and eventually masks, stuck pix-

els. To be labelled as "stuck", a pixel must have an occupancy equal to 100% or lower than 90% when a really high signal (more than 15 000 e^-) is injected. As the number of noisy pixels, also the number of stuck pixels (slightly) depends on actual threshold. Indeed, the lower is the threshold, the higher is the probability that a pixel is "always on" (100% occupancy).

- **SignalScan** : This procedure allows to obtain the shape of a signal induced in the sensor by a charge Q. The value of the charge can be selected modifying the value ΔV_{cal}. Fig. 3.3 shows the output of *SignalScan* when $Q = 10\,000\,e^-$.

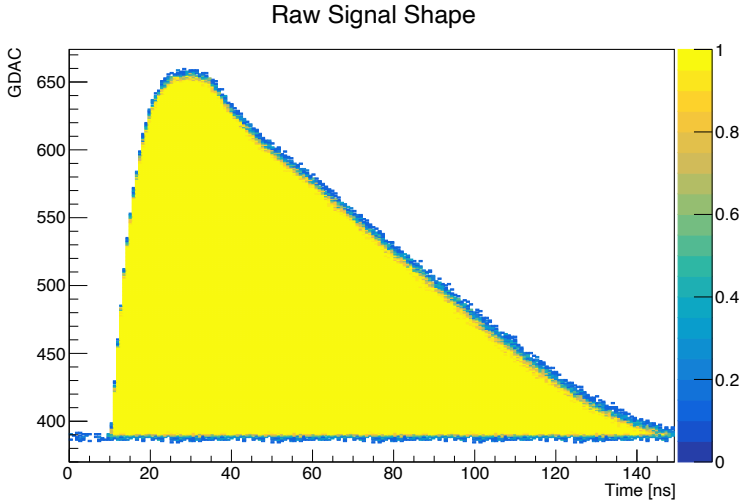


Figure 3.3: Example of output from *SignalScan*. In this case, the injected charge is equal to 10 000 e^- and the module used is a bare chip (20145) with Fast Discharge.

3.2 Chip properties and conversion factors

In this paragraph, some preliminary measurements will be described. The results of these measures will underlie any other measurements subsequently described.

3.2.1 Threshold stability as a function of the number of injected pixels

In paragraph 2.4.1 the injection circuitry for each pixel has been described. However, if during a scan executed by the Ph2ACF software is injected one cell at a time, the scan will require a lot of time. So, to reduce the scan duration, the injections can be parallelized by injecting several cells at the same time.

It must be noted that this procedure can slightly modify the value of the actual threshold. In order to study this behavior, the first thing to do is to equalize the threshold in order to minimize threshold dispersion among pixels. Once equalization is done, the procedure described below has been followed:

- **Set to N the number of injected pixels:** The number of injected pixels per column can be set by changing the mask of the Analog Injection.

- **Run a ThresholdScan:** In this way, the actual threshold value for a number N of injected pixels can be obtained.

The Fig. 3.4a shows the results obtained by changing N from 1 to 15. By looking at Fig. 3.4a it can be noted that injecting few pixels per column results into a threshold scan lasting several minutes. The scan time does not improve further beyond 8 pixels per column. Therefore, if not stated differently, the scans performed in this work are done injecting 8 pixels per column. In Fig. 3.4a the measurement has been done changing the number of injected pixels from 1 to 15 (red line) and from 15 to 1 (blue line), in order to verify the repeatability of the measurement.

As it has been said previously, Fig. 3.4b shows that the threshold linearly depends on the number of injected pixels per column. In particular, the results show that if the target threshold is set to $V_{cal} = 400$ injecting 8 pixels per column, the actual threshold that a single pixel has is close to 377. In terms of electrons, this difference is of about 150 electrons (the conversion factor from V_{cal} to electrons is discussed in the next paragraph). Given the little difference, the dependence of the threshold on the number of injected pixels can be overlooked.

3.2.2 V_{cal} to electrons conversion

An important parameter that must be determined is the conversion factor from V_{cal} to the charge expressed in number of charge units (electrons). The nominal conversion factor is close to 5.5 $\frac{e^-}{V_{cal}}$.

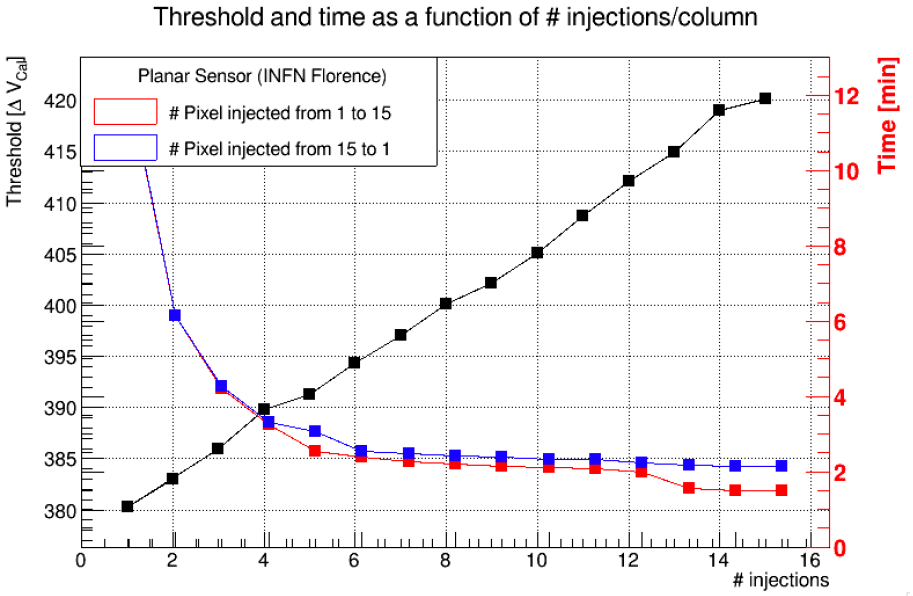
To obtain the value of the injected charge in units of electrons, the following formula can be used:

$$Q[e^-] = \frac{V_{inj}[V] \times C_{inj}[fF]}{1.6 \times 10^{-19} \frac{C}{e^-}} \tag{3.1}$$

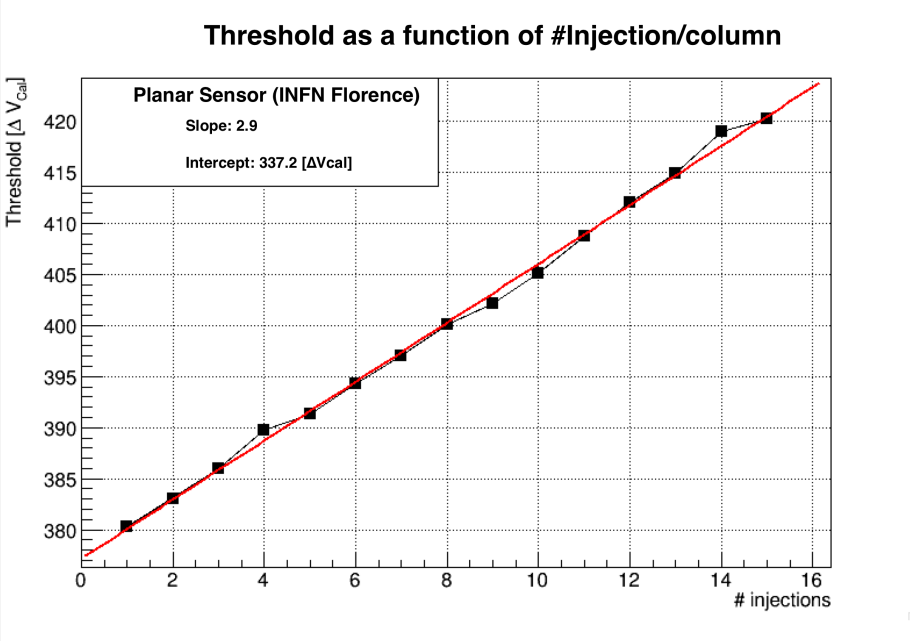
In Eq. 3.1, C_{inj} is the injection capacitance described in paragraph 2.4.1 and its nominal value is 8.02 fF, while $V_{inj}[V]$ is the difference between $V_{cal-High}$ and $V_{cal-Med}$ in volts. So, to obtain the conversion factor from V_{cal} to electrons, a voltage must be associated to both registers *VCAL_MED* and *VCAL_HIGH*. To do so, the Monitor Block of the CROC chips can be used. This block enables digitalization and readout of internal parameters, such as the temperature or voltages and currents from different parts of the chip. For example, the Monitor Block can be configured to have the value of *VCAL_HIGH* in volts on dedicated jumpers placed on the tested module. In this way, a value of the register *VCAL_HIGH* can be associated to a voltage, reading the jumpers output using a multimeter. The same association can be repeated with the register *VCAL_MED*.

So, using the Monitor Block, it's possible to obtain the value of ΔV_{cal} in volts. Repeating the measurement, changing the value of *VCAL_HIGH*, a linear behavior is observed (Fig. 3.5). The relation between the register ΔV_{cal} and its value in volts can be described using the following formula:

$$V_{inj}[V] = \alpha + \beta \times \Delta V_{cal} \tag{3.2}$$



(a)



(b)

Figure 3.4: (a) Threshold and Scan duration as a function of the number of injected pixels per column; (b) Linear fit of the threshold as a function of the number of pixels injected per column.

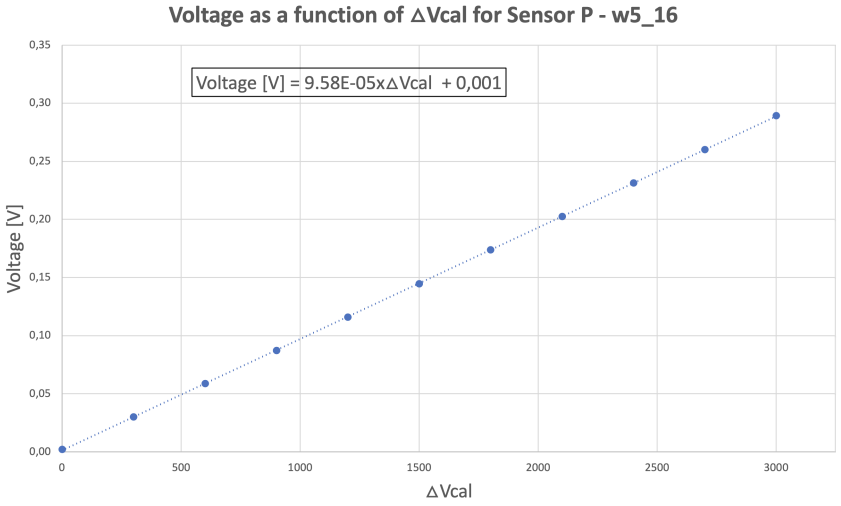


Figure 3.5: Voltage measured from the Monitor Block jumpers as a function of V_{cal}.

Since in Eq. 3.2 $\alpha \sim 0$, Eq. 3.1 becomes:

$$Q[e^-] = \frac{(\beta \times \Delta V_{cal}) \times C_{inj}[fF]}{1.6 \times 10^{-19} \frac{C}{e^-}} = \left[\frac{\beta \times C_{inj}[fF]}{1.6 \times 10^{-19} \frac{C}{e^-}}\right] \times \Delta V_{cal} \tag{3.3}$$

Using the value of the slope shown in Fig. 3.5, it is possible to compute the value of the conversion factor from V_{cal} to electrons for a specific module (module *w5-16* in this case). However, the same measurement has been repeated for all the modules used. Results are given in the conversion table 3.2. As it can be noted, the conversion factors are close to $5 \frac{e^-}{V_{cal}}$ for all the sensors, which means 10% smaller than the nominal value. In all the analysis done in this work, the proper conversion factor has been used for each module.

		Conversion Table			
		V_{cal} / **GDAC**	**e** / V_{cal}	V_{cal} / (TDAC V_{LDAC})	**e /ns**
3D Sensors	**w6_5-1**	8.4 ± 1.1	5.131 ± 0.005	0.13 ± 0.01	0.24Krum + 7.8
	w13_5-1	8.3 ± 1.3	5.120 ± 0.002	0.12 ± 0.01	0.24Krum + 7.3
Planar Sensors	**w5_1-6**	8.8 ± 0.1	4.803 ± 0.003	0.11 ± 0.01	0.24Krum + 7.7
	P-719	12.2 ± 3.8	5.057 ± 0.008	0.13 ± 0.02	0.27Krum + 37.5
	P-704	10.1 ± 2.2	4.722 ± 0.008	0.12 ± 0.02	0.27Krum + 8.6
Bare Chips	**Bare - 20145**	7.5 ± 0.6	4.776 ± 0.003	0.12 ± 0.02	0.20Krum + 6.5
	Bare - 20165	7.7 ± 1.0	5.000 ± 0.002	0.10 ± 0.01	0.22Krum + 7.0

Table 3.2: General Conversion table for all the modules used.

3.2.3 GDAC to V_{cal} conversion

The register named GDAC (or *GDAC_LIN*) is used to set the Global Threshold of the Linear AFE (as described in paragraph 2.1.1). However, the threshold is often given in V_{cal} units. So, it is important to obtain for each module the proper conversion factor from GDAC to V_{cal}. To do so, the ThresholdScan can be used. Indeed, the latter scan allows you to associate to a value of GDAC the corresponding value

of the actual threshold in V_{cal} unit using the S-Curves. So, the conversion factor from GDAC to V_{cal} can be obtained repeating the scan, for different GDAC values (Fig. 3.6).

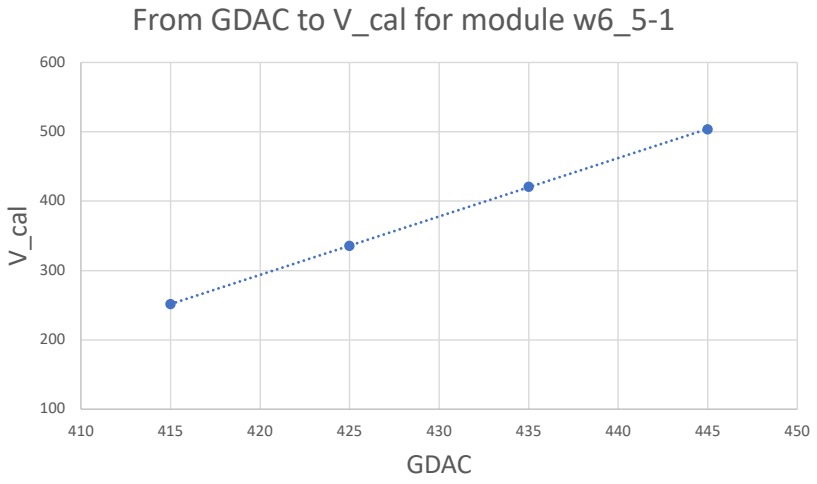


Figure 3.6: V_{cal} units as a function of GDAC register. This relation has been obtained using the ThresholdScan in Ph2ACF software.

The results shown in Fig. 3.6 are obtained using the 3D module named *w6_5-1*, however the same procedure described previously can be repeated for all the modules used. All the conversion factors are given in table 3.2.

3.2.4 $\mathrm{TDAC} \cdot \mathrm{V_{LDAC}}$ to $\mathrm{V_{cal}}$ conversion

As described in paragraph 2.1.1, the register LDAC sets the value of V_{LDAC} which determines the output dynamic range of the in-pixel threshold trimming DAC (TDAC), used to reduce the threshold dispersion among pixels.

In the previous paragraph, the conversion factor from GDAC to V_{cal} has been obtained. However, since the threshold in each pixel is determined also by V_{LDAC} and TDAC (see Eq. 2.1), the conversion factor from $\mathrm{TDAC} \cdot \mathrm{V_{LDAC}}$ to V_{cal} must be determined too. The first thing to do in order to *qualitatively* obtain that conversion factor is to study the behavior of the threshold (in V_{cal} units) as a function of TDAC for different values of V_{LDAC}. As said in paragraph 2.1.1, the value of TDAC can change pixel-to-pixel in order to reduce threshold dispersion. However, in the measurement described previously, it is set to be the same for all the pixels and ranges from 0 to 31. Fig. 3.7 shows $-(V_{cal} - V_{cal_0})$ as a function of TDAC. In the latter figure, V_{cal_0} is the value of the threshold in V_{cal} units when $\mathrm{TDAC} = 0$.

The relation between $-(V_{cal} - V_{cal_0})$ and TDAC can be expressed by the following formula:

$$-(V_{cal} - V_{cal_0}) = k \cdot \mathrm{TDAC} + h \tag{3.4}$$

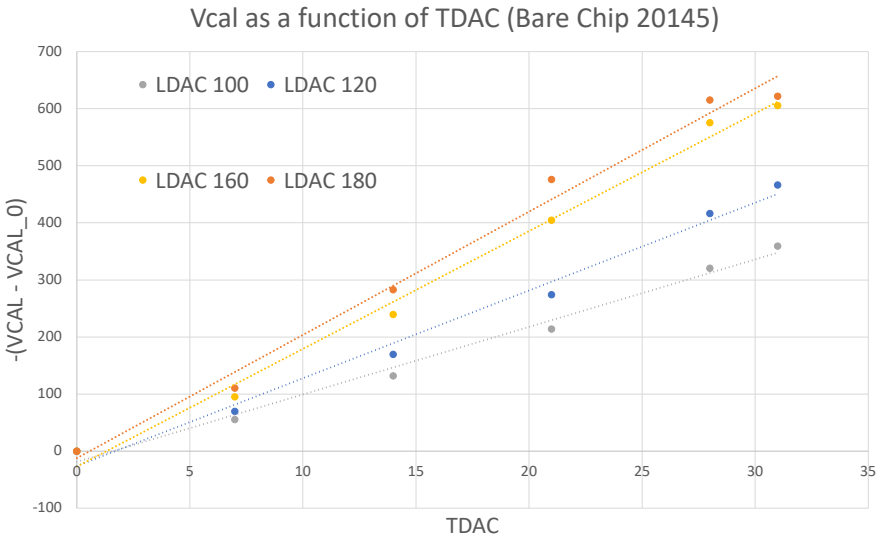


Figure 3.7: Threshold in V_{cal} units as a function of TDAC for different values of LDAC (i.e. different values of V_{LDAC}).

As shown in Fig. 3.8, the slope k is linear with V_{LDAC}:

$$k = B \cdot V_{LDAC} + A \tag{3.5}$$

In order to obtain qualitatively the value of the conversion factor, the intercepts can be neglected. So, the conversion factor from $TDAC \cdot V_{LDAC}$ to V_{cal} is represented by the slope B:

$$-(V_{cal} - V_{cal_0}) = B \cdot (V_{LDAC} \cdot TDAC) \tag{3.6}$$

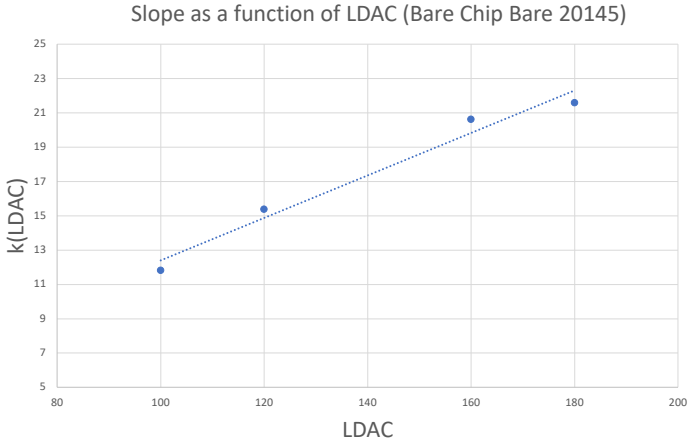


Figure 3.8: Slope k as a function of LDAC (i.e. V_{LDAC}). The slope of this linear behavior is the conversion factor from $TDAC \cdot V_{LDAC}$ to V_{cal}.

