

Analysis of threshold oscillations observed in CMS Phase 2 Inner Tracker modules^(*)

L. DAMENTI⁽¹⁾⁽²⁾ for the TRACKER GROUP of the CMS COLLABORATION

⁽¹⁾ *INFN, Sezione di Firenze - Firenze, Italy*

⁽²⁾ *Dipartimento di Fisica e Astronomia, Università degli Studi di Firenze - Firenze, Italy*

received 2 December 2024

Summary. — During Long Shutdown 3, between 2026 and 2028, the CMS detector will be updated in order to fulfill the requirements of the new LHC era, called High-Luminosity LHC. The updated version of the CMS pixel detector is called Inner Tracker, and it is based on silicon pixel sensor technology. The base unit of the detector is the detector module, which is the assembly of the silicon sensor connected to one or more readout chips and the appropriate printed circuit board for powering and readout. These modules went through an intense R&D and validation campaign. In this study, results obtained with prototype modules are presented. The modules are formed by a sensor with 145 152 pixels bump-bonded to the CMS readout chip. In this study, effects that affect the module threshold, which is a crucial parameter for the detection of a particle signal, have been observed. The deep characterization of this phenomenon allowed to optimize the module calibration procedure through the development of a dedicated software.

1. – The CMS Inner Tracker modules for HL-LHC

When the operation of the accelerator will resume after the Long Shutdown 3 (LS3), at the beginning of 2029 [1], 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 $7.5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ and to collect an integrated luminosity between 3000 fb^{-1} and 4000 fb^{-1} in about ten years of operation. In order to be able to fully exploit the physics potential during the HL-LHC running, the CMS experiment [2], as well as all other LHC experiments, will undergo a major upgrade during LS3, called Phase-2 Upgrade. The CMS tracker, in particular, will be completely replaced. The new system will be divided into the Outer Tracker (OT), based on silicon strip and macro-pixel sensors and the Inner Tracker (IT) based on micro-pixel sensors [3].

The IT consists of ~ 3900 hybrid pixel modules with either 1×2 or 2×2 readout chips (ROCs), for a total of 4.9 m^2 silicon pixel detectors. The total number of readout channels is close to 2 billion. Each module consists of a sensor bump-bonded to the

^(*) IFAE 2024 - “Poster” session

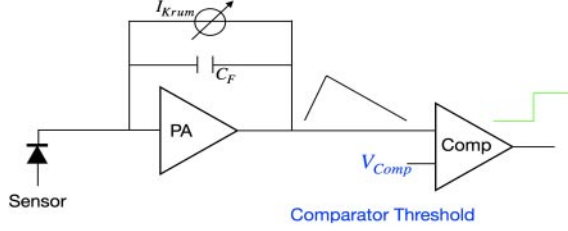


Fig. 1. – Simplified schematic of the CROC linear analog front-end.

ROCs and a high-density interconnect (HDI) printed-circuit board, to route signals and power. The elementary detector cell consists of a single pixel of the sensor attached to the corresponding readout channel of the ROC.

The CMS readout chip (CROC) 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/cm². In fig. 1, a simplified scheme of the CROC linear analog front-end (AFE) is shown.

The charge induced by a traversing particle in a pixel sensor (here represented as a reverse biased diode) is transferred to the CROC and then integrated by a preamplifier (PA) stage. The PA stage of the AFE is formed by a charge sensitive amplifier (CSA) with a Krummenacher feedback. The Krummenacher current (I_{Krum}) generated in the feedback plays a crucial role in the linear AFE as 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 of the preamplifier stage, the signal is characterized by a fast rising edge and a slowly decreasing falling edge. The pulse width is proportional to the deposited charge, while the return to baseline time depends on the Krummenacher current: the higher I_{Krum} , the faster the return to baseline [4]. The amplified signal is then transformed into a digital hit signal by a comparator with a programmable threshold, V_{Comp} . The value of V_{Comp} is determined by a global threshold bias together with a four-bit threshold adjust per channel to compensate for threshold dispersion among the channels. The dynamic range of the threshold adjust is also configurable by a global programmable bias.

A key feature of the IT is the low detection threshold of around 1000 e⁻ made possible by the CROC design and the stringent specifications.

