Alma Mater Studiorum · Università di Bologna

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

Preliminary characterisation measurements of CERN picoTDC

Relatore:
Dott. Pietro Antonioli

Correlatore:
Dott. Marco Giacalone

Presentata da:
Giovanni Zanasi

Anno Accademico 2022/2023

Abstract

In questa tesi sono presentate misure preliminari per valutare le prestazioni di un nuovo ASIC TDC sviluppato dal CERN (picoTDC) al fine di valutarne l'idoneità per l'implementazione all'interno del rilevatore TOF di ALICE. La parziale sostituzione dell'attuale HPTDC con il picoTDC comporterebbe, oltre al rinnovo dei componenti delle schede di lettura TDC sviluppate circa 20 anni fa, la semplificazione dell'architettura complessiva, consentendo l'integrazione di 64 canali per chip anziché gli 8 attuali dell'HPTDC. Per raggiungere questo obiettivo, sono stati eseguiti test mirati per valutare la risoluzione, la non linearità differenziale e la capacità di risposta ai segnali forniti dalla NINO FEA, la scheda di front-end del rivelatore TOF, del picoTDC.
Dopo una breve introduzione alle varie tipologie di Analog to Digital Converter (ADC) e Time to Digital Converter (TDC), vengono presentate l'architettura del picoTDC e la configurazione sperimentale utilizzata, seguite dall'illustrazione dell'attività di ricerca condotta in laboratorio. I dati ottenuti hanno evidenziato una buona risoluzione del dispositivo (inferiore a 5 ps) e una notevole compatibilità con la scheda di front-end del rivelatore di ALICE. Tuttavia, i test sulla non linearità differenziale hanno mostrato risultati ancora insoddisfacenti. Pertanto, per il futuro, sarà necessario svolgere ulteriori approfondimenti al fine di migliorare la configurazione del chip.

Contents

Introduction

The aim of this thesis is to conduct a preliminary analysis of the performance of the picoTDC converter in order to assess whether it can serve as a viable alternative to the HPTDC currently used in the ALICE TOF detector. The HPTDC, developed in the early 2000s and installed in the experiment at beginning of LHC operations (2009) served its purpose. However the longer than expected life cycle of the detector (the ALICE TOF is expected now to take data up to 2032) poses several challenges: the TDC cards hosting the HPTDC have now several obsolete components and the few HPTDC spares still available showed connection problems during the soldering process. The prospect of replacing the HPTDC with the picoTDC, which offers 64 channels integration instead of 8 as in the HPTDC, raises the possibility of significantly simplifying the architecture of the current TDC Readout Module (TRM) boards. Currently, to meet system requirements, TRM boards composed of 30 HPTDC units are used, resulting in considerable complexity.
In the introductory first chapter, I recall general elements about analog-to-digital converters (ADCs) and time-to-digital converters (TDCs). Starting with an analysis of ADCs, their key characteristics will be explored. Subsequently, the focus will shift to TDCs, highlighting their fundamental differences and potential applications. Lastly, various errors associated with analog-to-digital converters will be examined.
In the second chapter, the structure and functionalities of the two converters under investigation, namely the HPTDC and the picoTDC, will be described. An analysis of the HPTDC will kick off the chapter, delving into its design features. Subsequently, a more detailed focus will be placed on the picoTDC, scrutinizing how this device has the capability to overcome the limitations of the HPTDC. Finally, a precise description of the experimental setup employed to conduct the performance evaluations of the converters will be provided.
The third and final chapter represents the practical phase of this research, where various tests conducted to assess the performance of the picoTDC will be addressed. The results obtained from these tests will be presented in detail, with identification of several issues to be further investigated.

Chapter 1

Analog to digital converters

For research purposes, it is essential to have a system that converts the signals collected from observing a phenomenon into a format that can be processed by computers. To make this possible, an acquisition chain is required, consisting of various components designed specifically to measure, store, display, and analyze the information provided by the phenomenon under examination. Generally, an acquisition chain can be schematized into five main sectors: the "front-end", where the signal is formed via a specific sensors, signal transport, signal conditioning, digitization, and digital read-out [1]. This chapter aims to analyze the digitization and digital read-out parts, with a particular focus on analog-to-digital converters.
Sensors collect information regarding the phenomenon to be analyzed, converting typically an energy release into an analog electrical signal. This raw signal is then processed through amplifiers and filters to obtain a cleaner and more manageable signal that can now be transformed into a digital format. Analog-to-digital converters are responsible for converting this information so that it can be subsequently analyzed.

1.1 Analog to Digital Converter (ADC)

ADCs are devices that convert an analog input, typically a voltage, into a digital output, which is a binary code [2]. An analog-to-digital converter can be conceptualized as a "divider" - in fact, the output will return the input as a fraction of a reference voltage (V_{REF}): $Output = \frac{2^n \cdot G \cdot A_{IN}}{V_{REF}}$, where n is the number of bits in the output, determining the resolution, G is the gain factor, typically 1, and A_{IN} is the analog input voltage. Alternatively, since the ADC uses input voltages and a voltage reference V_{REF}, it can be considered as a comparator. The number of bits in the output steers the resolution of the converter, so a larger number of bits will result in smaller output steps and higher resolution. To achieve smaller steps, it is also possible to decrease the V_{REF}, but this will result in reduced dynamics and increased exposure to noise. Resolution can also be

defined as the quantity expressed by the Least Significant Bit (LSB)[1].
There are various ways to increase the resolution of an ADC, but each method comes with a cost. Increasing the number of bits raises the component's price, and a very small LSB can be challenging to measure considering the noise. Decreasing the V_{REF} too much results in a loss of dynamics in the input range and potential SNR (Signal-to-Noise Ratio) problems.
ADCs can be differentiated into various types based on the mechanism used to sample and convert signals. In this section, five types of converters will be discussed: Successive-Approximation ADCs (SAR), counter-based ADCs (Voltage-to-Frequency), flash ADC, sigma-delta ADCs (oversampling), and integrating ADCs (dual slope).

1.1.1 Successive-Approximation ADC (SAR)

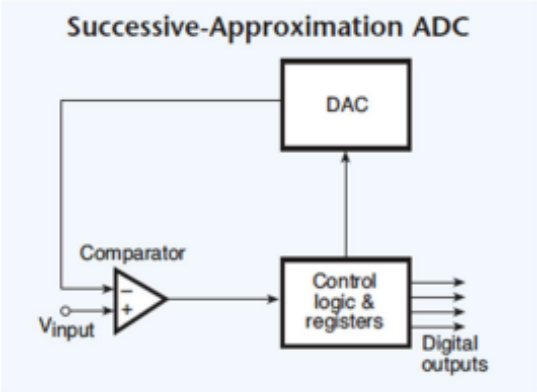

Figure 1.1: This ADC utilizes a DAC and a comparator. It starts with the DAC at zero, incrementing it bit by bit until it matches the input voltage, completing the conversion and storing the result in a register.[3].