The results here described are obtained using a bare chip (in particular chip 20145). The conversion factors for all the other modules are given in table 3.2.

3.2.5 *KRUM_CURR_LIN* to I_{Krum} conversion

As described in paragraph 2.1.1, the register *KRUM_CURR_LIN* is used to set the value of I_{Krum}, which is the current in the Krummenacher feedback of the preamplifier. In order to fully understand the chip general properties, the conversion factor from the value of the register to the value of I_{Krum} in $\frac{e^-}{ns}$ needs to be found.

One of the possible ways to find this conversion factor is using the procedure called *SignalScan* in Ph2ACF (described in paragraph 3.1.2). The latter procedure, indeed, allows you to obtain the shape of the signal induced by a fixed injected charge Q. Since one of the purposes of the Krummenacher current is to linear discharge the capacitance of the preamplifier, a linear fit with the decreasing part of the signal can be performed to obtain the value of I_{Krum} in units of $\frac{GDAC}{ns}$. Fig. 3.9 shows the shape of the signal induced by a charge $Q = 9000\,e^-$, with a fixed value of *KRUM_CURR_LIN*.

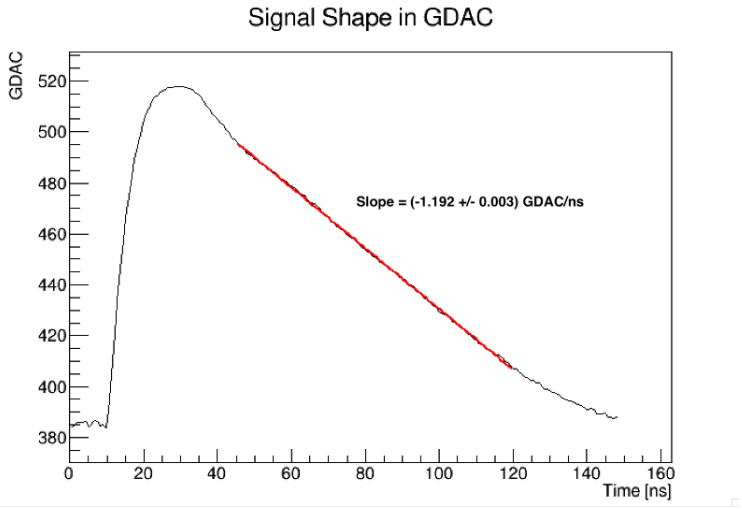


Figure 3.9: Signal Shape obtained by *SignalScan*. A linear fit with the decreasing part of the signal is performed to obtain the value of I_{Krum} in units of $\frac{GDAC}{ns}$.

The linear fit described previously can be repeated, changing every time the value of the register *KRUM_CURR_LIN*. Using the conversion factor from GDAC to V_{cal} and the conversion factor from V_{cal} to electrons, each value of *KRUM_CURR_LIN* can be associated to a value of the Krummenacher current in $\frac{e^-}{ns}$. The final conversion factor can be obtained by studying the linear relation between the register and I_{Krum}. Fig. 3.10 shows the latter relation obtained for a bare chip (in particular for the bare chip 20145). From that linear relation, it can be concluded that the conversion from *KRUM_CURR_LIN* to I_{Krum} can be described by the following formula:

$$I_{Krum} = (0.2 \times KRUM\_CURR\_LIN + 6.5) \left[\frac{e^-}{ns}\right] \tag{3.7}$$

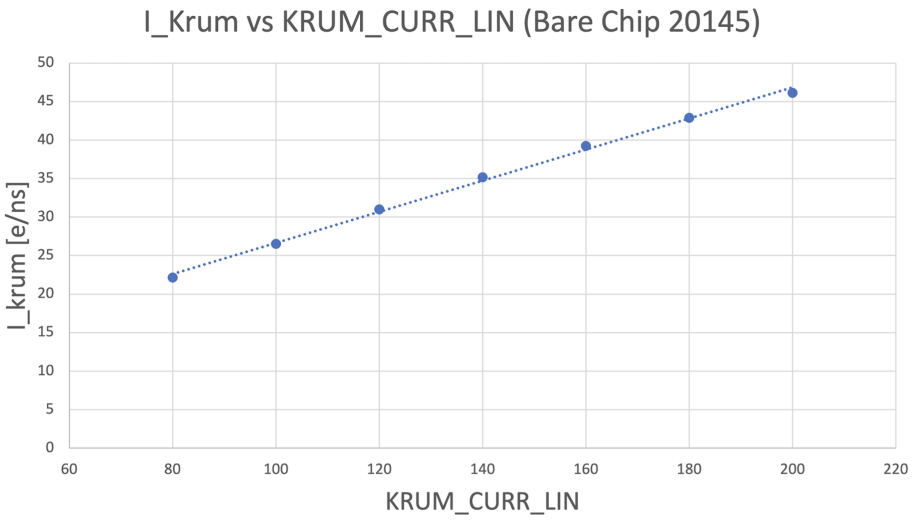


Figure 3.10: Linear relation between the register *KRUM_CURR_LIN* and the value of the Krummenacher current in units of $\frac{e^-}{ns}$.

This procedure has been performed on all modules, the results are given in table 3.2.

CHAPTER 4

THRESHOLD OSCILLATIONS

In the CMS Tracker, as in all other tracking device, the threshold plays a really important role, because it defines whether a hit is recorded in a given pixel. Collecting all the hits recorded in a given bunch crossing makes possible to reconstruct the trajectories of the particles produced in a proton - proton collision.

As it has been said in paragraph 2.4.2, in order to have a uniform sensitivity, it's very important that the threshold dispersion is kept small within the pixel matrix. However, not only *space* sensitivity must be uniform, but also *time* sensitivity: the value of the threshold must be constant over time, otherwise particles hitting the detector in different times will see a different threshold.

It has been found out that the threshold is characterized by a 40 MHz oscillation, which phase and amplitude can slightly change channel by channel. The studies of this spurious effect are described in this chapter. It is also important to understand if this oscillation could represent a potential problem in the future tracking system.

4.1 Synchronous and Asynchronous modes

In order to characterize threshold oscillation, it's very important to describe how the ROC logic identifies a hit, i.e. when the pixel connected to a given ROC channel sees a signal. Two different sampling modes can be used: *Synchronous mode* and *Asynchronous mode*.

In both sampling modes, a hit is recorded if the comparator output is up at the 40 MHz clock edge. This hit is assigned to the bunch crossing corresponding to that clock cycle. The main difference between Synchronous and Asynchronous mode is that in the former the comparator output is up as long as the signal exceeds the threshold, while in Asynchronous mode it goes up when the signal exceeds the threshold, and it remains up for 25 ns. Fig. 4.1 explains with a graphical sketch the difference between Synchronous and Asynchronous mode. The sketch 4.1*a* shows a

signal which is recorded as a hit in Asynchronous mode. However, the same signal will not be seen as a hit in Synchronous mode (sketch 4.1*b*), unless it has a bigger charge (sketch 4.1*c*) or arrives with a larger delay (sketch 4.1*d*).

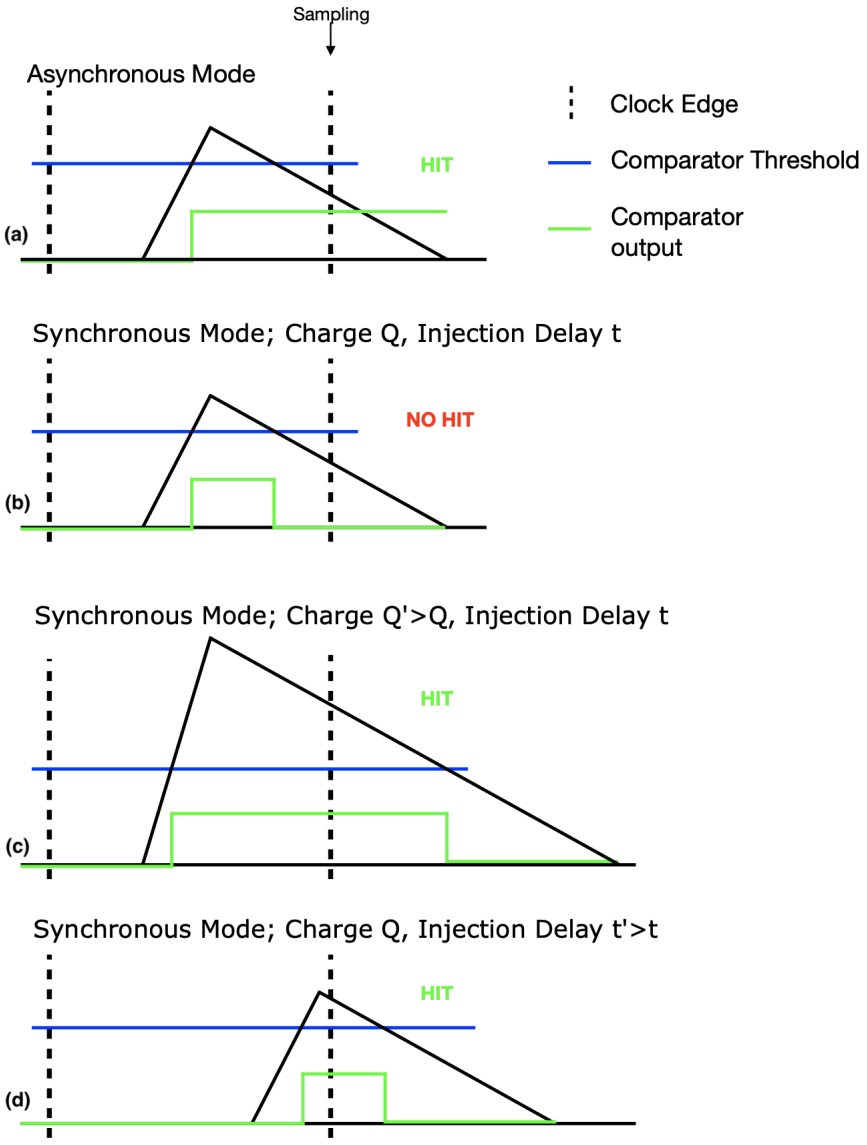


Figure 4.1: Sketches which show the main difference between Synchronous mode and Asynchronous mode.

Hence, a hit is recorded depending on the sampling mode and the interplay between charge and signal delay. In this respect, it is clear that the actual threshold (i.e. the smallest detectable signal for a given delay) can differ from the physical comparator threshold (V_{Thr}).

4.2 Tornado Plots

The tornado plot (or banana plot) is an important tool which can be used to characterize the threshold behavior. In order to describe this kind of plot, it's important to define the Hit Probability Detection (P) that, in general, depends on Q, Δt and V_{Thr}:

$$P = P(Q,\ \Delta t,\ V_{Thr}) \tag{4.1}$$

where $Q[\mathrm{e}^-]$ is the signal charge, $V_{Thr}[mV]$ is the comparator threshold and $\Delta t[ns]$ is the time between the arrival of the signal and the sampling. Within the scope of this work, we can assume that the signal is provided by the injection circuitry. In particular, Δt can be defined as:

$$\Delta t \ = \ (32 \times BX_{ID} \ - \ \text{Fine Delay}) \ \times \left[\frac{25\,\text{ns}}{32}\right] \tag{4.2}$$

In Eq. 4.2, Fine Delay is the value of the firsts six bits of CalibrationConfig register (see paragraph 2.1.1) in units of $\frac{25\,\text{ns}}{32}$. The sketch shown in Fig. 4.2 is useful to better understand the relation between Δt and Fine Delay. In Eq. 4.2, BX_{ID} is a number ranging from 1 to N that represent the ID of each of the N bunch crossings read by the ROC.

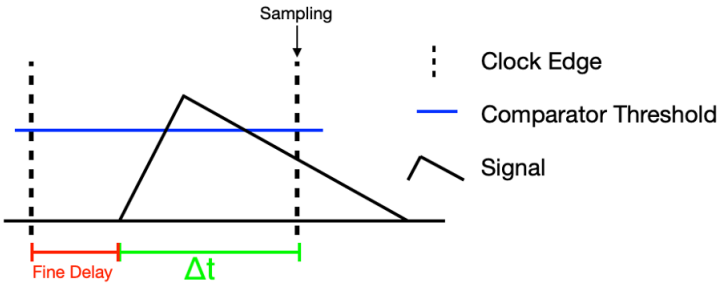
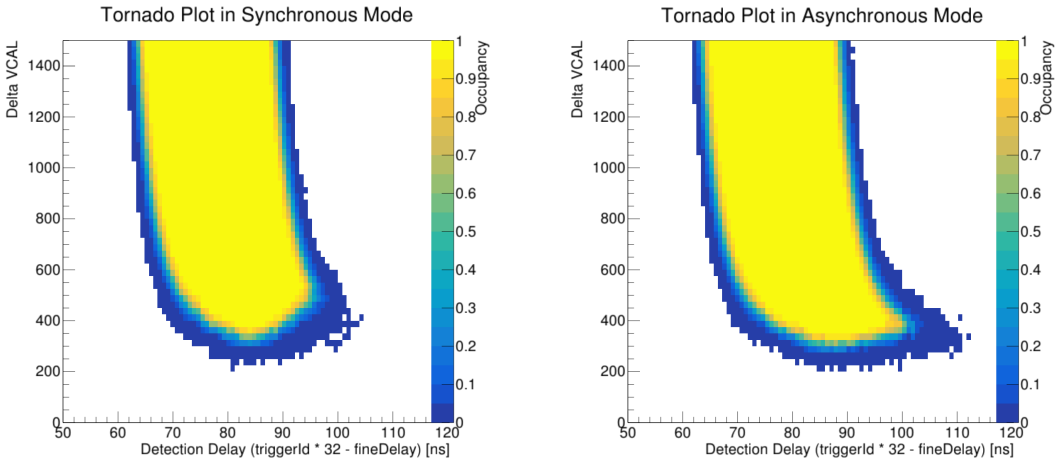


Figure 4.2: Sketch which shows the relation between Δt and Fine Delay.

Once defined P, a tornado plot is just a 2D scan on Q and Δt, with a fixed value of the comparator threshold. In Ph2ACF a tornado plot can be obtained by running a particular scan, called TimeWalk scan. The name of this procedure recall the fact that, by definition, the left side edge of the tornado plot represents the Time Walk curve[1]. The TimeWalk scan, in particular, produced both the tornado plot for a single channel and the tornado plot averaged across all channels.

It's important to note that the lower part of a tornado plot changes whether Synchronous or Asynchronous mode is chosen. To understand why, let's consider a high injected charge (e.g., 1400 $\mathrm{V_{cal}}$) and a given BX_{ID}. In this case, since the signal is really high, both in Synchronous and Asynchronous mode, it will be seen as a hit, whatever its Fine Delay is (ranging from 0 up to 25 ns), as it can be seen by looking

[1]Pulses of same shape and different amplitude crosses the threshold at different times. In particular, the higher is the charge, the bigger is the signal amplitude and so the smaller is the time t_w that the signal takes to exceed the threshold. The *Time Walk* is the plot that shows the behavior of t_w as a function of the injected charge.



(a) (b)

Figure 4.3: (a) Tornado plot in Synchronous mode; (b) Tornado plot in Asynchronous mode.

at the higher part of the tornado plots in Fig. 4.3. However, if the injected charge is small, a higher Δt is needed to see the signal as a hit for a given BX_{ID}. That is why the left part of the tornado plot bends to the right (both in Synchronous and in Asynchronous mode). Obviously, the tornado plot has a minimum: if a signal is lower than the threshold, it cannot be seen in any mode, regardless of the Fine Delay. Suppose that the minimum is $(\Delta t_{min}, Q_{min})$. In Synchronous mode, a signal injected with a Fine Delay D so to have $\Delta t = (32 \times BX_{ID} - D) \times \left[\frac{25\,\text{ns}}{32}\right] > \Delta t_{min}$ will be recorded as a hit only if $Q > Q_{min}$. In Asynchronous mode, instead, a signal with $Q = Q_{min}$ will be seen as a hit whatever its Fine Delay is (ranging from Δt_{min} to, ideally, $\Delta t_{min} + 25\,\text{ns}$).

4.2.1 Tornado Plots and Krummenacher current

In the previous paragraph, the difference between tornado plots in Asynchronous and Synchronous mode has been described; however, in order to completely understand the shape of this kind of plots, another important dependence must be considered. In particular, the bottom part of a tornado plot depends on the Krummenacher current. This aspect can be understood recalling the fact that a reduced I_{krum} leads to an increased ToT of the signal. That is why a signal which could be seen only in Asynchronous mode can be seen also in Synchronous mode just by lowering I_{krum}. For example, the signal sketched in Fig. 4.1*a* can be recorded as a hit also in Synchronous mode if its ToT is bigger.