2. – Sampling types and Injection Circuit

In order to understand the results obtained in this work, it is important to describe how the CROC logic identifies a hit, *i.e.*, when the CROC channel connected to a given pixel cell detects a signal. Two different sampling modes can be used: Synchronous and Asynchronous mode. In both sampling modes, a hit is recorded if the comparator output is high at the 40 MHz clock leading edge. This hit is then assigned to the bunch crossing (BX) corresponding to that clock cycle. The main difference between Synchronous and Asynchronous mode is that in the former the comparator output is high as long as the signal exceeds the threshold, while in Asynchronous mode it goes up when the signal exceeds the threshold, and it remains high for 25 ns. Figure 2 illustrates the difference between Synchronous and Asynchronous mode.

The choice of the sampling mode can be made using the DAQ software, called Ph2ACF (for “Phase-2 Acquisition and Control Framework” [5]). Ph2ACF is a system developed by the CMS Tracker Group. Apart from choosing the sampling mode, Ph2ACF also

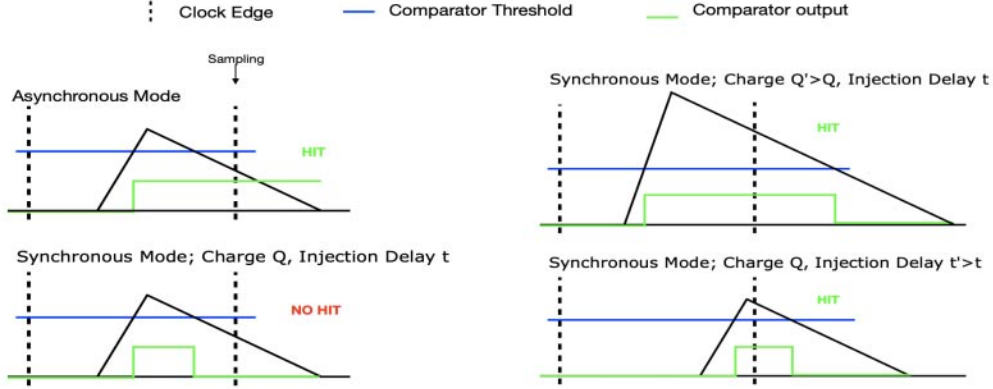


Fig. 2. – Sketches which show the main differences between Synchronous mode and Asynchronous mode.

implements a calibration procedure used to set the threshold of each pixel to a given value, called target threshold, by configuring the working parameters of the ROC. This procedure uses the injection circuit of the ROC to emulate the presence of a particle-induced signal. Two important parameters that need to be set during the threshold tuning procedure are the coarse delay and the fine delay. The former selects the bunch crossing after which the injection will be performed, while the latter determines when the injection is performed within the selected BX, with a granularity of 1 ns. The difference between coarse and fine delay is illustrated in Fig 3.

While studying the front-end performance, it has been observed that the minimum detectable signal in a pixel (called effective threshold) depends on the value of the fine delay that has been set during the threshold tuning procedure. A suboptimal fine delay value set during this procedure can lead to a value of the comparator threshold that is lower than the target threshold.

To visualize this, we can introduce the so-called Tornado Plot. In general, a Tornado Plot shows the results of a 2D scan of Q and Δt , where Q is the injected charge and Δt

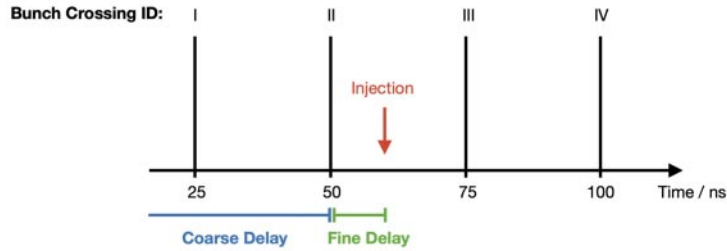


Fig. 3. – Sketch illustrating the difference between coarse and fine delay. In this case, the coarse delay is set to select the bunch crossing with ID = 2, while the fine delay determines when to perform the injection within BX 2.