SAR ADCs consist of a DAC[2], a comparator, and control logic elements. Once the analog voltage to be measured is present at the input of the comparator, the control logic sets all the bits to zero[3]. Subsequently, the MSB of the DAC is set to 1, forcing the DAC output to be at $\frac{1}{2}$ of the full scale. Now, the comparator compares the DAC output with the analog input: if the voltage of the DAC output is lower than the analog input, the MSB remains set to 1; otherwise, it is cleared. Once this first cycle is completed, the second MSB, with a weight of $\frac{1}{4}$ of the full scale, is set to 1, and the comparison is repeated as before. The process is then repeated until the LSB. At the end of the process, the output will be the digital code representing the input analog signal. This type of ADC is relatively slow (conversion speeds may reach some tens of MS/s (10^6 samples per second) depending on resolution), but it is cost-effective and therefore widely used in many data acquisition systems.

[1]The LSB (Least Significant Bit) and MSB (Most Significant Bit) are the bits that have the least and greatest weight, respectively, in the final value.
[2]Digital to Analog Converter

1.1.2 Counter based ADC

Also known as voltage-to-frequency ADC, this type of converter converts the analog input voltage into a pulse train with a frequency proportional to the input amplitude. The pulses are counted by a counter over a fixed period to determine the input frequency, and the digital voltage is obtained at the output.
These converters inherently have high noise immunity because the input is integrated within the counting interval. They are often used to convert slow and noisy signals and can be used remotely without significant noise issues.

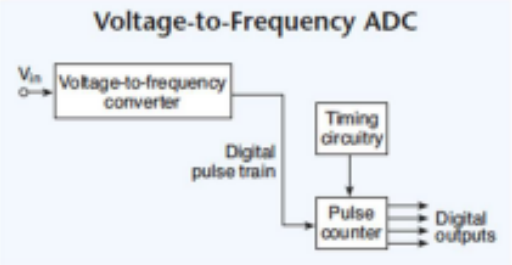

Figure 1.2: Voltage-to-frequency converters reject noise well and frequently are used for measuring slow signals or those in noisy environments[3].

1.1.3 Flash ADC

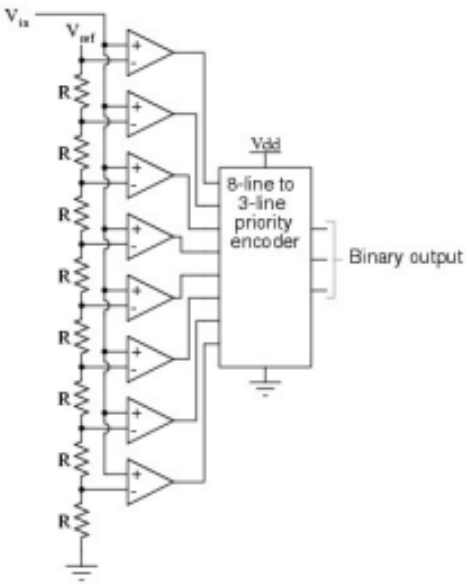

Figure 1.3: In flash ADCs several comparators are used in a parallelized way [2].

The flash ADC is a converter that maximizes conversion speed. The analog input voltage is compared to N different reference voltages. It arrives at parallel-connected comparators and is compared to a decreasing V_{REF} through the use of series-connected resistors. The comparators output binary values are processed through an encoder, which returns the initial voltage in binary format.
The flash ADC has the advantage of being very fast, but it also has several disadvantages. It requires a large number of components, including $2^N - 1$ comparators and 2^N resistors, where N represents the number of output bits. It is therefore difficult to reach high resolution over a large input voltage dynamics. It has high power consumption and high cost. However, it is still widely used due to its fast conversion speed.

1.1.4 Dual slope ADC

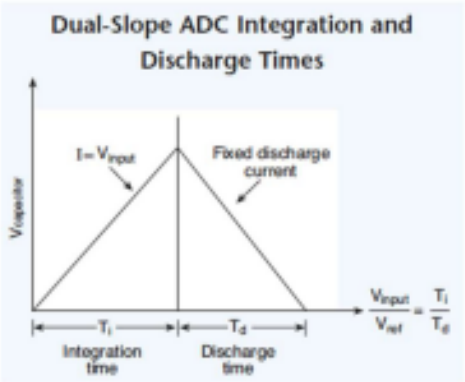

Figure 1.4: Dual-slope ADC principle: the ratio of integration time to discharge time provides a measurement of the input voltage. [3].

These ADCs use integration techniques that measure the time required to charge or discharge a capacitor to determine the input voltage. The system charges the capacitor for a fixed time with a current proportional to the input voltage, and then the time required to discharge it with a constant current determines its value. The greater the discharge time, the higher the V_{input} value. This technique is quite stable as it depends solely on the charging and discharging time and is not affected by factors such as capacitor capacitance, which can vary with temperature. Integrating the signal over a period of time allows for an output signal that is less influenced by noise when the inverse of the integration time is a multiple of the power line frequency.
Due to this quality, it is often used in precision digital multimeters and panel meters. However, it has a significant drawback: while it is common to find it with high precision, its conversion speed is very slow (up to a maximum of 60 Hz for 20 bits).

1.1.5 Sigma-delta ADC

A sigma-delta ADC consists of an integrator, a DAC, a comparator, and a summation junction. The input voltage is algebraically summed with the DAC's output voltage, and the integrator adds a previously stored value to the summing junction. When the integrator's output is greater than or equal to zero, the comparator output goes to logic one, and when it is less than zero, the comparator output goes to logic zero. The DAC modulates the feedback loop, continuously adjusting the comparator output to match the analog input and keep the integrator output at zero (the DAC keeps the integrator output close to the reference voltage level). Through a series of interactions, the output signal becomes a stream of one-bit data that feeds into a digital filter. The filter calculates the average of the series of logical ones and zeros, determines the bandwidth amplitude and transmission speed, and then outputs multi-bit data.
This type of converter, as it uses a one-bit DAC, is relatively inexpensive and can achieve high-resolution measurements using oversampling techniques. It does not require input conditioning circuits or calibration components. It is often used in audio signal digitiza-

tion and, like the previous one, in high-precision laboratory measurement instruments. Sigma-delta ADCs, as mentioned above, employ an oversampling technique to achieve high resolution. The oversampling approach introduces a trade-off with speed and involves converting the input signal at a much higher rate than required to extract fine details and enhance the overall resolution. While this technique provides remarkable precision, it unavoidably leads to a slower conversion rate compared to other ADC architectures. The extensive oversampling demands increased computation time to process the surplus data and perform the necessary noise filtering and averaging. Consequently, sigma-delta ADCs are better suited for applications where high precision is a priority over rapid data acquisition.