According to what has been said previously, a tornado plot in Synchronous mode with low Krummenacher current should look like a tornado plot in Asynchronous mode. In order to verify this feature, several tornado plots have been made, changing sampling mode and Krummenacher current. The results of this measurement are shown in Fig. 4.4 and are obtained using a 3D sensor. No differences are expected by changing the sensor's type, since this feature is linked to how the chip logic samples

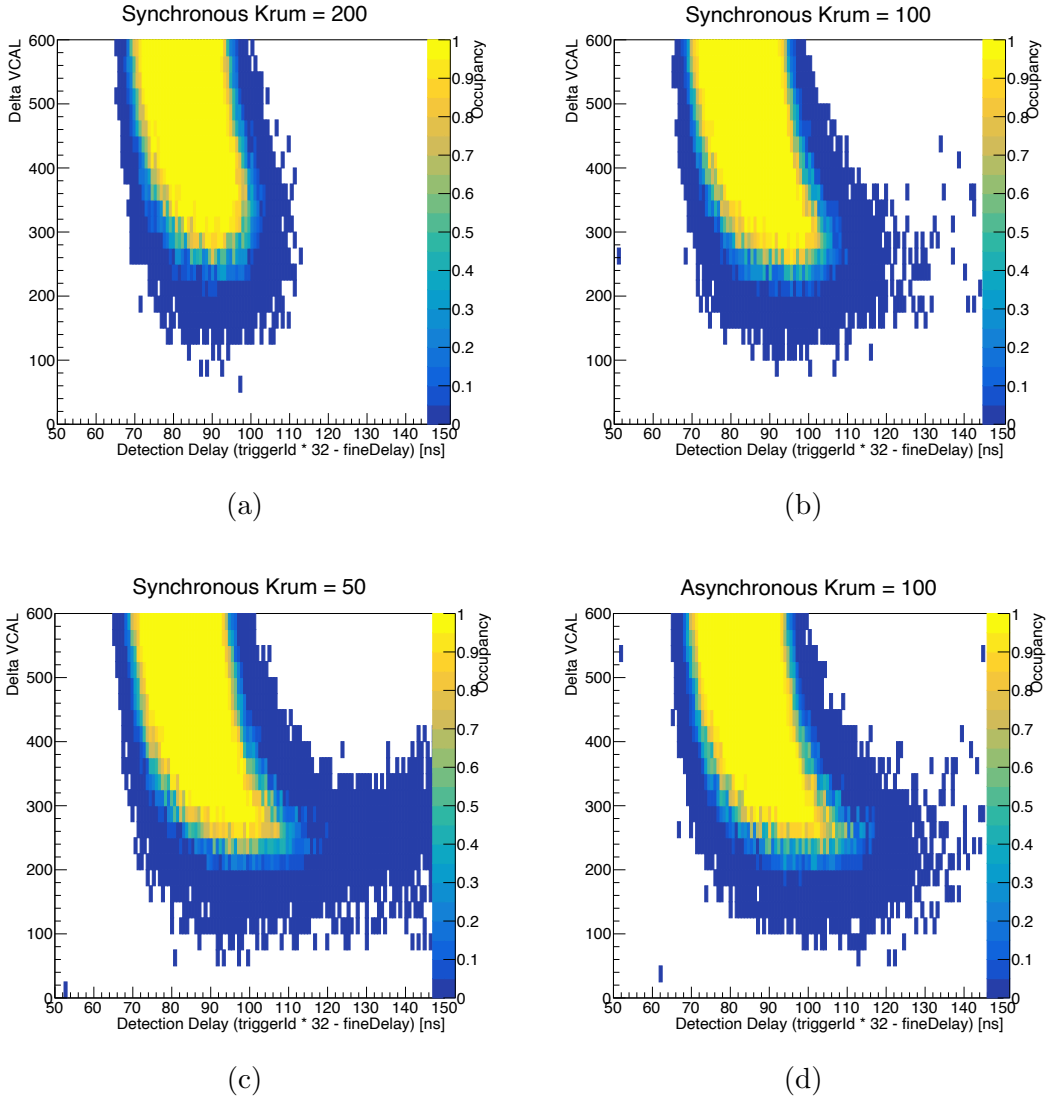


Figure 4.4: Tornado Plots changing sampling mode and value of Krummenacher current: (a) Synch mode with Krum = 200; (b) Synch mode with Krum = 100; (c) Synch mode with Krum = 50; (d) Asynch mode with Krum = 100.

and not to the physical properties of the sensor itself.

Another important feature of tornado plots which can be explained taking into account the Krummenacher current is the one concerning the so-called *bouncing hits*. These are noise hits that the chip logic see when a signal, returning to baseline, gets below the threshold. In fact, when slightly below threshold, the signal could exceed the threshold again because of a fluctuation due to noise (see Fig. 4.5 [29]). By definition, this effect should be maximum in Asynchronous mode and minimum in Synchronous mode. Indeed, bouncing hits could be seen in Synchronous mode only if the signal returns below the threshold close in time to a clock edge. However, also Krummenacher current can affect bouncing hits. Indeed, bouncing hits are more likely to be seen if the signal stays close to the threshold for a long period of time, that is, if it has a slow return to baseline.

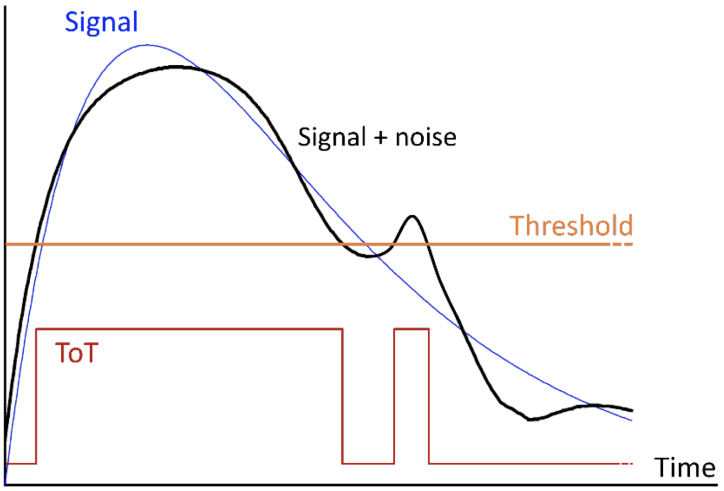


Figure 4.5: Sketch that shows the bouncing hits effect. When the signal is slightly below the threshold, it can exceed the threshold again because of a fluctuation due to noise.

To verify what previously said, four tornado plots have been produced, changing sampling mode and discharge speed. In particular, a Fast Discharge and a Normal Discharge have been taken into account. As it can be noted by looking Fig. 4.6 the bouncing hits effect is maximum in Asynchronous mode with Normal Discharge and absent in Synchronous mode with Fast Discharge. Another feature of this effect is that the distance between tornado plot and bouncing hits increases with the injected charge. This is due to the fact that the higher is Q, the longer is the ToT, and so the longer is the time before another hit can be seen.

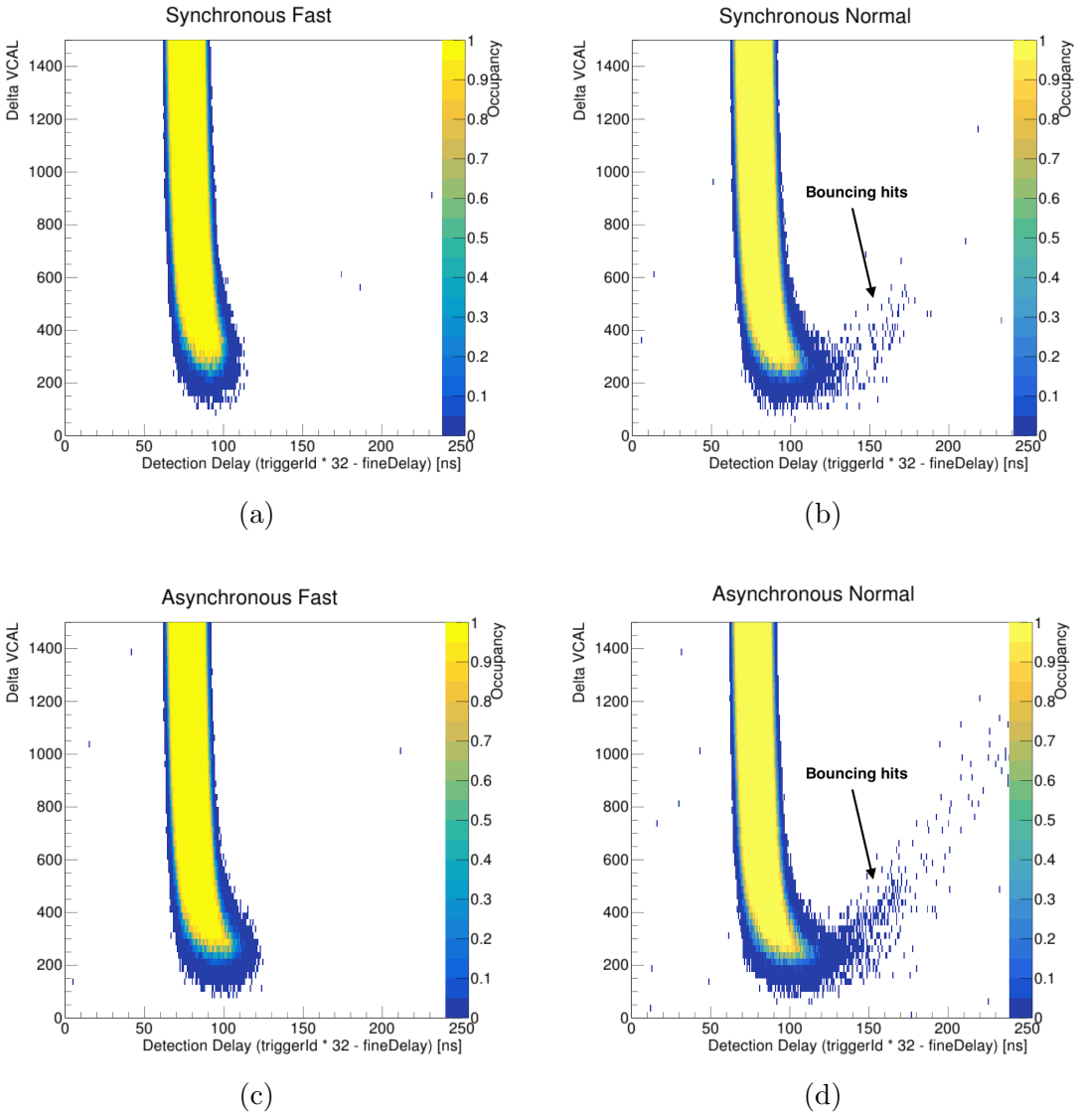


Figure 4.6: Bouncing effect changing sampling modes: (a) Synchronous Fast mode; (b) Synchronous Normal mode; (c) Asynchronous Fast mode; (d) Asynchronous Normal mode.

4.3 Fine Delay and Threshold Tuning

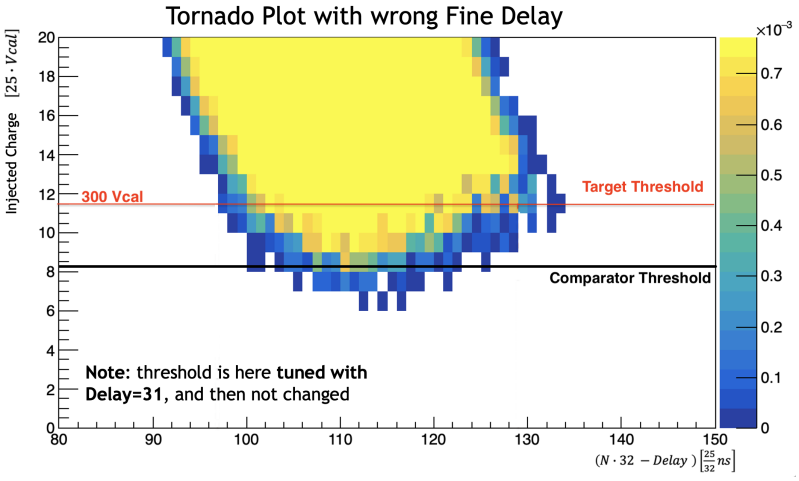
As said previously, the value of the threshold can be chosen by changing the value of GDAC registers in the configuration file. However, another important register must be taken into account during threshold tuning: the CalibrationConfig register. This register, indeed, allows you to choose the Fine Delay of the injections.

In order to understand the importance of this delay, suppose to consider a single channel, to fix the value of CalibrationConfig (e.g, CalibrationConfig = 31) and to tune the target threshold to a value of GDAC corresponding to 1500 e^- (using tuning procedure described in paragraph 3.1.2). In this configuration, the Tornado Plot is the one shown in Fig. 4.7a. By looking at the shape of the Tornado Plot, it can be noted that the sampling mode used is the Synchronous mode. However, the following considerations still stand also in Asynchronous mode.

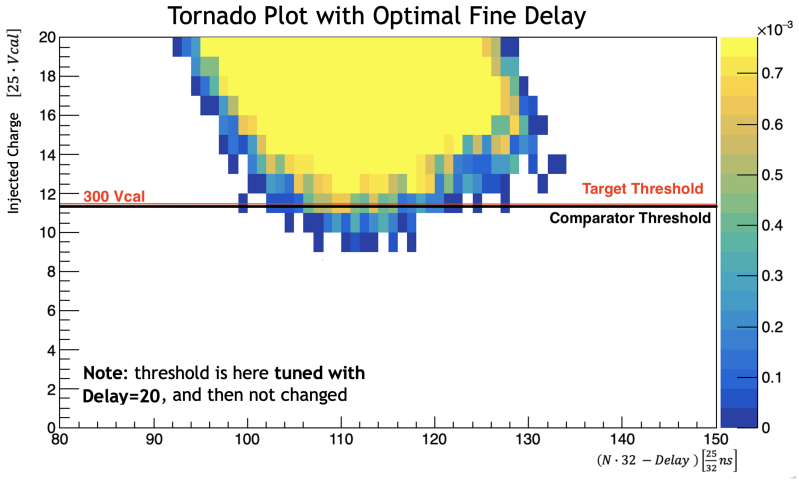
In Fig. 4.7a, the red line represents the target threshold set during the tuning procedure (in this case $V_{cal} = 300 \approx 1500e^-$), while the black line is the physical comparator threshold. Because of the choice of the Fine Delay, in Fig. 4.7a the red line does not match the black one. This means that if a signal has a delay corresponding, for example, to $\Delta t = 110 \left[\frac{25\,ns}{32}\right]$ it will be recorded as a hit even if it is lower than the resulting tuned threshold. In general, the smallest detectable signal (SdS) depends on injection delay and the comparator threshold V_{Thr} can always be considered as the SdS which corresponds to the point of minimum of the tornado plot. According to what previously said, choosing a value of the Fine Delay that is not its Optimal value will lead to a situation in which the threshold loses its meaning. In order to avoid this condition, it is important to choose a value of the Fine Delay which makes the comparator threshold to be equal to the target tuned threshold. This value can be called Optimal Fine Delay. If the smallest detectable signal corresponding to the minimum of the tornado plot is equal to the target tuned threshold, we are sure that each arriving particle sees a threshold which is *at least* equal to the value set in the configuration file. The Tornado Plot obtained setting the Optimal Fine Delay is shown in Fig. 4.7b.

It could be pointed out that if the Optimal Fine Delay is set, the tuned threshold is more like a lower limit: the threshold cannot be lower, but, according to what has been said, it can be higher. Fig. 4.8 shows the distributions of the time of flight (Tof) of particles produced in a simulated $t\bar{t}$ event. Each distribution is referred to one of the four layers of the TBPX, and the Tof is corrected for the nominal arrival time from the interaction point, assuming a straight trajectory and a speed close to the speed of light; this is called *TimeToDet*. For particles with high transverse momentum[2] (P_t), Tof $\simeq$ *TimeToDet*, so the peak of the distribution is almost centered in zero. On the other hand, the tail of the distribution is formed by low P_t particles for which Tof $\gg$ *TimeToDet*. The calibration procedure in the running experiment consists on the alignment of the minimum of the tornado plot with the maximum of the distribution. By doing so, low P_t particles could see a higher threshold, but all the particles of interest (which are high P_t particles) will see the same value of the

[2]The transverse momentum is the projection of the particle's momentum on the $x - y$ plane.



(a)



(b)

Figure 4.7: (a) Tornado plot setting the Wrong Fine Delay; (b) Tornado plot setting the Optimal Fine Delay.

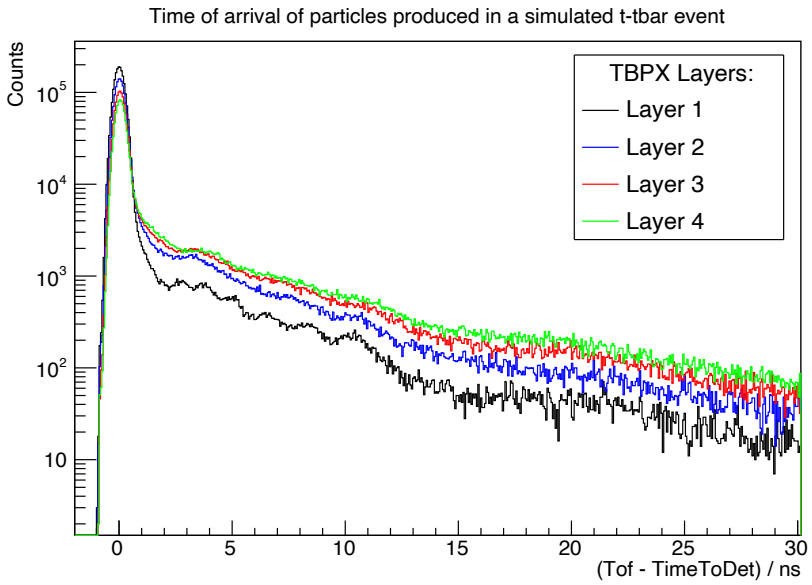


Figure 4.8: Distribution of the time of arrival of particles produced in a simulated $t\bar{t}$ event. In this distribution, the time is corrected by the time of arrival in the detector.

threshold, since the distribution is narrow. This alignment needs to be done layer by layer because each layer of the TBPX has a slightly different *TimeToDet*.

4.3.1 Optimal Fine Delay and Noise performance

The Fine Delay calibration is also important with respect to the noise. Noise fluctuations happen all the time, not necessary synchronous with the clock, so the smaller is V_{Thr}, the higher the noise sensitivity will be. The choice of the Optimal Fine Delay will improve noise performance because it will make the comparator threshold as high as possible. In case of Optimal Fine Delay, the higher value for V_{Thr} is the target threshold set with Ph2ACF.

In order to verify this, the number of problematic pixels[3] as a function of the SdS can be studied in two different cases: threshold tuned setting the Optimal Fine Delay and threshold tuned using a Wrong Fine Delay. In both cases, the threshold is tuned to 1000 e^- and the value of the SdS has been changed by changing the value of the GDAC register in the configuration file.

The measurement described previously has been repeated for different modules; Fig. 4.9 shows the results obtained using a 3D sensor (in particular, module FBK-w6_5-1). As it can be noted by looking at that figure, if the threshold is set to the value of 1000 e^-, the number of problematic pixels is more or less one order of magnitude bigger, if the Wrong Fine Delay is set.

[3]The number of problematic pixels is defined as the sum of the number of stuck and noisy pixels, which are obtained, respectively, running a *StuckPixelScan* and a *NoiseScan*.

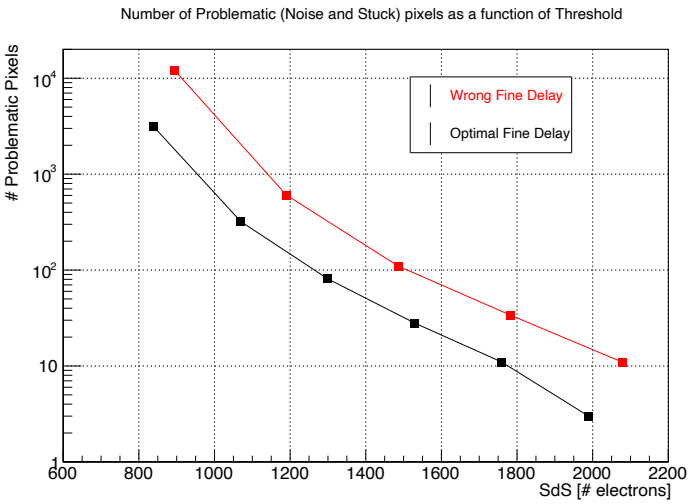


Figure 4.9: Problematic pixels as a function of the smallest detectable signal. This plot is obtained using the module FBK-w6_5-1.

To conclude, the choice of the Optimal Fine Delay is crucial for two main reasons: it makes all the particles of interest see a threshold which corresponds to the target tuned threshold, and it also provides a better noise performance.