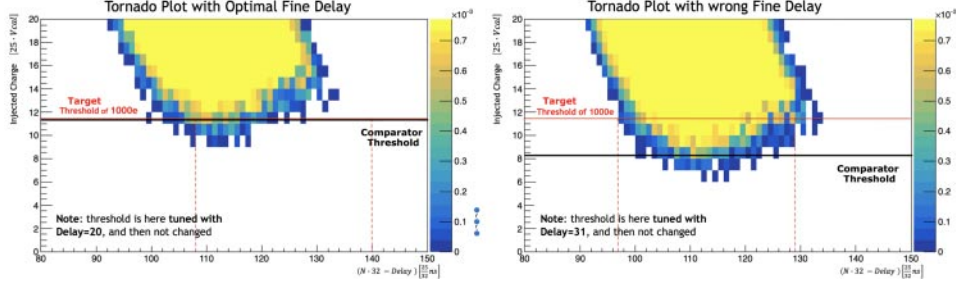


Fig. 4. – Tornado plots of injected charge as a function of the injection delay. Left: with an optimal fine delay, the comparator threshold matches the target. Right: with a suboptimal fine delay value, the comparator threshold is lower than the target threshold.

is the time between the arrival of the signal and the sampling edge, defined as:

$$(1) \quad \Delta t = (32 \times BX_{ID} - \text{fine delay}) \times \left[\frac{25 \text{ ns}}{32} \right].$$

Figure 4 shows that the comparator threshold does not match the target threshold if a suboptimal value of the fine delay is set during the threshold tuning procedure.

According to what was previously reported, it is clear that the choice of the fine delay can affect the noise performance. With a suboptimal fine delay, more pixels are likely to be flagged as noisy as with the lower effective threshold more noise hits will be detected. Figure 5 shows the number of problematic (*i.e.*, noisy) pixels as a function of the target threshold. It can be seen that the noise performance is worse when a suboptimal value of the fine delay is chosen for a fixed target threshold of 1000 electrons. Given the importance of the fine delay, a dedicated procedure has been introduced in Ph2ACF in order to find its optimal value [6].

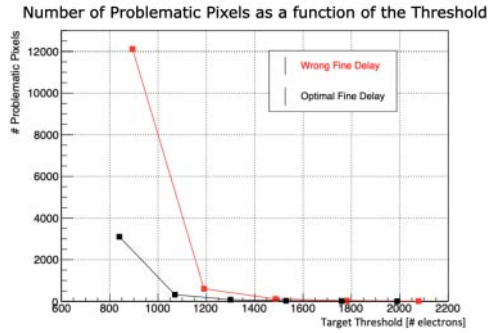


Fig. 5. – Number of problematic pixels as a function of the target threshold. The noise performance at lower threshold gets worse if a suboptimal value of the fine delay is set during the threshold tuning procedure.

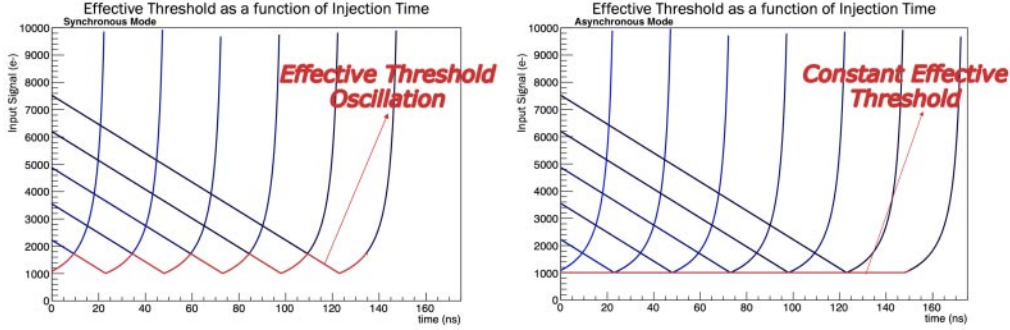


Fig. 6. – Simulated behavior of the effective threshold as a function of the fine delay in the Synchronous (left) and Asynchronous (right) mode. If the comparator threshold is constant, a variation of the effective threshold is visible in the Synchronous mode, while in the Asynchronous mode, the effective threshold is constant as expected.

3. – Threshold oscillation

In the previous paragraph, it was shown that the effective threshold depends on the fine delay value set during the threshold tuning procedure. However, this is true only for the Synchronous mode. In the Asynchronous mode, this kind of dependence is not expected, as the comparator output stays high for 25 ns regardless of how much the signal exceeds the comparator threshold. Nevertheless, a variation of the effective threshold has also been observed in the Asynchronous mode.

In order to better understand this unexpected behavior, a dedicated toy model has been implemented, which simulates the behavior of the effective threshold as a function of the fine delay both in the Synchronous and in the Asynchronous mode.

The left plot of fig. 6 shows the output of the toy model in the Synchronous mode. As expected, a variation of the effective threshold as a function of the fine delay is visible.