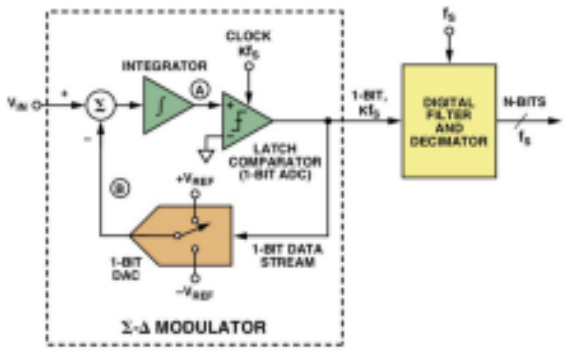

Figure 1.5: The Δ-Σ flow chain consists of a loop with negative feedback, incorporating a 1-bit A/D converter, a 1-bit D/A converter, and a summing node [2].

1.2 TDC and TDC types

TDCs (Time to Digital Converter) are instruments used to measure time intervals between two events with high precision and convert them into digital form. They are often nowadays implemented within ASICs (Application Specific Integrated Circuits).
This type of converter finds applications in various fields, such as the measurement of Extensive Air Shower directions in cosmic ray experiments based on the arrival times at different detectors, biomedical applications for PET (Positron Emission Tomography), background reduction in satellite measurements, particle physics for Time of Flight (TOF) measurements, and more. TDCs differ based on the techniques they use for conversion and the way they are assembled. In this section we will analyze four techniques. Some of them are at the basis of the HPTDC (High-Performance TDC) and the picoTDC which are two ASICs developed at CERN for high precision timing measurements.

1.2.1 Current integration TDC

The current integration TDC, or current-mode, is the most common configuration for TDCs. Similar to the dual slope ADC discussed earlier, this type of converter consists of a capacitor that charges linearly with a constant current. The charging is initiated by the start signal t_1 and stopped by the stop signal t_2. The charge Q will be proportional to the difference $t_1 - t_2$, from which the observed voltage variation can be derived: $V_{cap} = \frac{I(t_1-t_2)}{C}$. However, during the current integration and analog-to-digital conversion, the TDC cannot measure. To reduce the dead time, a flash ADC can be used.
In the figure, two Time to Amplitude Converters (TACs) are shown. TACs are converters used to convert small time intervals into pulse amplitudes.
The parameters that influence the resolution of this TDC include the stability of the current source, the linearity of the capacitor, and the resolution of the TAC. If one wishes to vary the dynamic range, it is possible to adjust either the current (I)

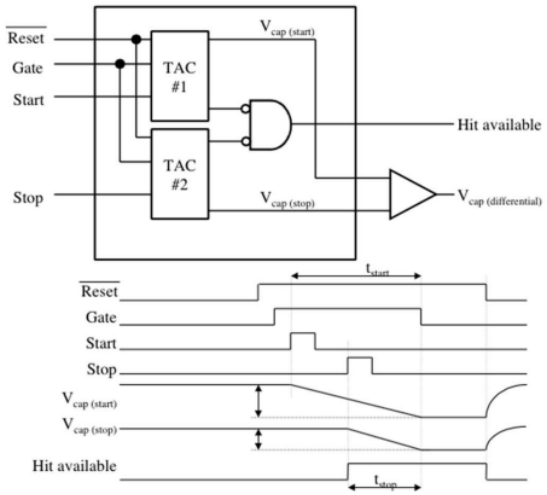

Figure 1.6: Schematic of a TDC based on the current integration technique, which includes two TACs. This TDC measures the time elapsed between an input pulse and a known reference time using two TACs to convert time into signal amplitudes, which are then processed to determine the time measurement. [2]

or the capacitance (C). Furthermore, to reduce sensitivity to noise, differential inputs can be used.

1.2.2 Counter based TDC

Counter based TDCs uses a reference clock oscillator and asynchronous binary counters controlled by a start and stop pulse. The resolution is determined by the frequency of the oscillator, and its quantization error[3] depends on the clock. This type of TDC offers virtually unlimited dynamic range because by adding a single bit, the counter's capacity is doubled at the expense of increased quantization error [4]. The start signal can be synchronous or asynchronous: if the start signal is synchronous with the clock, the quantization error is completely deterministic, whereas if the start and stop signals are asynchronous with respect to the clock, the quantization error can be completely random. The counter method is commonly used as the "coarse" time digitizing method and is employed to expand the measurement range. The resolution of a counter-based TDC is ultimately determined by its clock frequency. Using a primary very high frequency ($\simeq$1 GHz) can be complicated and high consuming.

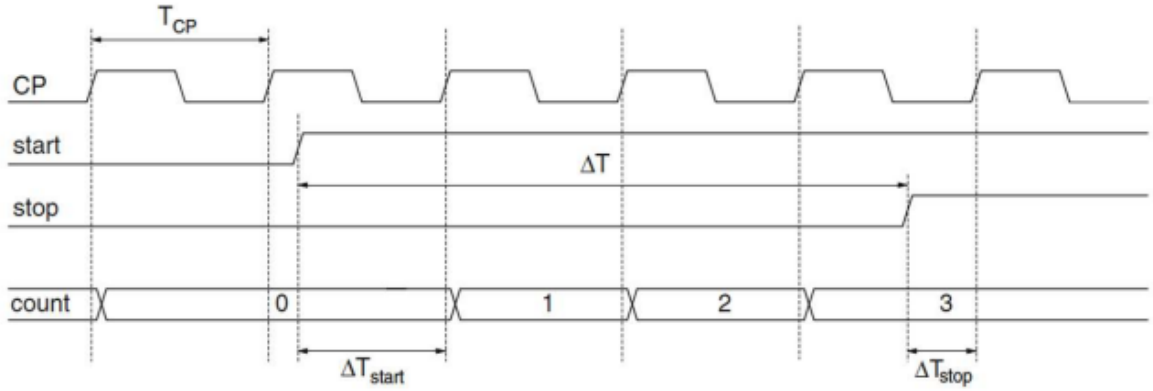

Figure 1.7: Counter base TDC: the interval between start and stop sigmal will be measured counting how many clock periods last between the two signals [2].

1.2.3 TDC based on Delay Locked Loop (DLL)

A DLL-based TDC represents a more sophisticated approach to time measurement. This method leverages a series of delay lines, each comprised of delay elements such as inverters, to record the moment an event occurs, typically referred to as a stop event. This recording is achieved through a snapshot of all delay lines, identifying which delays

[3]In the next section, we will analyze in more detail what quantization error is.