4.3.2 How to find the Optimal Fine Delay

In the previous paragraphs, the importance of the Fine Delay has been underlined. In order to find its optimal value, suppose to scan the value of Δt and to project the tornado plot along the y-axis for each value of the time. By doing so, several S-Curves can be obtained. For each of them, a different $(\Delta V_{cal})_{50\%}$ can be defined. The latter is the value of ΔV_{cal} that corresponds to 50% occupancy. Initially, increasing Δt results in a lower $(\Delta V_{cal})_{50\%}$; however, at a certain point, the value of $(\Delta V_{cal})_{50\%}$ starts to increase by keeping increasing the time. This behavior is shown in Fig. 4.10. Once performed the scan, the minimum value of $(\Delta V_{cal})_{50\%}$ can be associated to a given value of the time, named Δt_{min}. For example, $\Delta t_{min} = 108.8 \left[\frac{25\,\text{ns}}{32}\right]$ in the example given in Fig. 4.10. Please note that from Δt_{min} is possible to obtain the value of the Optimal Fine Delay by knowing the bunch crossing ID.

However, the procedure previously described is quite time-consuming, since it needs to produce an entire tornado plot and to run a complex script in order to find the value of the Optimal Fine Delay. An alternative procedure can be described. In order to find the value of the Optimal Fine Delay, we can scan the value of Δt while injecting a relatively high charge (i.e. $6000\,e^-$). Running this measurement corresponds to cutting the higher part of the tornado plot and project it along the x-axis (see Fig. 4.11). Please note that in order to obtain the right plot in Fig. 4.11 an entire tornado plot is not needed. Indeed, in the former procedure, the value of the injected charge is fixed.

By analyzing the projection shown in Fig. 4.11, it is possible to define a special value of Δt, called Δt_{Edge}, which is the value along the x-axis which corresponds to

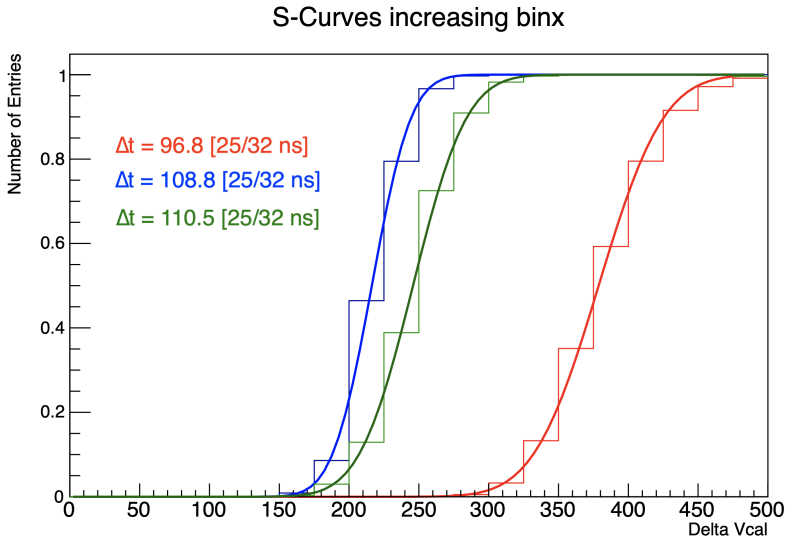


Figure 4.10: In this figure is shown how the value of $(\Delta V_{cal})_{50\%}$ decreases, reaches a minimum and increases again by increasing the Δt value.

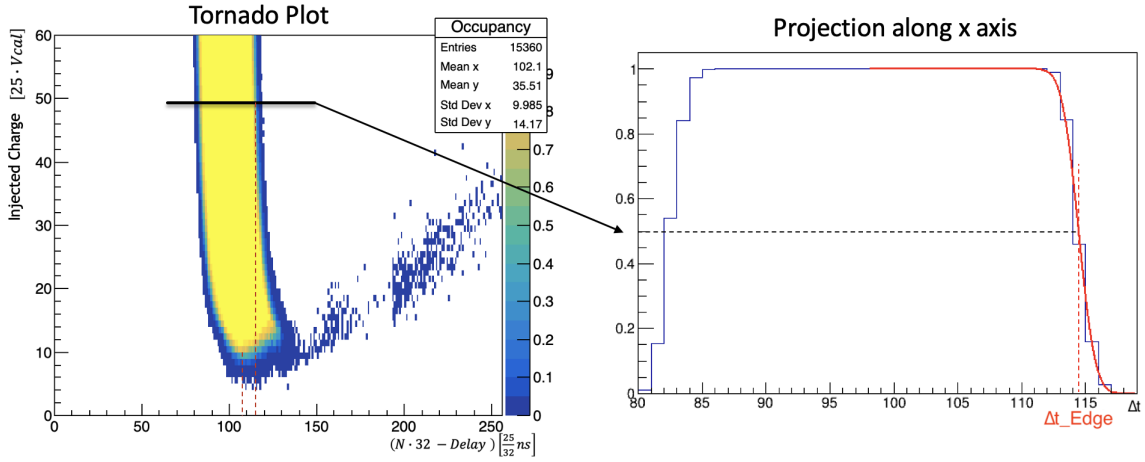


Figure 4.11: Description of the procedure which can be run to find the Optimal Fine Delay. The value obtained from this procedure must be increased by 5 units to get as close as we can to the minimum of the tornado plot.

50% occupancy, considering the right edge of the tornado plot (red part underlined in Fig. 4.11). In the given example, $\Delta t_{Edge} = 114.3 \left[\frac{25\,\text{ns}}{32}\right]$. Please note that the previous value of Δt_{Edge} differs from Δt_{min} by only $\sim 5 \left[\frac{25\,\text{ns}}{32}\right]$]. This relation between Δt_{Edge} and Δt_{min} has been verified repeating both procedures taking into account several different tornado plots. For the precision level required in this work, the value of the Optimal Fine Delay can be obtained starting from Δt_{Edge}, decreased by $5 \left[\frac{25\,\text{ns}}{32}\right]$.

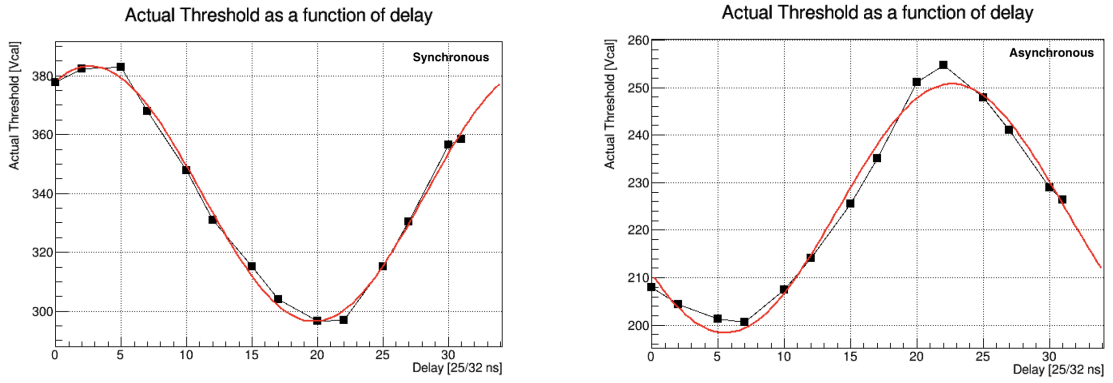
Please note that the second procedure described is much faster than the first one, since the injected charge is fixed and only a single fit needs to be performed. In Ph2ACF there is a proper scan which perform the procedure previously described, called *InjectionDelay* scan.

4.4 Threshold Oscillation in Synchronous and Asynchronous mode

Since the value of the smallest detectable signal depends on the injection delay for a given target threshold, in Synchronous mode a periodic variation of the *Actual Threshold*[4] is expected changing the value of the Fine Delay. This oscillation can be observed scanning the Fine Delay value while running the *ThresholdScan*. The plot showing the actual threshold as a function of the Fine Delay is shown in Fig. 4.12a. The oscillation minimum corresponds to $300\,\mathrm{V}_{cal}$ because the target threshold was set exactly to $300\,\mathrm{V}_{cal}$ (corresponding to $1500\,\mathrm{e}^-$) after setting the Optimal Fine Delay (if the Wrong Fine Delay had been chosen, the oscillation minimum would have been lower than $300\,\mathrm{V}_{cal}$).

On the other hand, considering the Asynchronous mode, no oscillation is expected, since in this mode the smallest detectable signal does not depend on injection time. However, if the same procedure is repeated in Asynchronous mode, an unexpected periodic dependence of the actual threshold on Fine Delay can be observed (Fig. 4.12b). In this case, this oscillation can be explained only taking into account an unexpected oscillation of the comparator threshold V_{Thr}.

The results previously shown are obtained using a planar sensor, however, the oscillation of the actual threshold and of the comparator threshold can be observed in any kind of sensors. Oscillation characteristics like amplitude and phase can slightly change sensor by sensor, as it will be shown in the next chapters.



(a) (b)

Figure 4.12: (a) Expected Threshold oscillatio in Synchronous mode; (b) Unexpected Threshold oscillation in Asynchronous mode.

4.5 Noise Performance and Threshold Oscillation

If the threshold is constant, noise performance depends only on the threshold value (the lower is the threshold, the higher is the probability to see a noise hit). However,

[4]The *Actual Threshold* is the threshold value obtained using the *ThresholdScan*. It can also be considered as the Smallest detectable Signal for a given delay and a fixed target threshold.

with an oscillating threshold, noise performance could depend also on time: suppose to have a noise signal with delay D_1 which does not exceed the threshold; if the threshold is constant, the signal is never recorded as a noise hit, regardless its delay. However, since threshold can change, the noise signal described previously could be recorded as a noise hit if it has, for example, a delay $D_2 > D_1$, because at that time the threshold is lower. The time dependence described previously leads to an increased number of noisy pixels due to the threshold behavior.

Noise can be studied in both sampling modes. Taking into account the differences between Synchronous and Asynchronous mode, different noise figures are expected. In particular, the effect previously described is not expected to be seen in Synchronous mode. Indeed, in Synchronous mode the signal is sampled every 25 ns, and so, since the comparator threshold oscillation has a 40 MHz component, at every clock edge, always the same value of the threshold is seen. Therefore, in Synchronous mode, noise performance do not depend on the comparator threshold oscillation. On the other hand, noise figures are expected to depend on comparator threshold oscillation in Asynchronous mode.

To verify the hypothesis formerly expressed, the first thing to do is to find the Optimal Fine Delay (using the procedure described in paragraph 4.3.2) and tune the threshold to a given value (for example, 15 000 e^-). Subsequently, it is important to describe the threshold oscillations for each channel. To do so, an appropriated Ph2ACF scan can be used. The latter is called *ThresholdOscillation* scan. It reproduces the plot shown in Fig. 4.12 for each injected channel. Fig. 4.13 shows a typical plot produced by the scan, for a reference channel. Each oscillation is fitted

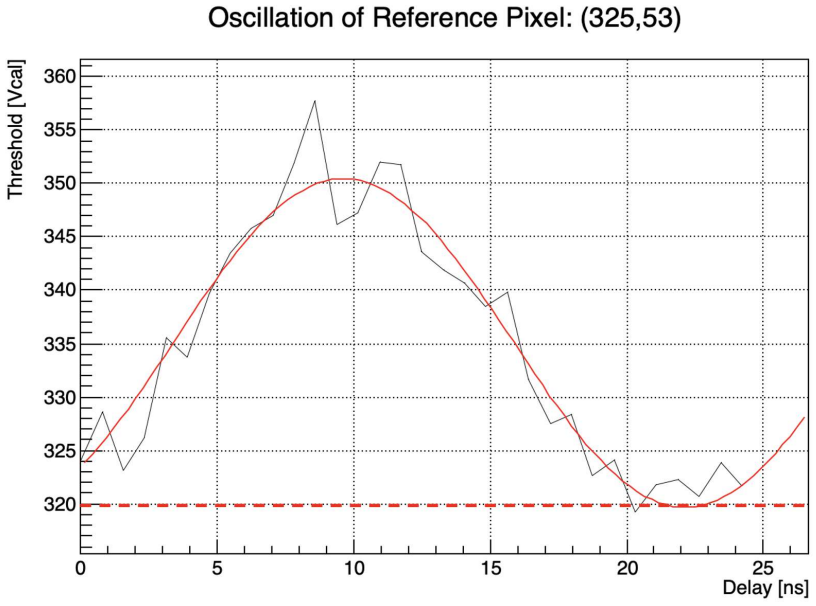
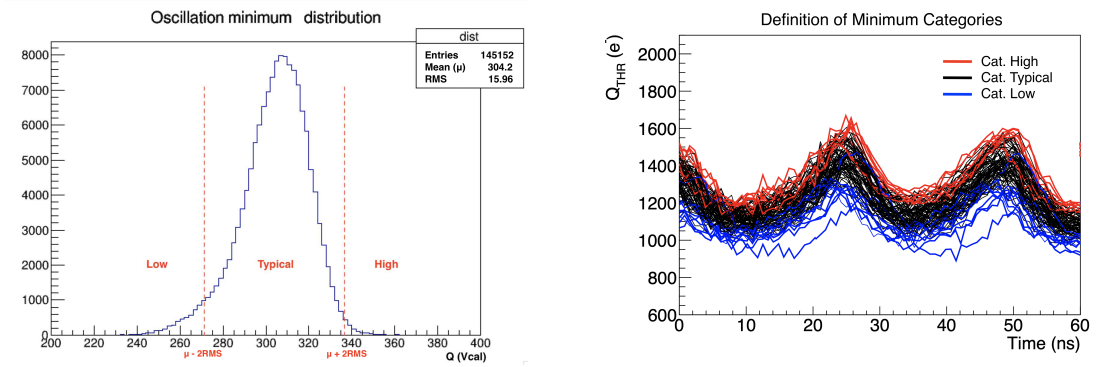


Figure 4.13: One of the plots obtained using the ThresholdOscillascion scan. This plot shows the threshold oscillation observed in a reference pixel. The dashed line corresponds to the oscillation minimum.

using a cosine function, from which the oscillation parameters can be obtained. The function used in the fit is:

$$Thr \;=\; Thr_0 \;+\; A \times cos\left(2\pi\,\frac{t-\phi}{25}\right) \tag{4.3}$$

In Eq. 4.3, A represents the oscillation peak amplitude, ϕ represents the oscillation phase, Thr_0 represents the oscillation baseline, while the frequency is fixed at 40 MHz. Since the function used is a cosine function, the oscillation phase can also be considered as the point of minimum of the oscillation. The distribution of oscillation minimums, obtained from fit parameters, is shown in Fig. 4.14a.



(a) (b)

Figure 4.14: (a) Distribution of oscillation minimums with the definition of the three categories (Low, Typical and High). (b) Oscillations of different pixels overlapped.

Defining the mean value of the distribution as μ and considering its RMS, by looking at the minimum distribution, three different pixels categories can be defined:

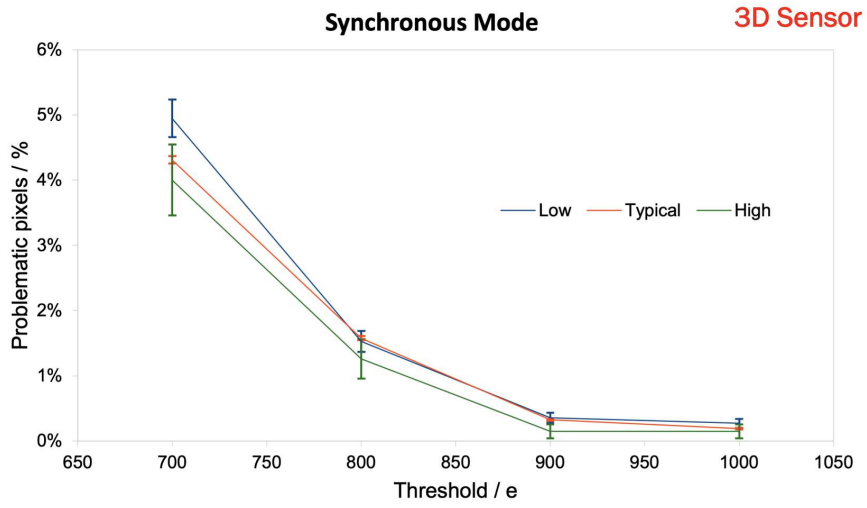
- **Low:** if $Q < \mu - 2RMS$
- **Typical:** if $\mu - 2RMS < Q < \mu + 2RMS$
- **High:** if $Q > \mu + 2RMS$

Low, High and Typical categories are shown both in Fig. 4.14a and in Fig. 4.14b. In the latter, in particular, the oscillation of different pixels are overlapped in order to better understand the differences between the three defined categories. The procedure previously described must be done in Asynchronous mode because in that sampling mode we are sensitive to V_{Thr} oscillation.

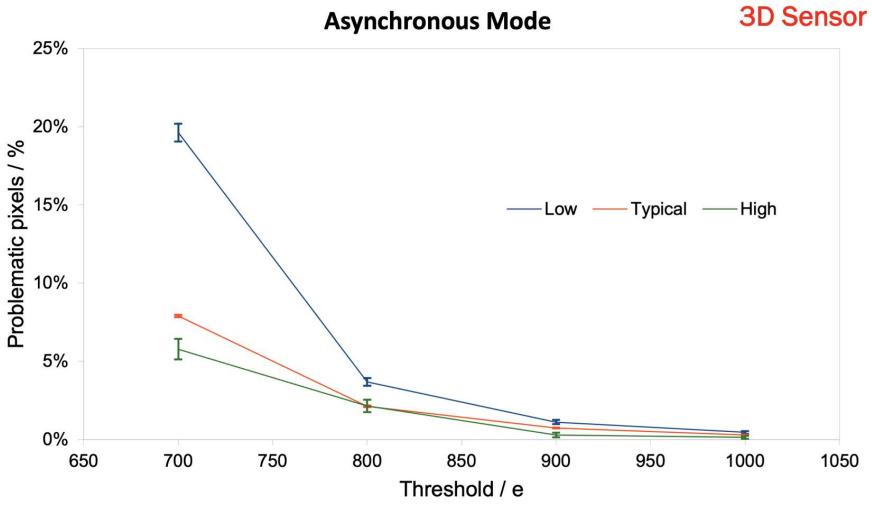
Once the categories are defined, for each value of the tuned threshold the total number of problematic pixels can be obtained by running a NoiseScan and a StuckPixelScan. This procedure can be repeated both in Synchronous and Asynchronous mode. In each mode, lowering the threshold will lead to an increased number of problematic pixels. However, according to our hypothesis, the number of noisy and stuck pixels will grow differently in different categories and in different modes. The results obtained with the procedure previously described are shown in Fig. 4.15. By

looking at Fig. 4.15a it can be noted that, as expected, in Synchronous mode the increase of the number of problematic pixels by lowering the threshold is the same through all the three categories. Therefore, Synchronous mode is not sensitive to comparator threshold (V_{Thr}) oscillations. On the other hand, Asynchronous mode is sensitive to these oscillations. This can be demonstrated by looking at Fig. 4.15b where it can be noted that in Low category the increase of the number of stuck and noisy pixels is larger compared to High and Typical category. The results previously described show also that, in general, the number of problematic pixels in Asynchronous mode is much bigger than the one obtained in Synchronous mode. The procedure described in this paragraph has been performed using a 3D sensor.

These results are really important because they show that the observed comparator threshold oscillation does not affect the noise performance in Synchronous mode, which is the sampling mode selected by CMS for its Phase-II upgraded tracker.