Using the toy model, the variation of the oscillation amplitude as a function of the Krummenacher current (I_{Krum}) has also been studied. In particular, for higher I_{Krum} the oscillation amplitude increases. As reported in fig. 7 (right) this behavior is well described by the toy model. It is worth to note that the toy model only has the objective to qualitatively reproduce the behavior of the AFE and hence a simplified signal shape has been used. As a result, the simulated oscillation amplitudes are not comparable with the ones observed experimentally.

On the other hand, if we simulate the behavior of the front-end in the Asynchronous mode, we obtain the right plot of fig. 6. As we previously said, in this sampling mode, we do not expect a variation of the effective threshold with the fine delay. It is worth to note that for both plots of fig. 6 the comparator threshold was set to the same value of $1000 e^-$. The experimental results obtained in the Asynchronous mode can be reproduced using the toy model only by adding an oscillating comparator threshold, as shown in fig. 7 (bottom right). The figure shows also that the dependence of the oscillation amplitude on I_{Krum} is the same as the one observed in the Synchronous mode. This threshold oscillation observed in the Asynchronous mode could be related to spurious effects in the digital part of the chip. However, more studies are needed to identify the origin of this unexpected oscillation.

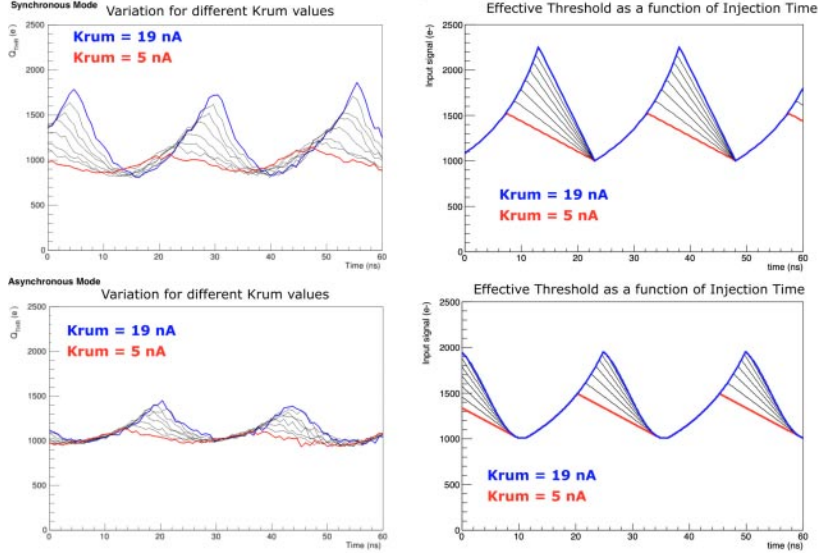


Fig. 7. – Simulated (left) and experimental (right) effective threshold oscillation as a function of the fine delay for Synchronous (top) and Asynchronous (bottom) mode.

4. – Conclusions

Between 2026 and 2028, the CMS detector will be upgraded in order to meet the challenges at the HL-LHC. This work focused on the characterization of the Inner Tracker pixel modules for the Phase-2 of CMS. In the first part of the paper the fine delay parameter, set during the threshold tuning procedure, and its importance for the noise performance of the module has been shown. In addition, an unexpected oscillation of the effective threshold has been observed in the Asynchronous readout mode of the CMS readout chip. This work showed efforts to characterize it and study its possible origin using a dedicated toy model.

REFERENCES

- [1] LHC long term schedule, <http://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm>.
- [2] CMS COLLABORATION, *The CMS experiment at the CERN LHC*, <https://cds.cern.ch/record/1129810/> (2008).
- [3] CMS COLLABORATION, *The Phase-2 Upgrade of the CMS Tracker*, <https://cds.cern.ch/record/2272264> (2017).
- [4] GAIONI LUIGI and LODDO FLAVIO, *CMS analog front-end: Simulations and measurements*, <https://cds.cern.ch/record/2746420/> (2020).
- [5] TRACKER GROUP of the CMS COLLABORATION, *CMS Tracker Phase-2 acquisition & control framework*, <https://gitlab.cern.ch/cms.tk.ph2/Ph2.ACF>.
- [6] DAMENTI LORENZO, *Development of calibration techniques and performance analysis of the CMS Inner Tracker for the High Luminosity phase of LHC*, <https://cds.cern.ch/record/2893555> (2023).