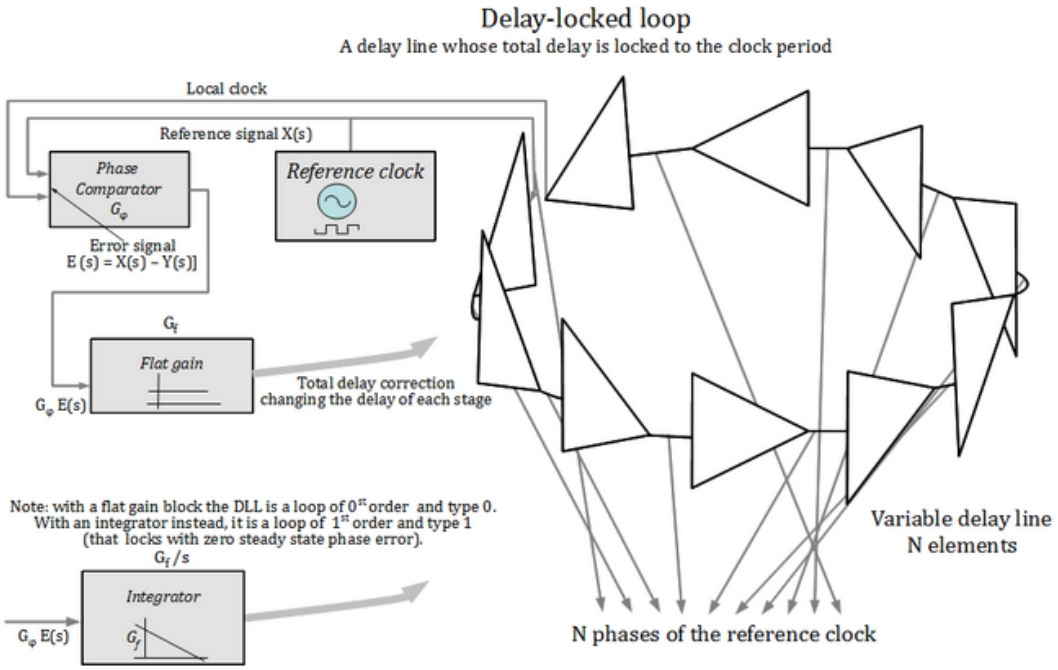

Figure 1.8: The Delay-Locked Loop (DLL) is a variable delay line that synchronizes its delay with the duration of a reference clock period. The type of DLL loop depends on the signal processing element within the loop, which could be a flat amplifier or an integrator [5].

have reached a high state (1), due to signal passing through the delay lines, and which have remained in a low state (0) . This process enables precise measurement of the time elapsed between two successive clock edges. Effectively, it creates a time-to-digital converter based on counters, where the clock frequency is multiplied by the number of elements in the DLL. At the core of a DLL is a delay chain consisting of multiple delay gates, with the input connected to the clock signal requiring adjustment. Each stage of the delay chain incorporates a multiplexer and a control circuit that dynamically adjusts the multiplexer selector to achieve the desired delay. The output of the DLL is the resulting delayed clock signal.
The use of DLLs extends beyond time measurement and finds applications in enhancing timing precision between the rising edge of a clock signal and the output of integrated circuits. Additionally, DLLs can be employed for clock recovery, where they compare the phase of their last output with the input clock signal and generate an error signal. This error signal is then integrated and fed back to control all delay elements, gradually reducing the error while maintaining the desired phase alignment. The integration process ensures accurate operation of the DLL, as the control signal directly influences phase adjustment.

1.2.4 TDC based on Phase Locked Loop

In this TDC configuration, the control system generates the output signal whose phase is correlated with the input signal, and the delay elements are continuously adjusted using a variable voltage to keep these phases aligned. Maintaining the input and output phases in lockstep also implies keeping the input and output frequencies equal. Therefore, in

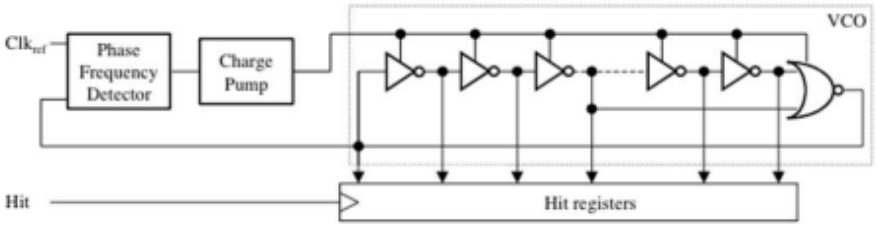

Figure 1.9: PLLs are frequently employed for generating frequencies and clocks in telecommunications and data transmission systems [2].

addition to synchronizing the signals, a phase-locked loop (PLL) can track an input frequency or generate a frequency that is a multiple of the input frequency. The loop provides a self-calibration mechanism.
To distinguish between a DLL and a PLL, it's important to note that a DLL utilizes a variable phase block, whereas a PLL utilizes a variable frequency block. A PLL compares the phase of its oscillator with the input signal to generate an error signal, which is then integrated to create a control signal for the voltage-controlled oscillator (VCO). The control signal influences the frequency of the oscillator, and since phase is the integral of frequency, a second integration is inevitably performed by the oscillator itself. There are TDCs that employ both techniques (PLL and DLL) to achieve better resolution, de-facto multiplying internally the frequency of the primary clock.

1.3 Errors

In this paragraph, we will address some of the most common errors that ADCs and TDCs are subject to in order to understand the testing strategies for these converters.

1.3.1 Quantization error

An ideal ADC assigns a unique digital code to each analog voltage with infinite precision. However, in reality, due to the finite resolution of the ADC, there are finite gaps between consecutive digital values, and the magnitude of these gaps is determined by the least significant bit (LSB). The quantization error is the difference between the analog signal and the closest available digital value at each sampling instant of the A/D converter [1]. To elaborate further, the quantization error arises because the analog signal is approximated by the digital representation. It occurs because the actual analog value falls within a certain range that is represented by a single digital value. The quantization error represents the discrepancy between the true analog value and the quantized digital value.
In order to minimize the impact of quantization error, it is recommended that the quantization error does not exceed half of the LSB. This is known as the half-LSB criterion. By adhering to this criterion, the quantization error remains within a tolerable range and is evenly distributed, both positively and negatively, around the true analog value. By increasing the number of bits in the ADC, the LSB becomes smaller, resulting in finer resolution and reducing the quantization error. However, it's important to consider that increasing the number of bits also comes with increased cost and complexity.
It is useful to explore the relationship between RMS noise and quantization error. In a quantization bin of size q, the probability density function $f(x)$ for the measurement between x and $x + dx$ is flat and equal to $\frac{1}{q}$. In general, $\sigma^2_{RMS} =< x^2 > - < x >^2$. In this case:

$$\sigma^2_{RMS} = \frac{1}{q}\int_0^q x^2 dx + \left[\frac{1}{q}\int_0^q x dx\right]^2 = \frac{1}{q}\cdot\frac{q^3}{3} - \frac{1}{q^2}\left[\frac{q^2}{2}\right]^2$$

This results in the more general expression:

$$\sigma^2_{RMS} = \frac{LSB}{\sqrt{12}}$$

where LSB represents the Least Significant Bit. This expression quantifies the RMS measured values due to the quantization error introduced by the size of the LSB. For TDCs, similar concepts apply, but the errors are related to time measurement rather than voltage or current.