(a)



(b)

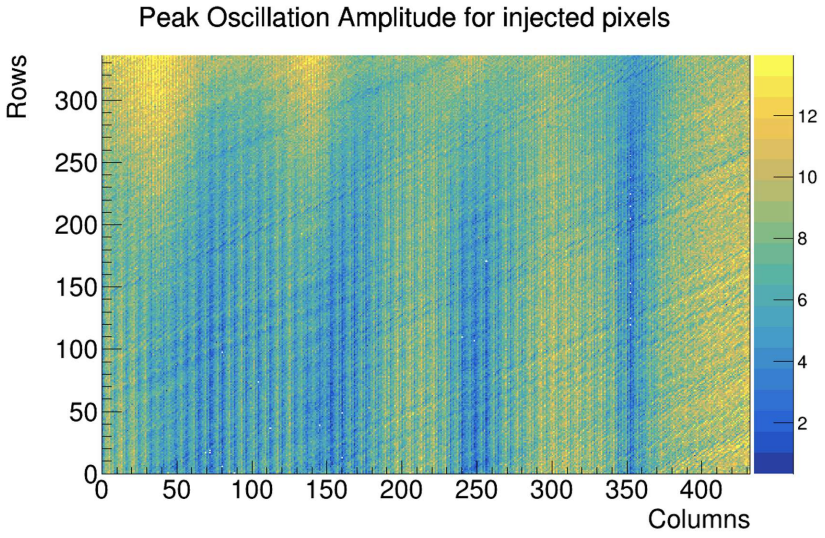
Figure 4.15: Number of problematic pixels in each category as a function of the tuned threshold: (a) Synchronous mode; (b) Asynchronous mode.

CHAPTER 5

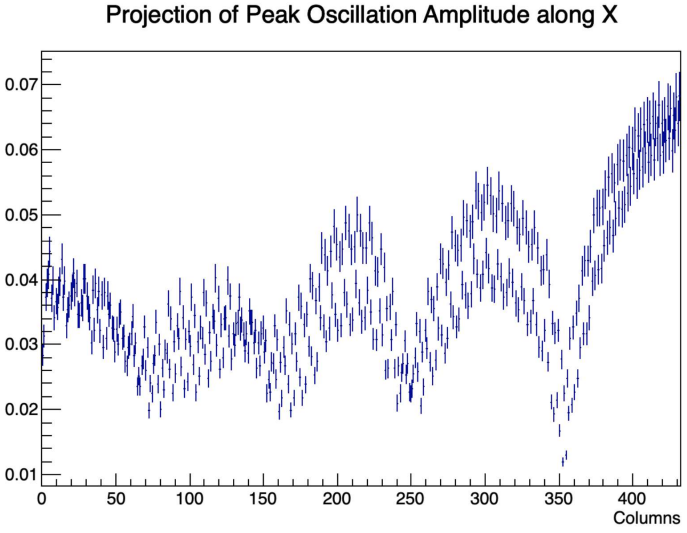
OSCILLATION ANALYSIS

5.1 Oscillation changes within the pixel matrix

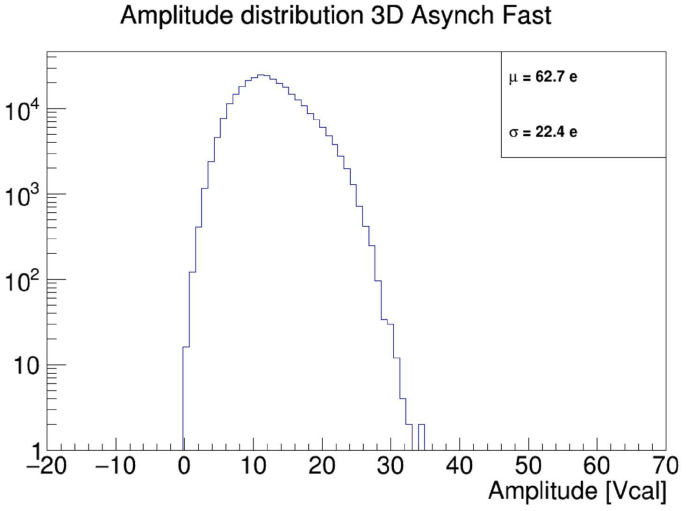
The parameters obtained from the fit described in the last chapter (i.e. oscillation peak amplitude, phase and baseline) can also be used to understand how V_{Thr} oscillation changes within the pixel matrix. In particular, the same procedure can be repeated changing the value of Krummenacher current. Two configurations have been used: Fast Discharge ($KRUM\_CURR\_LIN = 190$) and Normal Discharge ($KRUM\_CURR\_LIN = 80$). Since the goal of this analysis is to describe the oscillation of the comparator threshold, the Asynchronous mode has always been used. In this paragraph, the results obtained using a 3D sensor in Asynchronous mode with Fast Discharge will be discussed. For each parameter (amplitude, phase and average) three different plots have been made: the first plot is a map showing how that particular parameter changes within the pixel matrix (Fig. 5.1a, 5.2a and 5.3a), the second is a projection of the map along the x axis, i.e. along the columns (Fig. 5.1b, 5.2b and 5.3b), and the last one is the parameter distribution (Fig. 5.1c, 5.2c and 5.3c). By looking at the maps of the different oscillation parameters, it is clear that threshold oscillation in each pixel depend on where the pixel is with respect to the power distribution lines. Indeed, if the pixel is close to one of those lines, the oscillation amplitude is close to zero, while it grows as the distance from the distribution lines increases. This dependence results in a "column pattern". In addiction to this, a "diagonal pattern" can be observed in every map. This pattern is probably due to the injection mask used by the injection circuit.



(a)

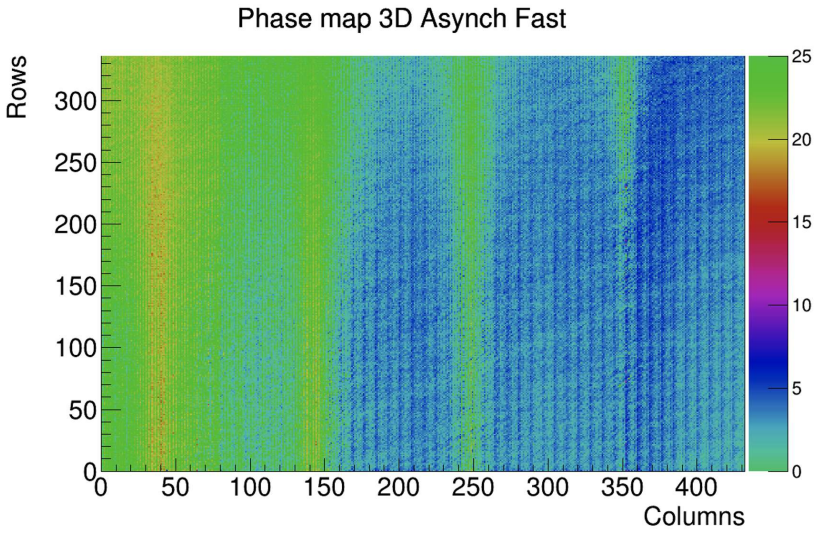


(b)

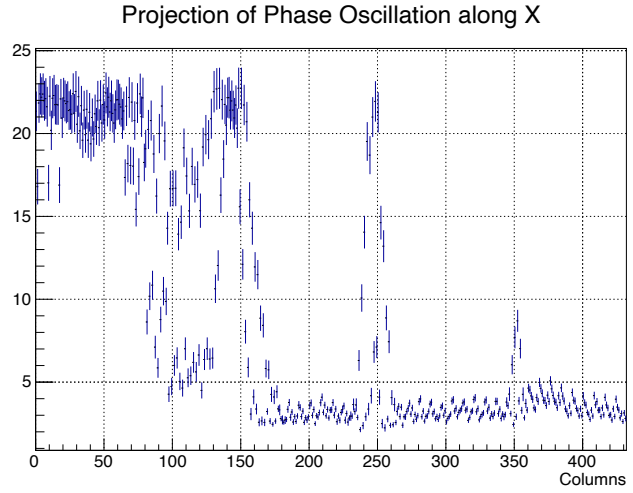


(c)

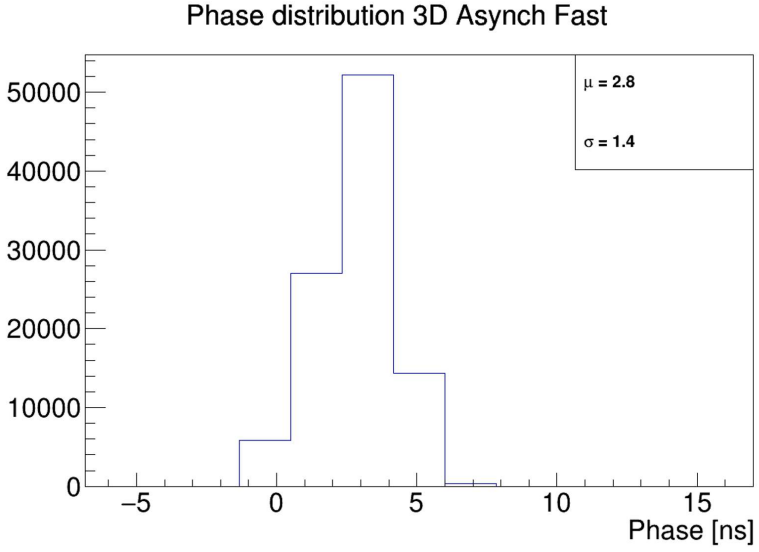
Figure 5.1: Map, projection and distribution of peak oscillation amplitude in Fast Discharge Mode



(a)

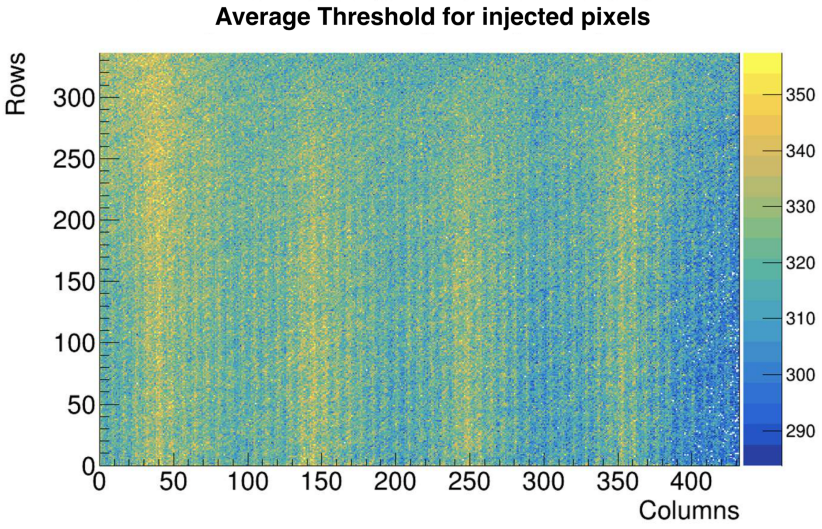


(b)

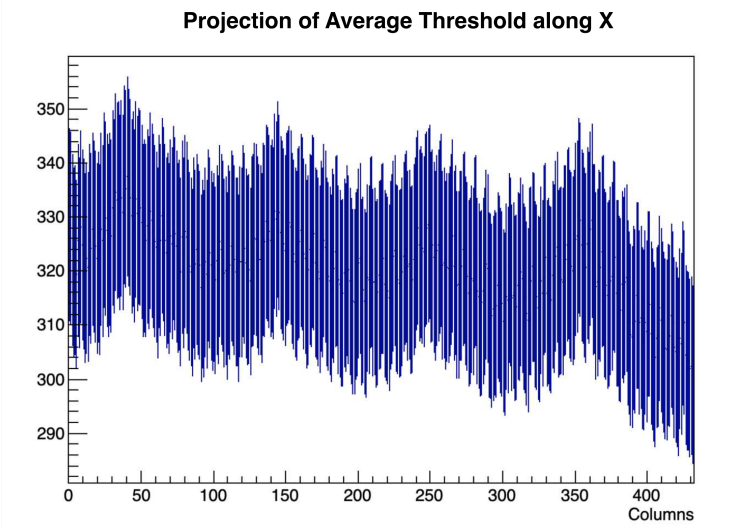


(c)

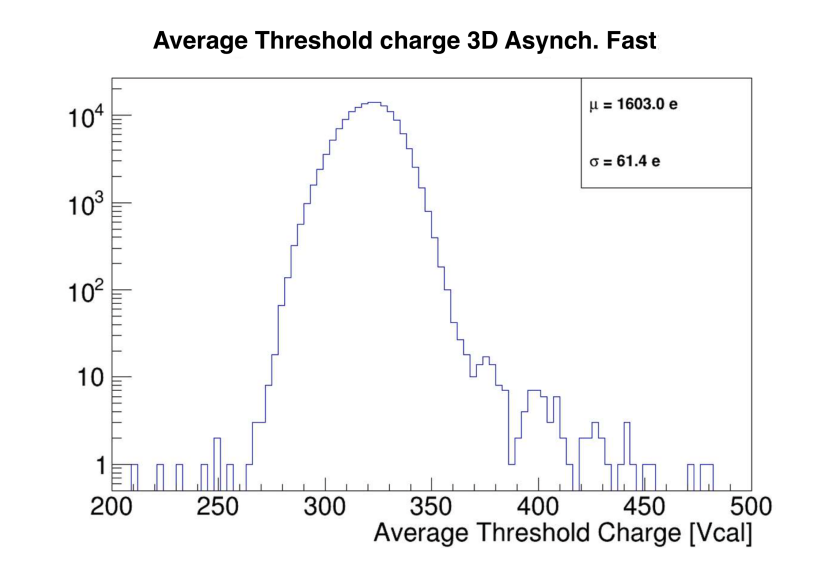
Figure 5.2: Map, projection and distribution of oscillation phase in Fast Discharge Mode



(a)



(b)



(c)

Figure 5.3: Map, projection and distribution of oscillation average in Fast Discharge Mode

5.2 Comparison with different sensors

The results previously discussed were obtained using a 3D sensor, injecting all the 145 152 channels and only in Asynchronous mode. In order to increase analysis's statistic and verify results repeatability, a larger number of sensors and bare chips must be tested, also in Synchronous mode (both with Fast and Normal Discharge). However, injecting all the pixels results in a really time-consuming scan (it can take several hours to end). So the number of injected pixels can be reduced (e.g, from 145 152 to 14 000). In this way is not possible anymore to study in detail the maps of oscillation parameters, but the sensor statistic can be increased within reasonable times.

5.2.1 Preliminary Checks

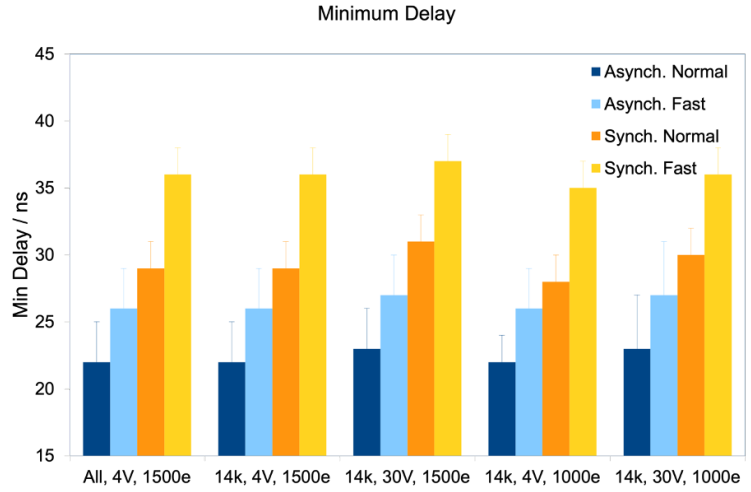
In order to compare the results obtained using different sensors, some preliminary checks must be done. In particular, we must be sure that oscillation parameters do not depend on:

- **The Number of injected pixels:** because results obtained injecting all channels will be compared with results obtained injecting only 14 000 channels.
- **The Sensor Bias Voltage:** this check must be done because one of the 3D sensors used was damaged, and it could not be powered at its nominal voltage. The voltage used in this case was however sufficient to bias the sensor.
- **The Tuned Threshold:** some results are obtained with a tuned threshold of 1500 e^-, since one of the planar modules used was too noisy to tune its threshold to 1000 e^-. So, this check must be done because results obtained with different tuned threshold have been compared.

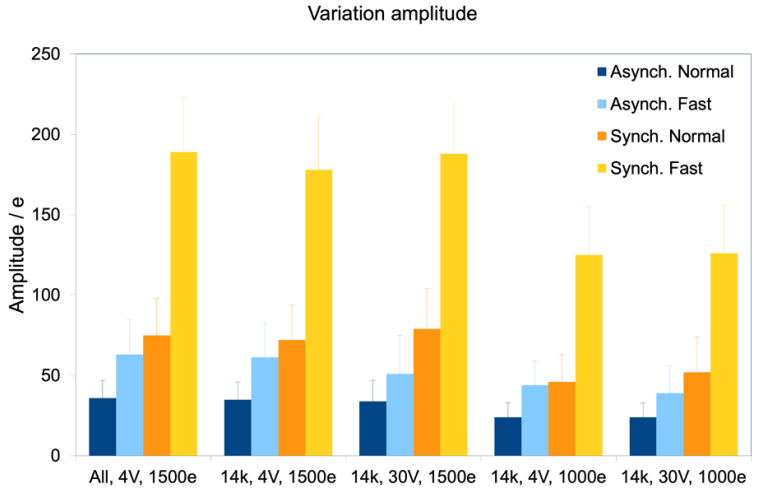
A priori, no dependence on the number of injected pixels is expected. Indeed, the injection procedure is just a loop which ends once all the pixels are injected; so reducing the number of injected channels just means exiting the loop earlier. But this should not change the behavior of the actual or the comparator threshold. The same argument stands for the dependence on bias voltage; it should not interfere with oscillations as long as the voltage is sufficiently high to deplete the sensor. At first glance, no hypothesis can be done on the dependence of oscillations on tuned threshold.

In order to fulfill the preliminary checks previously described, the first part of the procedure shown in the previous paragraph has been repeated changing the number of injected pixels, the bias voltage and the tuned threshold both in Synchronous ad Asynchronous mode and with Fast and Normal Discharge. The results obtained are shown in Fig. 5.4. As it can be noted by looking at Fig. 5.4a, as expected, the oscillation phase does not depend nor on the number of injected pixels nor on the bias voltage; moreover, no dependence on tuned threshold is observed. Even for oscillation amplitude, no dependencies are observed on the number of injected pixels and on the bias voltage, however, by looking at Fig. 5.4b it can be noted that

this oscillation parameter depends on tuned threshold. In particular, the higher is the tuned threshold and the higher is the oscillation amplitude.



(a)



(b)

Figure 5.4: Oscillation Phase (a) and Amplitude (b) changing the number of injected pixels, the bias voltage and the tuned threshold.

5.2.2 Oscillation Amplitude and Tuned Threshold

The unexpected dependence of oscillation amplitude on tuned threshold can be investigated further on tuning the threshold to values higher than 1500 e^-. By doing so, a linear dependence is observed, both in Synchronous and Asynchronous mode (Fig. 5.5).