1.3.2 Offset, gain and full-scale error

These three errors are connected among each other and all related to how the converter deviates from its ideal conversion behavour:

- Offset error or zero-scale error: This error refers to the translation or shift of the input signal relative to the ideal signal along the x-axis. It measures the deviation of the input signal from the expected zero value.
- Gain error: refers to a deviation in the slope or gain of the ideal transfer function. It represents a scaling factor mismatch between the input and output signals. When the offset is subtracted from the transfer function, the gain error is equivalent to the full-scale error. By adjusting the gain and offset of a converter channel through hardware or software calibration, it is possible to achieve a specific input-output transfer function.
- Full-scale error or FSE: This error represents the deviation of the actual full-scale value from the ideal full-scale value. It is caused by offset errors or gain errors that differ from the ideal values. FSE affects the measurement accuracy across the entire range of the converter.

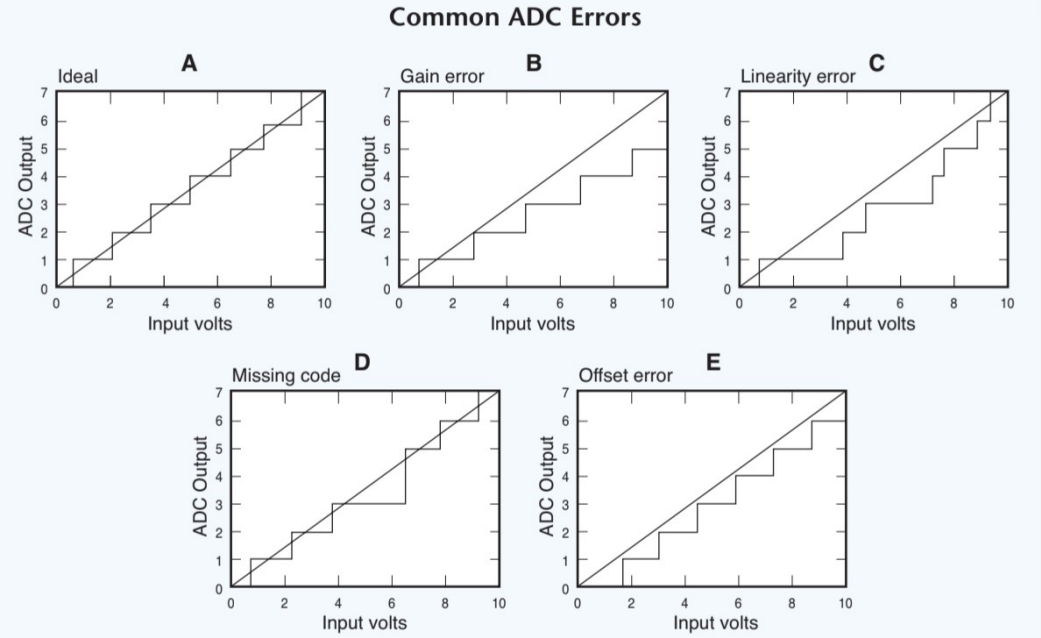

Figure 1.10: In the graphs, the solid line shows analog input voltage and the perfect, infinitely-resolved output. Graph A depicts the ideal response, while B, C, D, and E display the impact of identified errors on ADC output.[3]

Calibration techniques can be employed to mitigate these errors by adjusting the gain and offset of the converter [3]. Typically, the offset is set to zero, and the gain is set to a precise full-scale voltage applied at the input. This ensures that the converter provides the desired input-output relationship.
By addressing these errors through calibration, both hardware and software adjustments can be made to improve the accuracy and performance of the converter.

1.3.3 Linearity errors: INL e DNL

When the input voltage and output readings of a converter deviate from the ideal conversion line (i.e., the ideal input-output relationship) by more than the ideal step function (shown in Figure 1.10-A), the error can be expressed in terms of non linearity errors. There are two distinct types of non-linearity errors: Differential Non-Linearity (DNL) and Integral Non-Linearity (INL).
Differential Non-Linearity (DNL) describes errors associated with step size and is characterized by small-scale errors. The term "differential" refers to the difference between the ideal step and the actual step. DNL reflects how much the input amplitude needs to change for the output code to change by one unit. A positive DNL expands the input range relative to the nominal range, while a negative DNL reduces it.
Integral Non-Linearity (INL), on the other hand, describes the cumulative DNL error along the transfer function. It represents the maximum deviation from the transfer function relative to the straight line connecting two points, without considering quantization, gain, and offset errors.
The non-linearity of a well-calibrated converter produces larger errors near the midpoint of the input range. As a general rule, the non-linearity of a good converter should be equal to or less than one LSB (Least Significant Bit).
To achieve accurate and precise conversion, minimizing non-linearity errors is crucial. These errors can be reduced through careful design, component selection, and calibration techniques. By maintaining DNL and INL within acceptable limits, the converter can provide more reliable and consistent results across its entire operating range.

1.3.4 Missing code

A high-quality converter should generate an accurate output for any input voltage within its resolution range. It means that no consecutive digital code should be skipped. However, some converters may exhibit errors where they are unable to produce an accurate digital output for a given analog input. In Figure 1.10-D, it is shown that a specific 3-bit ADC fails to provide an output representing the number 4 for any input voltage.
When a converter exhibits such errors, it compromises its ability to faithfully represent the input signal in its digital output. This can lead to distortion, loss of information, and reduced precision. The converter's accuracy refers to its ability to provide output

values that closely match the corresponding input values. The resolution, on the other hand, is determined by the number of available digital codes or bits, and it indicates the smallest incremental change that can be detected or represented by the converter.

Chapter 2

The picoTDC and the experimental setup

The Time-Of-Flight (TOF) detector is one of the sub-systems of the ALICE experiment at the LHC. The TOF, via the measurement of the time of flight of particles crossing the MultiGap Resistive Chambers at 3.7 m from the interaction point, is responsible of hadron particle identification for intermediate momenta ($0.5 < \mathrm{p} < 4\ GeV/c$) within ALICE. The TDC used for the time measurement is the HPTDC ASIC developed at CERN for LHC applications in the first decade of this century. A new ASIC has now been developed which offers better resolution and integration (and reduced power consumption).
The HPTDC is a high-performance time-to-digital converter used to accurately measure the time interval between events. In the context of the TOF experiment, the HPTDC has been employed to measure the flight time of particles traversing the detection system. However, due to its limited capacity of only 8 channels, it has been necessary to combine 30 of these components on a single board (TRM: TDC Readout Module) to meet the requirements of the system. The picoTDC represents a step forward compared to the HPTDC in terms of resolution and integration. This device has 64 channels. Its increased capacity reduces the need to combine numerous components like in the TRM, allowing for the creation of simpler boards. This thesis presents first tests done on picoTDC evaluation board to consider the picoTDC as an alternative to the HPTDC for new TRM boards. The primary goal of this phase is to assess the performance of the picoTDC under various test conditions, determining its reliability and precision. In summary, this chapter provides an analysis of the HPTDC's capabilities and a description of the picoTDC as a potential alternative. Additionally, it describes the experimental configuration used to evaluate the performance of the picoTDC.

2.1 HPTDC

The High Performance general-purpose TDC (HPTDC) was a state-of-the-art device built in 0.25 μm CMOS technology for LHC application in the early 2000's developed by the Microelectronics group at CERN. Previous versions of general-purpose TDCs were highly appreciated in the high-energy physics community due to their programmable nature, enabling them to be used in various experiments with diverse system requirements. The TDC's high flexibility allows it to be optimized for specific experimental conditions, which might not be fully known before the experiment commences. The data-driven architecture ensures efficient operation in both triggered and un-triggered applications. Additionally, a time-tag-based trigger matching function allows programmable trigger latency, supporting overlapping triggers and the assignment of hits to multiple events. The HPTDC requires a 40 MHz clock as a time reference, which aligns with the front-end electronics' synchronization to the LHC machine's bunch crossings. A Delay Locked Loop (DLL) with 32 delay elements can provide a resolution of approximately 250 ps RMS using the 40 MHz clock. A Time-Of-Flight detector may demand improved time resolution, attainable through a Phase Locked Loop (PLL) generating a 320 MHz clock with 20 ps RMS resolution.

2.1.1 Architecture

The time reference for TDC measurements is established by a Delay-Locked Loop (DLL) comprising 32 delay elements and a clock-synchronous counter. Both these components are driven by the same clock source. This clock reference can be directly obtained from the TDC chip's clock input or derived from an on-chip Phase-Locked Loop (PLL) operating at frequencies of 40/160/320 MHz [6]. The PLL serves the dual purpose of enhancing time resolution through clock multiplication and minimizing input clock jitter. A hit measurement involves capturing the state of the DLL and the coarse counter when a hit (either leading or trailing edge) is detected. Each channel can store up to 4 measurements in its buffer before being written into a level 1 buffer with a capacity of 256 words, shared among a cluster of 8 channels. These individual channel buffers function as modest derandomizer buffers, paving the way for the amalgamation of hit measurements into the L1 buffer. The measurements saved within the level 1 buffer can be directly transferred to a common read-out FIFO, with a depth of 256 words. Alternatively, a trigger matching function can identify events linked to a trigger, allowing for selective data handling. Trigger details, encompassing a trigger time tag and an event ID, can be temporarily held in a trigger FIFO with a capacity of 16 words. A programmable time window is available for accommodating the time dispersion of hits associated with the same event.

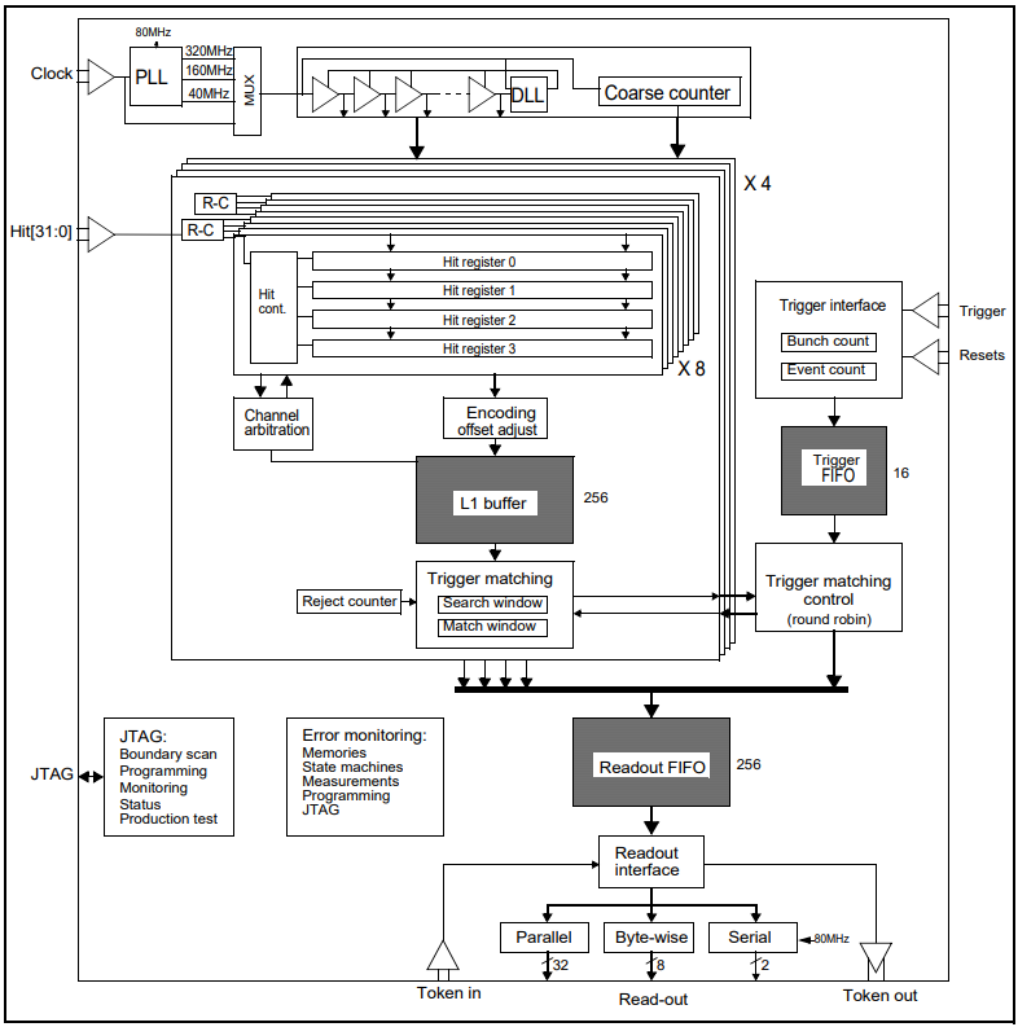

Figure 2.1: Architecture of HPTDC [6]

2.1.2 Phase Locked Loop

The Phase-Locked Loop (PLL) plays a pivotal role in facilitating clock multiplication, elevating the input clock from 40 MHz to 160/320 MHz when high-resolution time measurements are imperative. Alternatively, it can serve to generate a stable clock of the same frequency, proving beneficial when the focus is on eliminating jitter from the input clock.