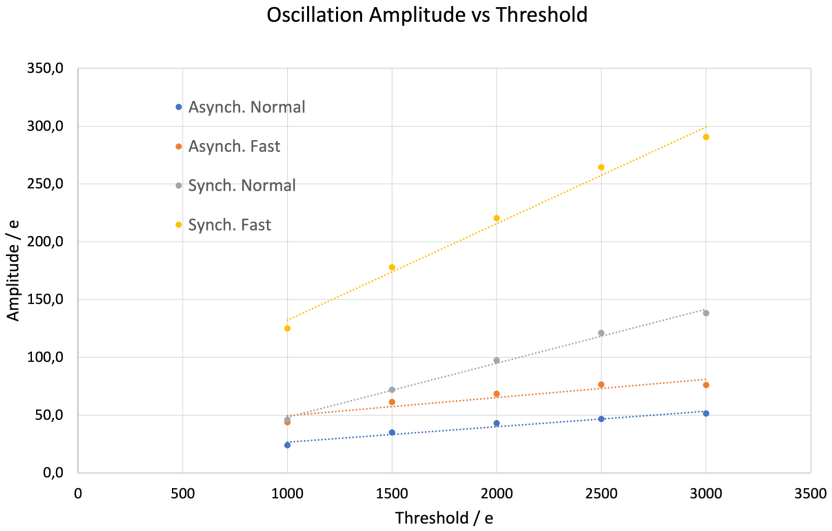


Figure 5.5: Description of the linear dependence of oscillation amplitude on the tuned threshold.

First, we focus on Synchronous mode, where the linear dependence can be explained taking into account, once again, how this sampling mode works. Suppose to have a signal S_1 with charge Q and delay t which exceeds the threshold exactly at a clock edge. Consider now a second signal S_2, with charge Q' and delay $t' > t$. In order to see S_2 in the reference bunch crossing, its charge must be higher than Q. So, since the only vertical scale is the threshold, everything should scale with it. This means that the higher is the threshold, the higher needs to be Q' in order to see the signal S_2 in the reference BX. So, the bigger will be the amplitude of the smallest detectable signal oscillation with the delay. The example previously described is sketched in Fig. 5.6.

However, the latter explanation could not be applied to Asynchronous mode. Indeed, in this sampling mode, it is still not clear why the oscillation amplitude depends on the tuned threshold.

5.2.3 Final Comparison

In light of the results obtained while running preliminary checks, the analysis has been repeated for all the sensors which threshold could be tuned to 1000 e^-, injecting only 14 000 channels. Fig. 5.7. shows the results obtained for all the tested modules. Some general features can be underlined. Both in Synchronous and Asynchronous mode, oscillations amplitude is bigger with Fast Discharge than with Normal Discharge. This result can be explained considering that with Fast Discharge the signal shape is narrow and this leads to a "finer sampling" of the oscillation. On the other hand, with Normal Discharge the signal shape is wider, and the oscillations are averaged out. This leads to a smaller amplitude, both in Synchronous and Asynchronous mode.

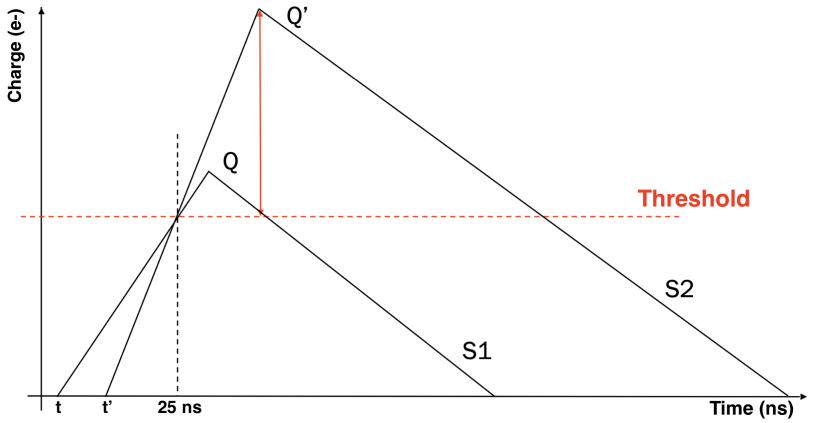


Figure 5.6: Sketch of the example used to explain the linear behavior of oscillation amplitude with respect to tuned threshold.

By looking at Fig. 5.7 it can also be noted that oscillations amplitude is larger in Synchronous mode than in Asynchronous mode, for a given discharge speed.

Overall, the amplitude is larger for planar sensors than for 3D sensors. Oscillations amplitude in a bare chip is compatible with the planar one. In order to understand why threshold oscillations are different from sensor to sensor, a more detailed description of oscillations shape is needed (see Chapter 6 for more details).

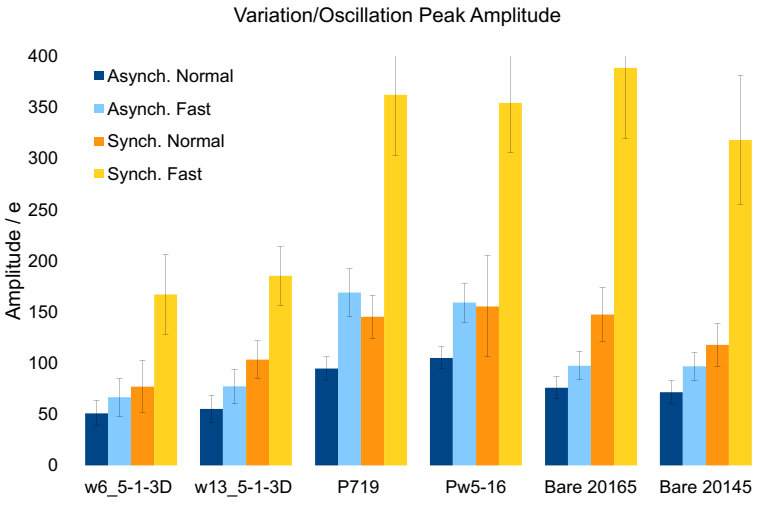


Figure 5.7: Threshold oscillation amplitude for different tested modules and different modes (Asynchronous, Synchronous, Fast Discharge and Normal Discharge).

CHAPTER 6

DETAILED DESCRIPTION OF OSCILLATIONS

In previous chapters, threshold oscillations has always been described by a cosine function. However, to explain sensors differences, a more detailed description is needed. To do so, a simplified toy model can be arranged to describe qualitatively the phenomenology.

In order to describe this toy model, it is important to define the signal shape that has been used. In particular, it is sketched in Fig. 6.1.

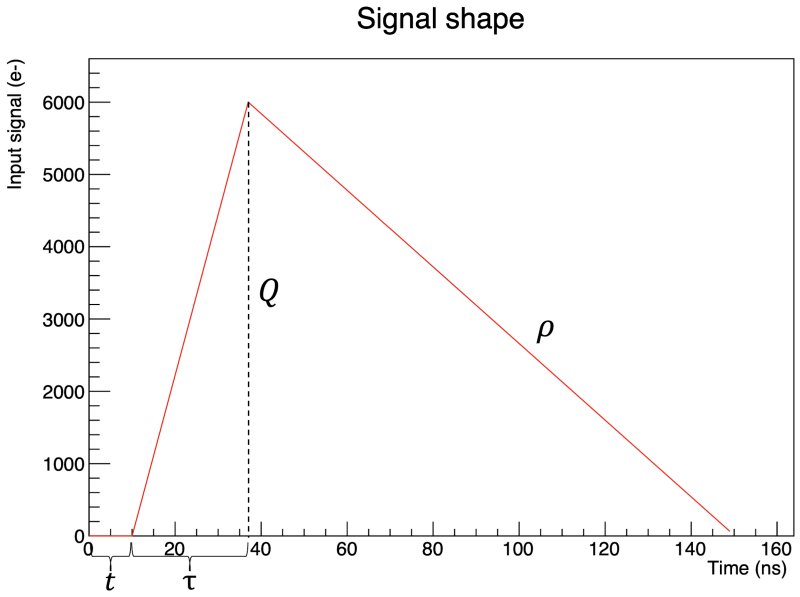


Figure 6.1: Signal shape used in the toy model. It is characterized by a charge Q, a rise time τ a delay t and a slope ρ. The higher is the Krummenacher current, the larger is $|\rho|$.

As it can be noted, it is characterized by: a charge Q (e.g., 6000 e^- in the sketched signal), a characteristic Rise Time τ (equals to 27 ns in the example), an injection delay t and a slope ρ which depends on Krummenacher current.

6.1 Production of Synchronous mode

To describe how the toy model works, suppose to focus on a single bunch crossing (BX) and to tune the target threshold to a value of 1000 e^-. Consider, initially, the Synchronous mode only. In these conditions, it is possible to define a "lucky delay", t_0, which is the delay at which a signal of charge Q_0 slightly exceeds the threshold at the clock edge. Consider now a second signal with charge Q and delay t. Since the Synchronous mode has been selected, if $t > t_0$, the charge Q needs to be larger than Q_0 in order to see that signal in the reference BX. This situation is the same sketched in Fig. 5.6. The bigger is t, the bigger is Q. At limit, when t is equal to the clock edge time, Q diverges. This effect can be called *Time Walk Effect*, since the time walk curve describe exactly the behavior of t_w (defined as the time that the signal takes to exceed the threshold) as a function of the injected charge. The same considerations can be done if the second signal has a delay $t < t_0$. Indeed, even in this second case, its charge needs to be bigger than Q_0, in order to see the second signal in the reference bunch crossing. However, in this second case, a new parameter can be taken into account, that is Krummenacher current. Suppose to have a signal with delay $t < t_0$ which exceeds the threshold, but falls below it before the clock edge. In this case, to see the signal in the reference BX, it is not necessary to increase the charge, but it is sufficient to decrease the Krummenacher current. In this way, the ToT increases and the signal could be above threshold at the clock edge. This second effect can be called *Krummenacher Effect*, and it is sketched in Fig. 6.2.

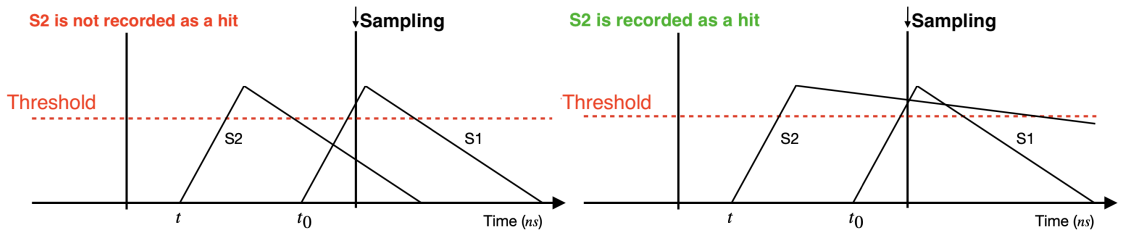


Figure 6.2: Graphical sketch of the Krummenacher effect: a signal with delay $t < t_0$ can be recorded as a hit in the reference bunch crossing if its ToT increases. To do so, the value of the Krummenacher current needs to be smaller.

Simulating the Synchronous mode in the toy model, the *actual Threshold* can be described as a function of the injected delay. Fig. 6.3 shows the threshold behavior for a single bunch crossing with two different values of the Krummenacher current. Both with Fast and Normal Discharge, the right-side edge of the plot is due to Time Walk Effect, while the Krummenacher effect is responsible for the left-side edge. With Fast Discharge and fixed delay, the charge of the signal needs to be bigger than with Normal Discharge to see the signal in the reference BX. So, the bigger is the Krummenacher, the bigger is the slope of the left-side edge of the plot.

The result previously shown was referred to a single bunch crossings. However, the toy model can reproduce the measurement of the actual threshold for several

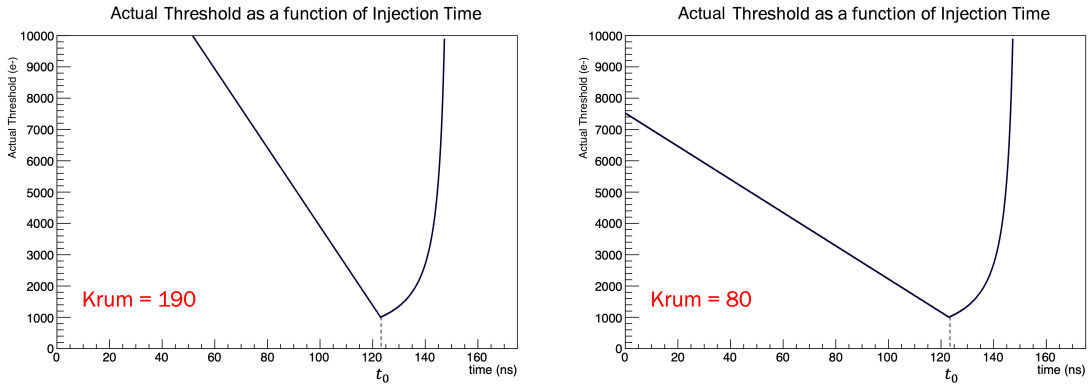


Figure 6.3: Actual Threshold behavior obtained with the toy model. Two discharge speeds are shown: Fast Discharge (KRUM = 190, left) and Normal Discharge (KRUM = 80, right).

bunch crossings. By doing so, an oscillation of the actual threshold can be observed considering the overlay of all the curves (Fig. 6.4). It is important to note that in Synchronous mode we are not sensitive to the oscillation of the comparator threshold (V_{Thr}), and that the observed oscillation of the smallest detectable signal with the delay is expected, because of how Synchronous mode works.

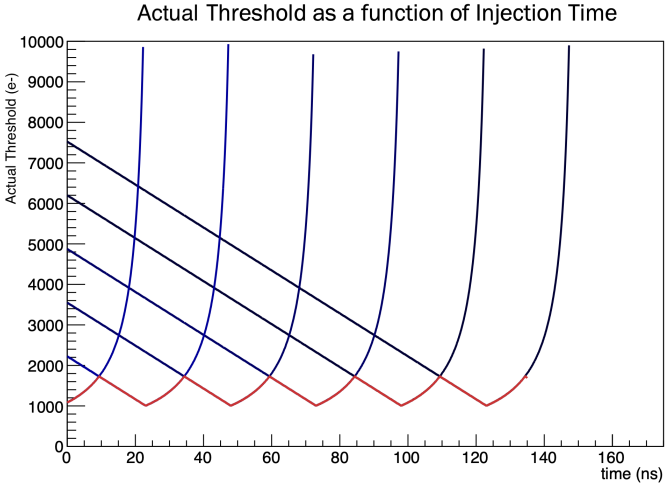


Figure 6.4: Behavior of the actual Threshold for several bunch crossing, i.e. clock cycles. The red line is the overlay of all the curves and represents the threshold oscillation which is observed in Synchronous mode.

6.1.1 Actual Threshold Oscillation and Krummenacher current

As said previously, the value of the Krummenacher current can change the behavior of the actual threshold as a function of the injected time. In particular, the smaller is the Krummenacher current, the smaller is the slope of the left-side edge of the plot. This change will affect threshold oscillation too. Indeed, the smaller is the

slope, the smaller the actual oscillation amplitude will be.

This feature can be experimentally verified by running several *TimeWalk* scan extending the time range beyond a single clock cycle, i.e. [0,60]ns, and changing every time the value of the register *KRUM_CURR_LIN*. Fig. 6.5 shows the experimental result (on the left) obtained considering a reference channel compared to the simulated behavior (on the right) obtained from the toy model. The behavior of actual threshold oscillation with the Krummenacher current is reasonably well described by the toy model. However, simulated oscillation amplitude does not match the experimental amplitude. This is due to the extremely simplified signal shape used in the toy model.

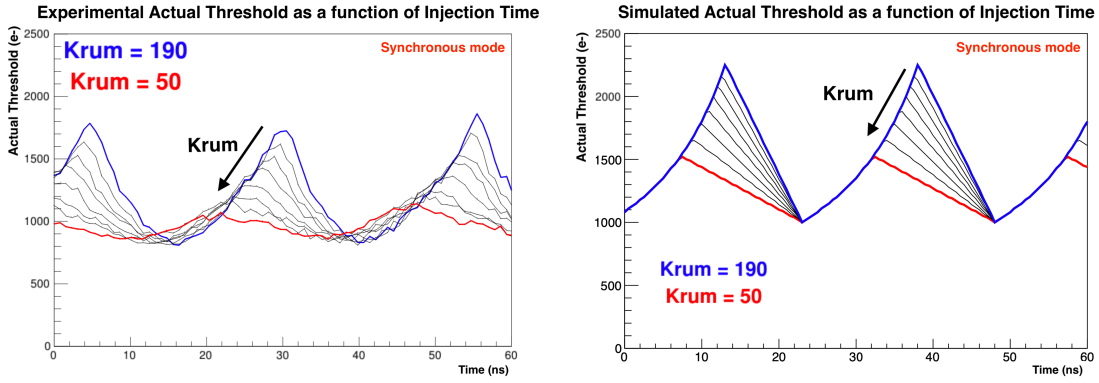


Figure 6.5: Actual threshold oscillation behavior for various Krummenacher values for both experimental (left) and simulated (right) oscillation. The Krum values 190 and 50 refer to the value of the register set in the configuration file.

The sensor used for the measurement previously described is a planar sensor (in particular, sensor P719). For the measurement, the threshold was tuned to 1000 e^-.

6.2 Production of Asynchronous mode

The toy model can qualitatively reproduce the behavior of the Asynchronous mode as well. To describe what is expected in this mode, consider again a signal with charge Q_0 and delay t_0, which exceeds the threshold at the clock edge. Like in Synchronous mode, also in Asynchronous mode a signal with delay $t > t_0$ needs to have a charge Q bigger than Q_0 in order to be seen in the reference BX. This means that the Time Walk effect described in paragraph 6.1 is expected to be seen also in Asynchronous mode. However, since in this sampling mode a hit is recorded when a signal exceeds the threshold, regardless of its delay, a signal with charge $Q = Q_0$ and delay $t < t_0$ will be recorded as a hit in the reference BX until its delay t is bigger than $(t_0 - 25\,\text{ns})$. When $t < (t_0 - 25\,\text{ns})$ the signal with charge Q is recorded as a hit in the reference BX only if its ToT increases (Krummenacher effect). This behavior is sketched in Fig. 6.6.

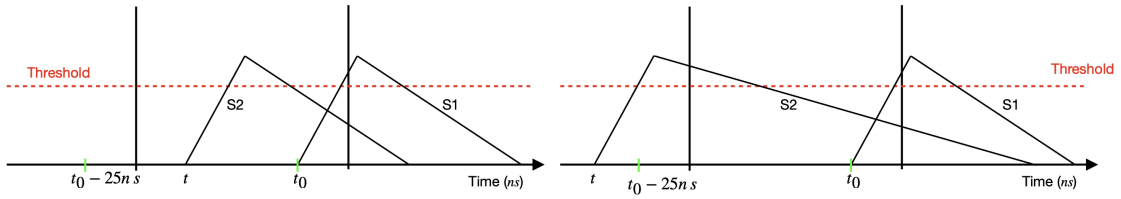
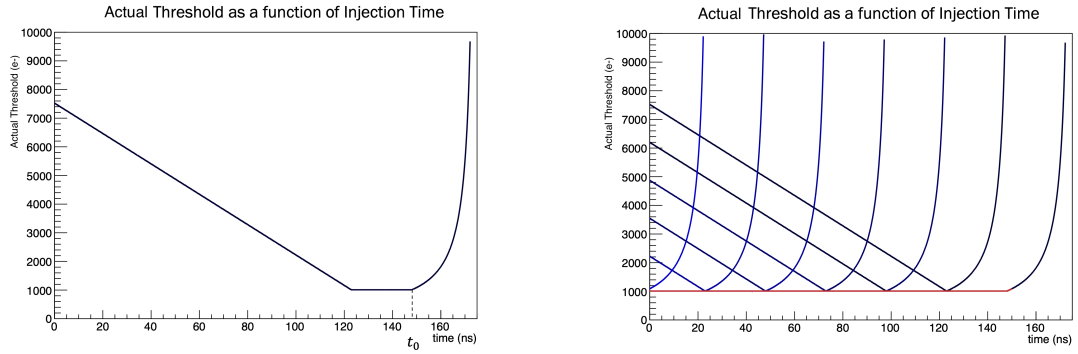


Figure 6.6: In Asynchronous mode, the signal S_2 is recorded as a hit until $t_0 - 25\,\text{ns} < t < t_0$ (left). When $t < t_0 - 25\,\text{ns}$, S_2 is recorded as a hit only if the value of the Krummenacher current decreases (right).