The core of this system is the Voltage Controlled Oscillator (VCO), responsible for generating a symmetrical clock signal whose frequency and phase are meticulously compared with the reference signal. When the Phase-Frequency Detector (PFD) detects any discrepancy in frequency or phase, it initiates adjustments in the control voltage applied

to the VCO via a charge pump and filter circuit. By subjecting the frequency generated by the VCO to division prior to its comparison with the input reference, the VCO frequency becomes effectively multiplied by the designated factor. The dynamics of the PLL control loop can be tailored through programming, specifically by configuring the currents utilized in the charge pump. It's crucial to initialize the PLL after a stable reference clock has been provided to the TDC chip. Achieving the desired frequency and phase lock typically takes approximately 10 milliseconds. This interplay of components within the PLL guarantees clock multiplication for enhanced precision and effective jitter filtering.

2.1.3 Delay Locked Loop

The Delay-Locked Loop , crucial for generating fundamental timing signals necessary to achieve the desired time resolution, encompasses three principal components:

1 A chain of 32 delay elements, each adjustable through a control voltage, introducing controlled delays.

2 The phase detector measures the phase discrepancy between the primary clock and the clock delayed by the delay chain.

3 The charge pump and filter capacitor jointly generate a control voltage for the delay elements, based on the phase error data from the phase detector.

Following a DLL reset, a certain time is required for it to attain lock. Once successful locking is achieved, the DLL's lock status is established, and the TDC is primed to undertake time measurements. During standard operations, continuous monitoring of the DLL is maintained. The control loop's dynamics within the DLL can be regulated by adjusting the current levels employed in the charge pump circuit. Opting for lower current levels results in a gradual lock tracking process with reduced jitter post-locking. Conversely, higher current levels yield rapid locking but heightened jitter. When employing a Phase-Locked Loop (PLL) preceding the DLL for purposes such as clock multiplication or jitter filtering, it's crucial for the PLL to achieve stable lock before initializing the DLL. Failure to do so could cause the DLL to enter an erroneous locking state. The DLL operates in distinct modes based on the operating frequency (40/160/320 MHz).

2.1.4 Very high resolution mode

Through the execution of multiple samplings of the DLL signals, controlled by a precisely calibrated R-C delay line, a further and final enhancement in time resolution can be achieved. The interpolation within a DLL cell is accomplished by subjecting the DLL signals to four equidistant samplings, spanning the interval of a delay cell. These sampling instances demand utmost precision and stability, necessitating the generation of

sampling signals with minute delays (25 ps). This task is managed by employing four R-C delay lines exhibiting minimal temperature and supply voltage dependencies, although it remains sensitive to processing parameters and requires individual chip calibration. The data from these four channels can serve as individual time measurements, then used for achieving a high-resolution interpolation. These measurements can be independently accessed for debugging and calibration purposes, or alternatively, on-chip interpolation can amalgamate them into a singular, high-resolution time measurement (25 ps LSB). Due to inherent process variations within the R-C delay line, a meticulous calibration of its parameters becomes imperative. This calibration can be efficiently executed via a statistical code density test that relies on a source of random hits. While this calibration procedure could theoretically be entirely integrated onto the chip, it does introduce a level of complexity to the chip's design. Prototypical chips have substantiated that once the appropriate calibration constants are determined, they remain stable during normal operational conditions. Thus, a software-driven, infrequent calibration procedure from external sources is deemed acceptable. Furthermore, intrinsic non-linearities inherent in the DLL can be mitigated by implementing minute, adjustable delays on each signal tap originating from the DLL. In standard applications, these delays can remain initialized at zero. However, for scenarios necessitating stringent integral linearity, these adjustments can effectively counteract disparities within the DLL's delay line. The determination of optimal adjustment parameters can similarly be derived from a straightforward code density test, providing a comprehensive mechanism for enhancing performance.

2.2 PicoTDC

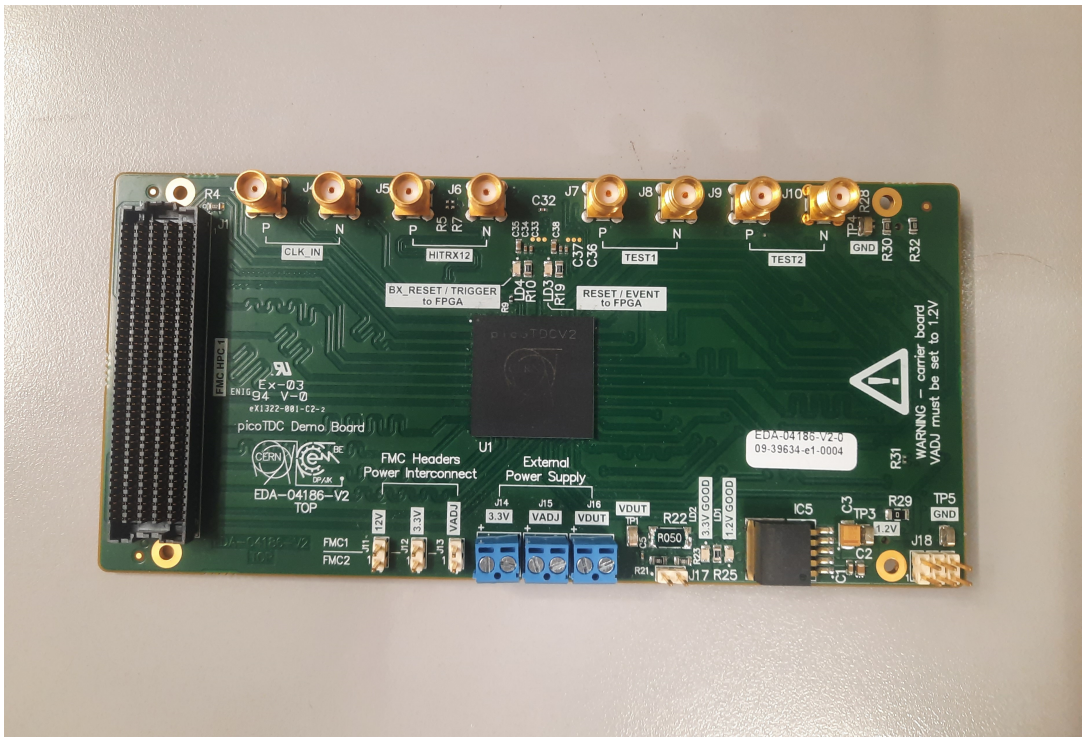

Figure 2.2: "PicoTDC Demo Board by CERN" - A demonstrative board for the picoTDC, provided by CERN.