The behavior of the actual threshold as a function of the injected delay in Asynchronous mode is shown in Fig. 6.7a. As previously said, in Asynchronous mode the Time Walk effect and the Krummenacher effect are separated by a plateau which lasts exactly 25 ns. The described behavior is referred to a single BX, however, as in Synchronous mode, many bunch crossings can be taken into account. Considering the overlay of all the curves, in Asynchronous mode, no actual threshold variation is expected. As it can be noted by looking at Fig. 6.7b, the threshold should not change nor by changing the injection delay nor by changing the Krummenacher current, because of the presence of the 25 ns plateau.



(a) (b)

Figure 6.7: Simulation results in Asynchronous mode: (a) Actual Threshold behavior simulated considering a single reference BX; (b) Overlay of multiple bunch crossing. No oscillation is expected in Asynchronous mode.

6.2.1 Comparator Threshold Oscillation

The results obtained using the toy model show that, as expected, in Asynchronous mode the actual threshold should not vary with injection delay. However, as shown in the previous chapters, a threshold oscillation has been experimentally observed in Asynchronous mode too.

To explain the experimental results, an oscillation of the comparator threshold (V_{Thr}) can be taken into account. To do so, the toy model can be modified by considering an oscillating comparator threshold, instead of a constant one. The formula which

describes this oscillation is the following:

$$Thr \;=\; Thr_0 \;+\; A \times cos\left(2\pi\;\frac{t-\phi}{25}\right) \tag{6.1}$$

In the latter equation, Thr_0 represents the constant value of the comparator threshold used in the previous simulations, while A and ϕ are the oscillation amplitude and phase, respectively. The oscillation frequency has been set to 40 MHz.

By adding this oscillating comparator threshold in the toy model, an oscillation can be noted by looking at the overlay of the plots for many BXs (Fig. 6.8).

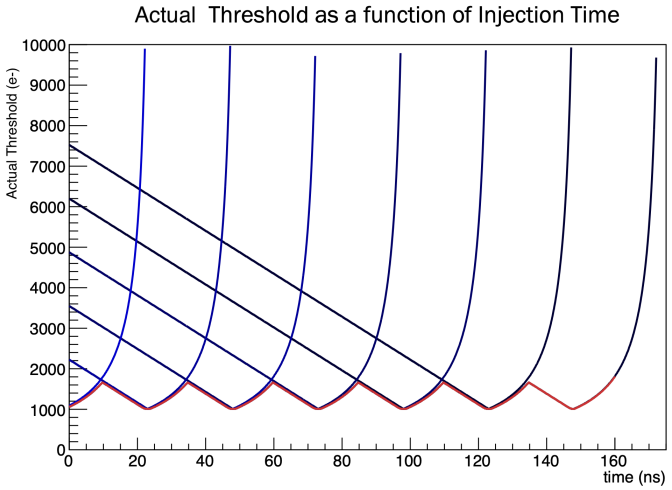


Figure 6.8: Behavior of the actual threshold in Asynchronous mode considering an oscillating comparator threshold.

As the amplitude of the oscillation observed in Synchronous mode, also the oscillation amplitude in Asynchronous mode depends on Krummenacher current. In particular, oscillation amplitude gets lower for lower I_{Krum}.

To demonstrate that the results obtained by the modified toy model reflect the experimental behavior, the same measurement described in the previous paragraph can be repeated. The only difference is that the Asynchronous mode is selected in the latter case. Fig. 6.9 shows the experimental results (on the left) compared to the simulated one (on the right). As for Synchronous mode, the modified toy model describes the behavior of threshold oscillation with Krummenacher current. Again, the different amplitude is due to the oversimplified signal shape which has been used.

The sensor used to obtain the results previously described is the same used for the measurements in Synchronous mode.

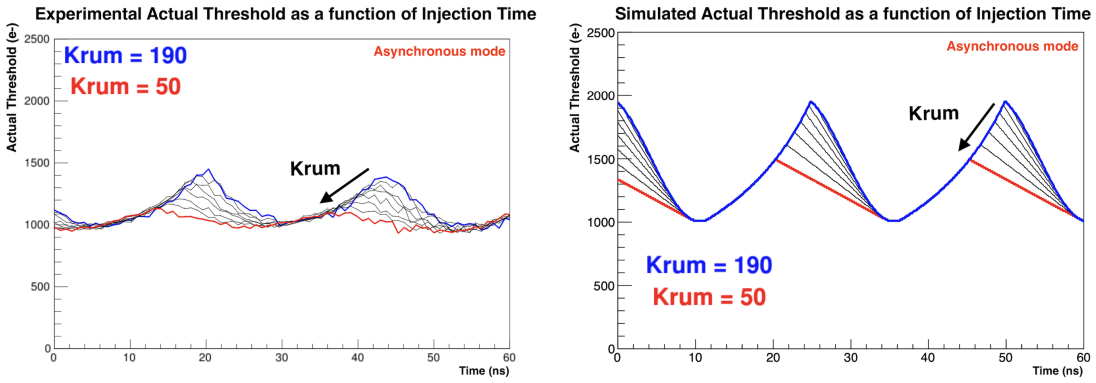


Figure 6.9: Actual threshold oscillation behavior with Krummenacher current for both experimental (left) and simulated (right) oscillation.

6.3 Explanation of the differences between sensors

The toy model previously described can be used to understand why the amplitude of threshold oscillations changes with the sensor type. Indeed, as shown in Fig. 5.7, the amplitude is bigger for planar sensors and bare chips than for 3D sensors.

Consider, initially, the Synchronous mode only. As previously said, the behavior of the actual threshold oscillation with injection delay is due to two combined effects: Time Walk (TW) effect and Krummenacher effect. Let's see how these effects can be influenced by the sensor properties, starting from the Time Walk effect. In general, TW is proportional to the pixel capacitance. The bigger is the latter, the slower is the sensor and the bigger is the time walk[1]. So, a higher oscillation amplitude is expected for slower sensors. In a planar sensor, the sensor's capacitance is close to 2.5 fF, while in a 3D sensor it is much bigger; indeed, it can be close to 40 fF. The capacitance of a bare chip is zero. However, in order to compare the speed of different modules, the feedback capacitance must be taken into account too. The latter is more or less the same for all the modules, and it is close to 6.2 fF. So, the total capacitance of the modules used is the following:

$$\begin{aligned} C_{BareChip} &\approx 6.2\ fF \\ C_{Planar} &\approx 2.5\ fF\ +\ 6.2\ fF\ \approx 8.7\ fF \\ C_{3D} &\approx 40\ fF\ +\ 6.2\ fF\ \approx 46.2\ fF \end{aligned} \tag{6.2}$$

According to what previously shown, the Bare Chip is the fastest module, so it should have the smallest oscillation of the actual threshold. At the same time, the 3D sensor should have the highest threshold variation because it is the slowest module. Moreover, threshold oscillation in a planar sensor should be slightly bigger than threshold oscillation in a bare chip. However, it is worth to be noted that the total capacitance of a planar module is similar to the total capacitance of a 3D

[1] When we say that a module is *slow*, we mean that its capacitance is large and so the time needed to charge it is bigger.

module if the inter pixel capacitance is taken into account too. Please note that the interpixel capacitance of a 3D module is smaller with respect to the planar one, since the electrodes (with the same polarity) in a planar module are closer than the electrodes in a 3D module. By taking into account all these effects, we can conclude by saying that the Time Walk effect is similar for planar and 3D modules.

Consider now the Krummenacher effect. Even setting the same value of register *KRUM_CURR_LIN*, the slope of the left-side edge of the plot shown in Fig. 6.3 can slightly change with the sensor. This is due to the fact that for low injected charges (i.e., $Q \lesssim 2000\,e^-$) the preamplifier output can not be described as a triangular wave. It can be better described as a difference between two exponential with different time constant:

$$[S(t)]_{Q\lesssim 2000\,e^-} = Q_{Max} \times \left(e^{-\frac{t}{\tau_{fast}}} - e^{-\frac{t}{\tau_{slow}}} \right) \tag{6.3}$$

where Q_{Max} is the charge injected in the channel. In Eq. 6.3, the time constant τ_{slow} is due to Krummenacher current, while τ_{fast} is linked to the capacitance of the sensor. In particular, the bigger is the capacitance, the bigger is τ_{fast}. According to what previously said, changing the module could result in different signal shapes for small injected charges. Fig. 6.10 shows the predicted signal shape obtained fixing the value of the Krummenacher current and the injected charge for two different values of τ_{fast}.

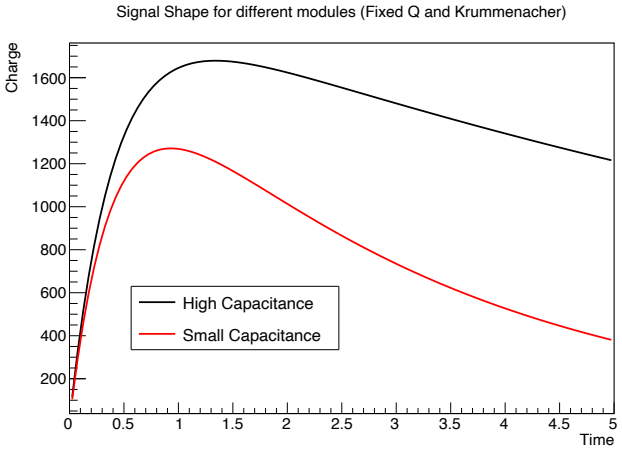


Figure 6.10: Predicted signal shape obtained by fixing the charge and the value of the Krummenacher current and changing only the capacitance. Both the charge and the time are in arbitrary units.

As it can be noted by looking at the latter figure, the bigger is the sensor capacitance and the slower is the return to baseline of the signal. At the end, it can be said that the bigger is the capacitance of the sensor, the smaller is the slope of the left edge of Fig. 6.3.

According to what we obtained, the amplitude of threshold oscillation is expected to be smaller for 3D sensors than for planar sensors. For what concerns a bare chip, the amplitude of threshold oscillation is similar to planar's amplitude. Considering both the Time Walk effect and the Krummenacher effect, we expect to see an actual

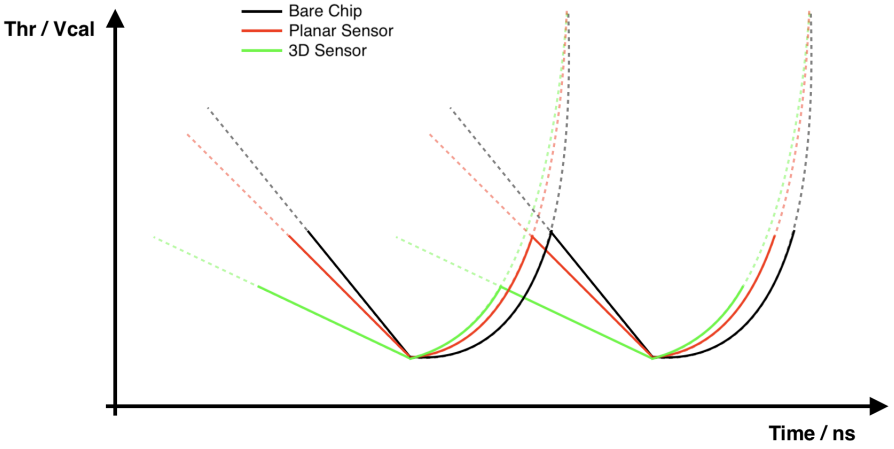


Figure 6.11: Expected threshold variation for different modules in Synchronous mode.

threshold oscillation like the one sketched in Fig. 6.11. The expected behavior described can be verified by measuring the actual threshold oscillation for different modules, using the *ThresholdOscillation* scan and referring to a single channel in Synchronous mode. The results of this latter measurement are shown in Fig. 6.12. As it can be noted, both Time Walk effect and Krummenacher effect can be used to qualitatively understand why the observed oscillation amplitude in a 3D sensor is smaller than the oscillation amplitude observed in a planar sensor or in a bare chip.

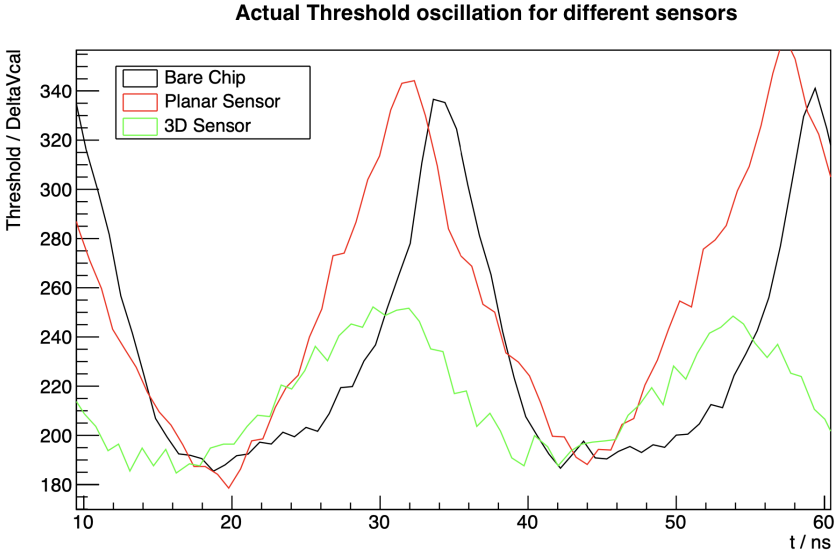


Figure 6.12: Comparison between threshold variation of different modules in Synchronous mode.

6.4 Origin of Threshold oscillation

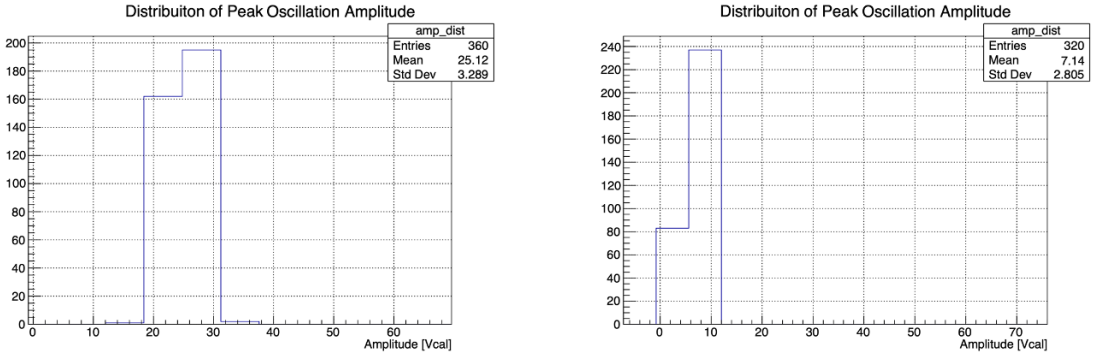
In previous paragraphs, the oscillations of the actual threshold in Synchronous mode and the oscillations of the comparator threshold in Asynchronous mode have been described. While the former is an expected behavior due to how Synchronous mode works, the latter is completely unexpected in Asynchronous mode. In order to understand what could be the origin of comparator threshold oscillation, the clock generation and distribution within the matrix can be studied.

6.4.1 Clock distribution within pixels matrix

The distribution of the pixel's digital part within the matrix is similar to the distribution of their analog part. Indeed, as low voltages, also the 40 MHz clock is first distributed to the core columns.

In order to see if the comparator threshold oscillation phenomenon is linked to how the clock is distributed to the matrix, the oscillation amplitude of channels belonging to the same reference core columns can be observed in two different cases: all core columns enabled and only a reference core column enabled.

The measurement previously described has been performed for different modules; Fig. 6.13 shows the results obtained using the bare chip 20165. As it can be noted by looking Figures 6.13a and 6.13b the mean value of the distribution of threshold oscillation amplitude obtained if only the reference core column is enabled is significantly smaller than the mean value of the same distribution obtained if all core columns are enabled.

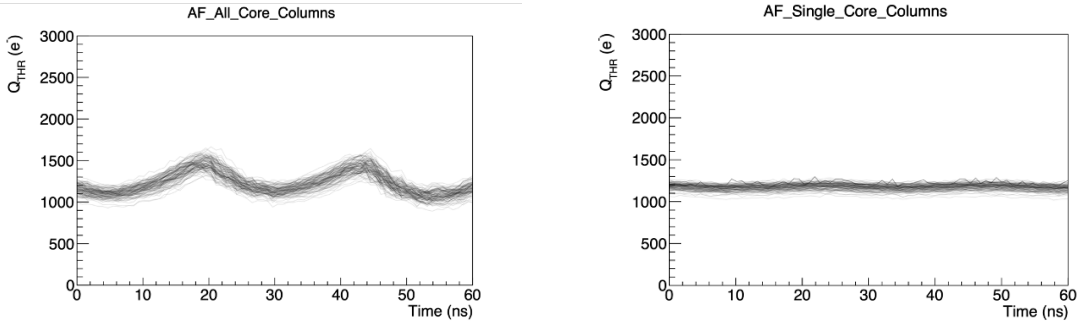


(a) (b)

Figure 6.13: Distribution of threshold oscillation amplitude in two cases: (a) All core column enabled; (b) Single core column enabled

This difference is even more evident by looking at Fig 6.14, where the oscillations of all injected channels are superimposed in a single plot. In the latter figure, it is evident that V_{Thr} oscillation is essentially absent if a single core column is enabled.

In the results discussed previously, the threshold was tuned to 1000 e^-, using the Optimal Fine Delay, and the Krummenacher current register was set to 190 to have



(a) (b)

Figure 6.14: Overlay of threshold oscillation in two cases: (a) All core column enabled; (b) Single core column enabled

a Fast Discharge of the preamplifier's capacitance. In particular, Fast Discharge has been selected because with this setting the oscillation amplitude is bigger with respect to the Normal Discharge (as shown in paragraph 5.2).

6.4.2 Clock Distribution and Synchronous mode

The measurement described in the previous paragraph is performed with Asynchronous mode. This is due to the fact that we wanted to study the oscillations of the comparator threshold, that can be observed only with that sampling mode. However, the same procedure can be repeated also in Synchronous mode. As described in paragraph 6.1, actual threshold oscillation, which is observed in Synchronous mode, is due to two main effects: Krummenacher effect and Time Walk effect. When a core column is disabled, only its digital part is turned off. So, since the analog part of a core column is enabled even when the core column is disabled, no differences are expected in threshold oscillation amplitude if all core columns are enabled or if only a reference core column is enabled.

Fig. 6.15 shows the comparison between threshold oscillation in the two cases for a bare chip (chip 20165) and for a planar sensor (module FBK-w5_1-6).

As it can be noted by looking at Fig. 6.15a and 6.15b the amplitude of actual threshold oscillation if all core columns are enabled, as expected, is compatible with the oscillation's amplitude if a single core column is enabled. However, if the same comparison is done considering the module with sensor, things are different. Indeed, Fig. 6.15d shows that if a single core column is enabled, the amplitude of actual threshold oscillation decreases significantly.

A possible explanation to what has been observed can be found once again in what causes the actual threshold oscillation. In paragraph 6.1, the capacitance has been described as an element which slows down the module, especially for a planar sensor. It is possible that injecting a single core column reduces the inter-pixels capacitance, making the module faster. But, as described in the previous chapter, if a module is faster, the Time Walk variation is smaller. So, fixed the module and the value of the Krummenacher current, a faster module is characterized by a smaller actual

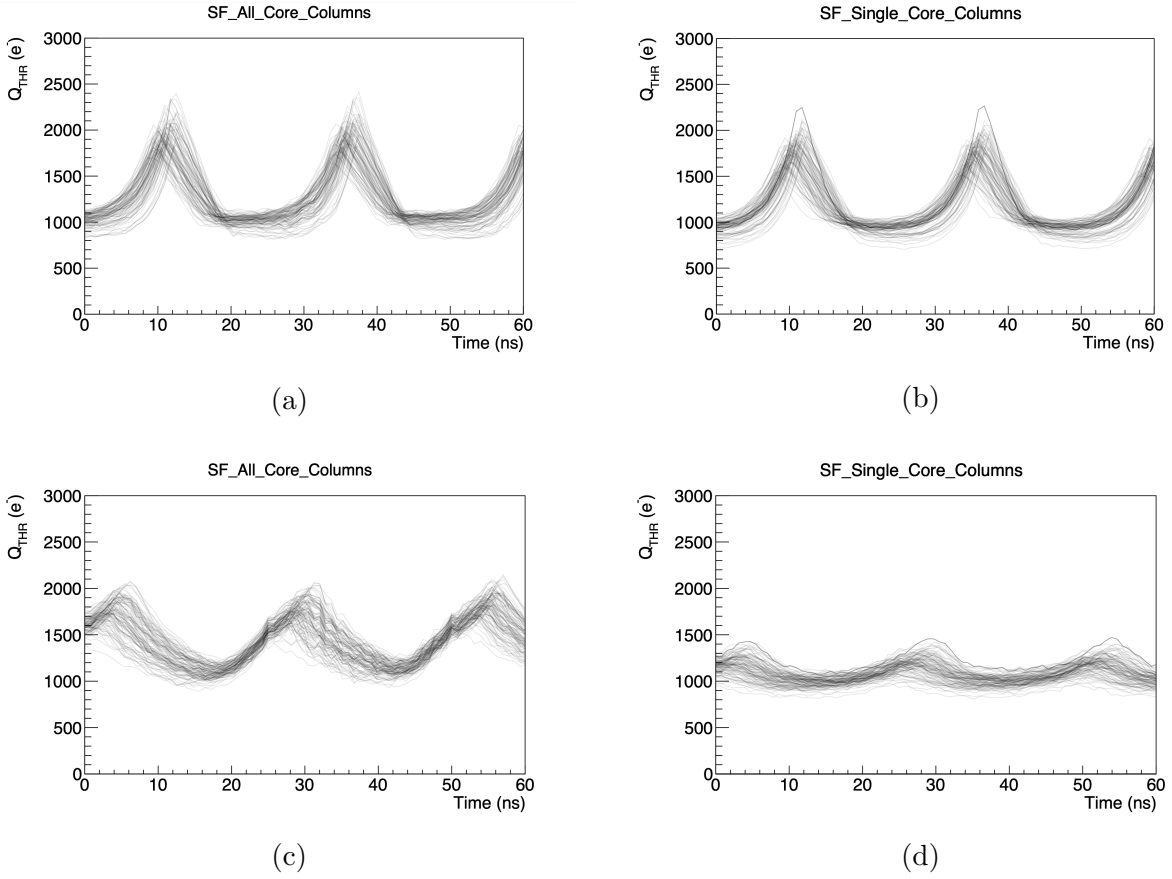


Figure 6.15: Actual threshold oscillation for a planar sensor and a bare chip, considering different configurations: (a) Bare chip with all core columns enabled; (b) Bare chip with single core columns enabled; (c) Planar sensor with all core columns enabled; (d) Planar chip with single core columns enabled.

threshold oscillation. The explanation previously described is sketched in Fig. 6.16. Since the inter-pixels capacitance cannot be defined in a bare chip (because it has no sensor), its speed will not change by changing the number of enabled core columns.

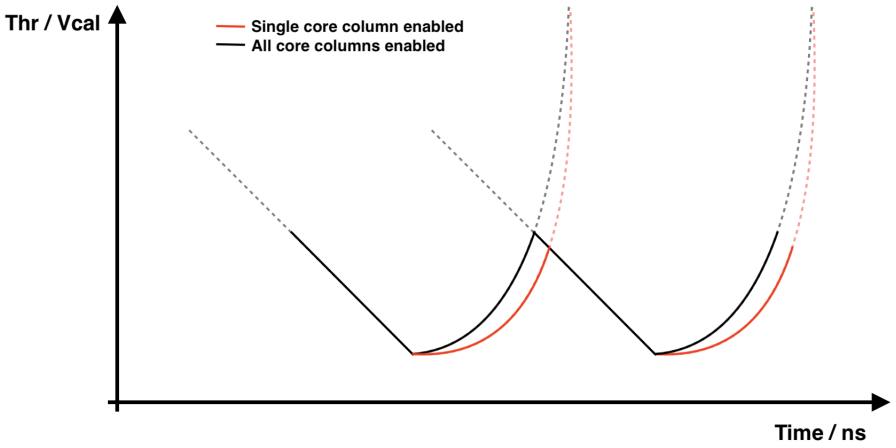
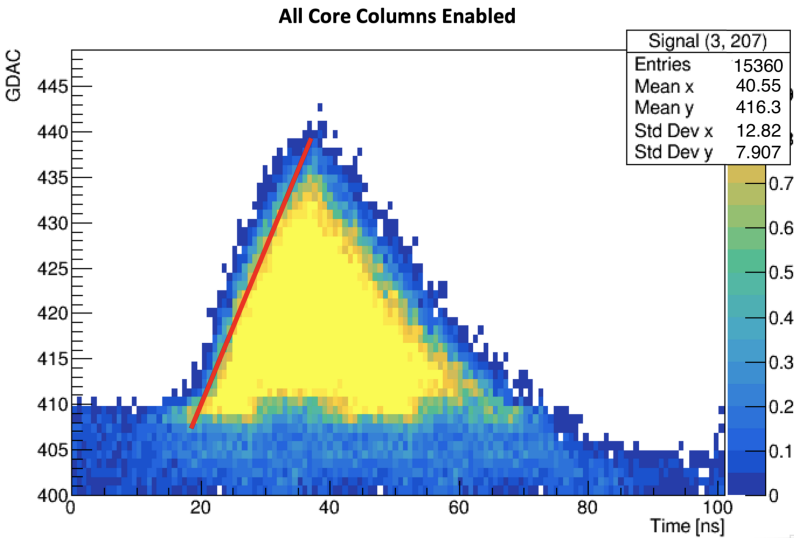
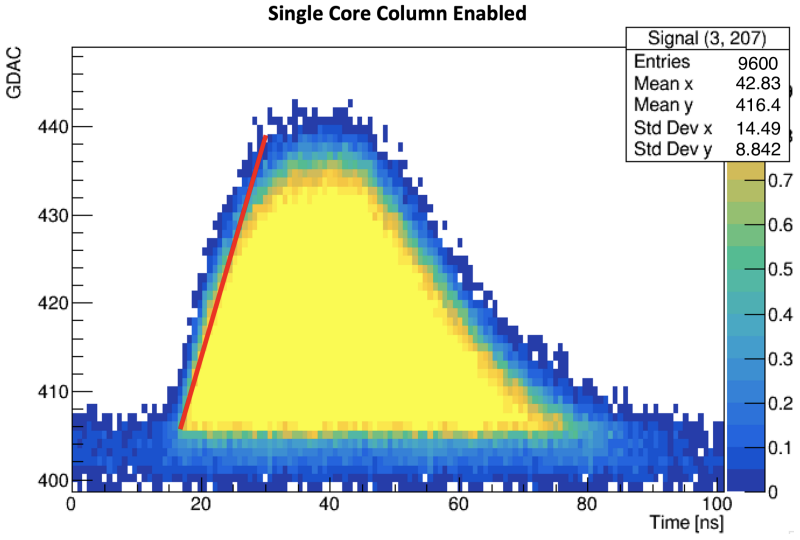


Figure 6.16: Sketch which shows that if a single core column is enabled the TW variation is smaller and this will result in a smaller amplitude oscillation of the smallest detectable signal.

If the hypothesis described previously is true, a variation in the signal shape should be observed. Indeed, if the module is faster, the rise time of the signal should be smaller. A *SignalShape* scan can be used to compare the shape of the signals, changing the number of enabled columns. Fig. 6.17 shows the shape of the signals which can be obtained injecting into the planar sensor a charge of $1500\,e^-$. By looking at Fig. 6.17 it is evident that the signal has a shorter rise time if a single core column is enabled. This result confirm the hypothesis previously described.



(a)



(b)

Figure 6.17: Shape of the signal induced in a planar sensor by a charge of 1500 e^- in two cases: (a) All core columns enabled; (b) Single core column enabled. By looking at the red lines, it is evident that the rise time of the signal is smaller if a single core column in enabled. This means that in the latter case the module is faster.

CONCLUSIONS

The main purpose of my thesis work is the development of calibration techniques and the performance analysis of the CMS read-out chip for the CMS Phase–II Inner Tracker. To do so, several detector modules with different pixel sensors have been used.

In the first part of this work I characterized the modules under test by performing sone preliminary measurements. Among these, I measured the conversion factor between V_{cal}, an internal chip voltage that represents a basic unit, and electrons. The nominal figure is 5.5 $\frac{e^-}{V_{cal}}$, however, I have found out that it is actually closer to 5 $\frac{e^-}{V_{cal}}$, i.e. 10% smaller. This discrepancy is not negligible with respect to calibration of the detector. My measurements also show that this conversion factor can slightly change module by module. Other important conversion factors are the one from GDAC to V_{cal} and the one from TDAC $\cdot$ V_{LDAC} to V_{cal}. The last measurement of this kind performed in this work is the one concerning Krummenacher current. Indeed, it is useful to know for each module how to convert the value of the register *KRUM_CURR_LIN* to a current measured in $\frac{e^-}{ns}$.

My work has then focused on the importance of the parameter Fine Delay within the tuning of the target threshold. Indeed, if the wrong value of the Fine Delay is set during threshold tuning, the comparator threshold will be smaller with respect to the target threshold. This has been verified by using the occupancy plot vs. signal magnitude and delay, known as *tornado* plot. Moreover, the measurements performed in my thesis work show that the choice of the Fine Delay could affect also the noise performance of the module. Indeed, if the wrong value of that register is set, the number of problematic pixels (i.e. the sum of stuck and noisy pixels) can be up to an order of magnitude larger. Given the importance of the Fine Delay, within this work I conceived a procedure to quickly find the optimal value of the Fine Delay for the module being calibrated. This procedure must be run before threshold tuning.

The results obtained in my thesis also show that even if the Optimal Fine Delay is set, the comparator threshold can be considered as a lower limit, since the smaller

detectable signal depends on the injection time. However, all the high transverse momentum particles produced in p-p collision can see the same value of the threshold, since the distribution of their time of arrival is narrow.

The other main topic covered in my thesis is threshold oscillations. According to the measurements I did, an oscillation of the actual threshold with respect to the injection time is expected in Synchronous mode. This behavior cannot be eliminated since it is essentially due to the fact that in this mode, the signal is sampled every 25 ns. In order to understand the phenomenology of the threshold oscillation, I set up a simplified toy model. The results obtained show that this behavior depends only on Krummenacher current and on Time Walk. In particular, the lower is the Krummenacher current, the smaller is the peak-to-peak oscillation amplitude.

The threshold oscillation analysis has been repeated for different modules, and the results obtained show that the oscillation amplitude is smaller in 3D sensors than in planar sensors or bare chip. This difference can be explained considering that for small signals ($Q \lesssim 2000\,\mathrm{e}^-$) the return to baseline depends not only on Krummenacher, but also on the capacitance of the sensor. In particular, the bigger is the capacitance, the slower is the return to baseline. So, since the 3D sensors have the higher capacitance, this should be enough to explain the differences in terms of oscillation amplitude.

On the other hand, in Asynchronous mode, the actual threshold is expected to be constant, regardless of the injection time. However, the results I got show an oscillation of the actual threshold also in Asynchronous mode. The behavior of the threshold in Asynchronous mode can be explained only considering an oscillation of the comparator threshold (V_{Thr}), which amplitude decreases with the Krummenacher current.

In this work, I've shown also that noise performance in Synchronous mode is not affected by the oscillation of the comparator threshold. This is due to the fact that this oscillation has a 40 MHz frequency, so, since in Synchronous mode the signal is sampled each 25 ns, at every clock edge the value of the comparator threshold is always the same. This is a very important result since the Synchronous mode is the mode chosen by CMS for detector operations at HL-LHC.

To understand the origin of comparator threshold oscillation, different measurements have been performed. The results of my thesis show that the comparator threshold oscillations are affected by the clock activity in the chip. However, the origin of this unexpected behavior has not yet been found, and further investigations are needed.

BIBLIOGRAPHY

[1] O. Brüning, H. Burkhardt, and S. Myers. The large hadron collider. *Progress in Particle and Nuclear Physics*, 67(3):705–734, 2012.

[2] ATLAS Collaboration, G Aad, E Abat, J Abdallah, AA Abdelalim, A Abdesselam, O Abdinov, BA Abi, M Abolins, H Abramowicz, et al. The ATLAS experiment at the CERN large hadron collider, 2008.

[3] CMS Collaboration. The CMS experiment at the CERN LHC. *Jinst*, 3:S08004, 2008.

[4] CMS Collaboration. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *arXiv preprint arXiv:1207.7235*, 2012.

[5] A Augusto Alves Jr, LM Andrade Filho, AF Barbosa, I Bediaga, G Cernicchiaro, G Guerrer, HP Lima Jr, AA Machado, J Magnin, F Marujo, et al. The LHCb detector at the LHC. *Journal of instrumentation*, 3(08):S08005, 2008.

[6] Kenneth Aamodt, A Abrahantes Quintana, R Achenbach, S Acounis, D Adamová, C Adler, M Aggarwal, F Agnese, G Aglieri Rinella, Z Ahammed, et al. The ALICE experiment at the CERN LHC. *Journal of Instrumentation*, 3(08):S08002, 2008.

[7] CMS luminosity - public results. `https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults`.

[8] G.Ruggiero. Detectors for tracking. *Lecture Notes, University of Florence*, 2021.

[9] G.Pasquali. Rumore elettrico e misure di energia. *Lecture Notes, University of Florence*, 20 May 2021.

[10] Andrea Benaglia. The CMS ECAL performance with examples. *Journal of Instrumentation*, 9(02):C02008, 2014.

[11] CMS technical design report for the phase 1 upgrade of the hadron calorimeter. Technical report, CMS-TDR-010, 2012.

[12] CMS Collaboration. The TriDAS project. technical design report, vol. 1: The trigger systems. *Cited on*, page 81, 2000.

[13] Wolfgang Adam, T Bergauer, C Deldicque, J Ero, R Fruehwirth, M Jeitler, K Kastner, S Kostner, N Neumeister, M Padrta, et al. The CMS high level trigger. *European Physical Journal C*, 46(3), 2006.

[14] LHC long term schedule. http://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm.

[15] The Phase-II Upgrade of the CMS Endcap Calorimeter. Technical report, CERN, Geneva, 2017.

[16] Joel N Butler and Tommaso Tabarelli de Fatis. A MIP timing detector for the CMS phase-II upgrade. Technical report, Fermi National Accelerator Lab.(FNAL), Batavia, IL (United States), 1900.

[17] CMS collaboration. The phase-2 upgrade of the CMS L1 trigger interim technical design report. Technical report, 2017.

[18] A Dominguez. CMS technical design report for the pixel detector upgrade. Technical report, Fermi National Accelerator Lab.(FNAL), Batavia, IL (United States), 2012.

[19] Alessandro La Rosa. The CMS outer tracker for the high luminosity LHC upgrade. *Journal of Instrumentation*, 15(02):C02029, 2020.

[20] CMS Collaborations. Report on the physics at the HL-LHC and perspectives for the HE-LHC. *arXiv preprint arXiv:1902.10229*, 2019.

[21] Wolfgang Adam, T Bergauer, D Blöch, M Dragicevic, R Frühwirth, V Hinger, H Steininger, W Beaumont, D Di Croce, X Janssen, et al. Comparative evaluation of analogue front-end designs for the CMS inner tracker at the high luminosity LHC. *Journal of Instrumentation*, 16(12):P12014, 2021.

[22] Jaya John John on behalf of the RD53 Collaboration. RD53 pixel chips for ATLAS and CMS phase-II upgrades, 25 October 2022.

[23] RD53 Collaboration. *The RD53B-CMS Pixel Readout Chip Manual*. CERN-RD53-PUB, 2021.

[24] Luigi Gaioni and Flavio Loddo. CMS analog front-end: simulations and measurements. Technical report, 2020.

[25] Fondazione bruno kessler. https://www.fbk.eu/it/.

[26] Hamamatsu photonics k.k. https://www.hamamatsu.com/jp/en.html.

[27] Sarah Seif El Nasr-Storey. Phase-II DAQ training workshop preparing CBC for data taking. *Unpublished workshop slides*.

[28] Ilka Antcheva, Maarten Ballintijn, Bertrand Bellenot, Marek Biskup, Rene Brun, Nenad Buncic, Ph Canal, Diego Casadei, Olivier Couet, Valery Fine,

et al. ROOT—A C++ framework for petabyte data storage, statistical analysis and visualization. *Computer Physics Communications*, 182(6):1384–1385, 2011.

[29] Mauro Dinardo on behalf the DAQ team. DAC for ROC. *Inner Tracker Electronics System Workshop*, 6 March 2023.

ACKNOWLEDGEMENTS

I am deeply grateful to Stefano Mersi, for guiding me through my first experience in a large research group such as the CMS group. His knowledge and the passion that he shows for his work always inspired and helped me to improve as a physicist and as a researcher. I have to thank Stefano also for his patience and for trusting me since the very beginning.

I would also like to thank all the CMS group members that helped me to interpret the results of my measurements. In particular, thanks to Antonio Cassese, Stella Orfanelli, Alkis Papanastassiou and Luigi Gaioni.

Moreover, I would be remiss in not mentioning my supervisor, Giacomo Sguazzoni, for introducing me at CERN for the first time and for helping me enormously in writing this thesis. If it is a success, it is all thanks to his advices and support.

Finally, I would like to extend my sincere gratitude to my family, for their never-ending support, and, in particular, to my sister. She does not know, but she had always been my inspiration.