The term "pico" in picoTDC refers to its expected extremely high time resolution (3.05ps LSB), allowing it to measure time intervals at the picosecond level. TDCs are designed to provide accurate and high-resolution timing information, making them suitable for applications that require precise time measurements. They are used in various scientific, industrial, and research fields, including:

1 Time-of-Flight (ToF) Measurements: In systems that rely on measuring the time taken for signals (such as light or particles) to travel from a source to a detector, picoTDCs can provide accurate ToF measurements.

2 LIDAR (Light Detection and Ranging): LIDAR systems use laser pulses to measure distances and create detailed 3D maps. picoTDCs can accurately measure the time it takes for the laser pulse to reflect back to the sensor, allowing precise distance calculations.

3 Particle Physics Experiments: In particle physics research, picoTDCs can be used to measure particle decay times and other time-related properties in high-energy

physics experiments. In this context, the ToF meeasurement, combined with momentum measurement via a tracking device in a magnetic field, can provide particle identification (knowing the momentum and velocity of a particle is possible to infer its mass)

4 Fluorescence Lifetime Measurements: picoTDCs can be used in fluorescence spectroscopy to measure the lifetimes of fluorescent molecules, which can provide valuable information about molecular interactions and dynamics.

5 Medical Imaging: In medical imaging techniques like Positron Emission Tomography (PET), picoTDCs can be utilized to accurately determine the time of gamma-ray photon emissions, aiding in precise localization and imaging.

The applications of TDCs capable to reach picosecond resolutions are vast, and they play a critical role in fields where precise time measurements are essential for understanding complex phenomena and improving the performance of various systems.

2.2.1 Architecture

The picoTDC measurements, similarly to the HPTDC, rely on a multi-stage approach to enhance temporal resolution. A primary clock operating at 40 MHz, which serves as the basis for the coarse counter implementation with a time resolution of 25 ns [7]. This counter-based TDC is then refined by an integrated Phase-Locked Loop (PLL) operating at 1.28 GHz. The PLL generates also a 320 MHz clock signal for the digital logic. This PLL not only performs clock multiplication from a 40 MHz reference clock but also acts as a filter, mitigating temporal variations (jitter) present in the input clock. Building upon this, the system incorporates a Delay Locked Loop (DLL) with 64 delay elements. This DLL registers the state of the 64 time taps, enabling the system to transition from the initial 781 ps resolution provided by the 1.28 GHz clock to an improved binary size of 12.20 ps. Additionally, a resistive interpolator subdivides the 64 time taps into 256 time taps, further enhancing temporal resolution to a remarkable 3.05 ps[7]. In the measurement process, the hit signal is sampled for each time tap during every clock cycle. Rising or falling edges in this data are detected to pinpoint the precise time value, which is then stored upon hit detection. Consequently, this architecture is capable of detecting a hit per channel for each cycle of the 1.28 GHz clock. Each channel boasts the capability to store up to 4 measurements in a local derandomizer before being written into a 512-word channel buffer operating at 320 MHz. Subsequently, these measurements can be directly transferred to a 512-word readout FIFO memory, shared by a group of 16 channels. Alternatively, a trigger matching function can be employed to select events associated with a trigger. For this purpose, trigger information—including a trigger time tag, event ID, and bunch count ID—can be temporarily stored in separate 512-word trigger FIFOs for each channel group. To account for the temporal spread of

hits related to the same event, a programmable time window of adjustable size is made available for trigger matching. Optionally, the trigger time tag can be subtracted from the measurements, allowing only trigger-relative measurements to be read. Finally, the accepted data can be read through either four separate 8-bit parallel interfaces, each dedicated to a channel group, or from a single 8-bit parallel interface shared by all four channel groups. This flexible data transfer mechanism significantly enhances the overall versatility and performance of the system.

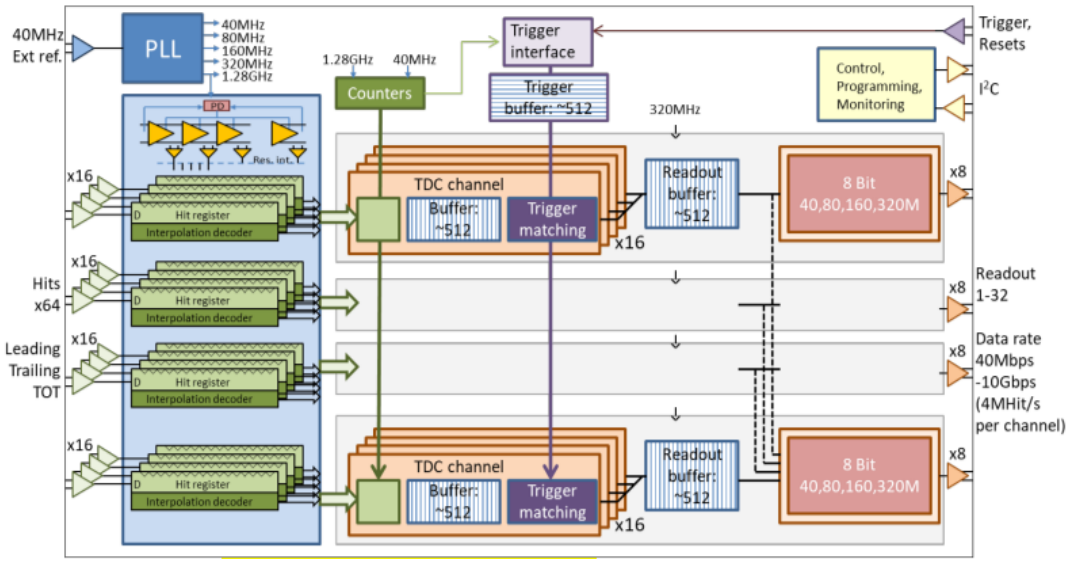

Figure 2.3: Architecture of the picoTDC [7].

2.2.2 Phase Locked Loop

The Phase-Locked Loop (PLL) is responsible for achieving clock multiplication, converting a 40 MHz input clock to 1.28 GHz. It is composed of various components, as shown il Figure 2.4. At the core of the PLL, the Voltage Controlled Oscillator (VCO) generates a symmetrical clock signal, whose frequency and phase are compared to the reference signal. If any frequency or phase mismatch is detected by the Phase-Frequency Detector (PFD), the control voltage to the VCO is adjusted via a charge pump and filter circuit. To accomplish the desired clock multiplication, the frequency generated by the VCO is divided before being compared with the input reference, effectively multiplying the VCO frequency by the specified factor. The dynamics of the PLL control loop can be tailored through programming, involving the adjustment of the currents used in the charge pump and modifications to the filter characteristics.
To ensure proper operation, it is essential to initialize the PLL only after a stable reference clock has been supplied to the TDC chip. The PLL will achieve the correct

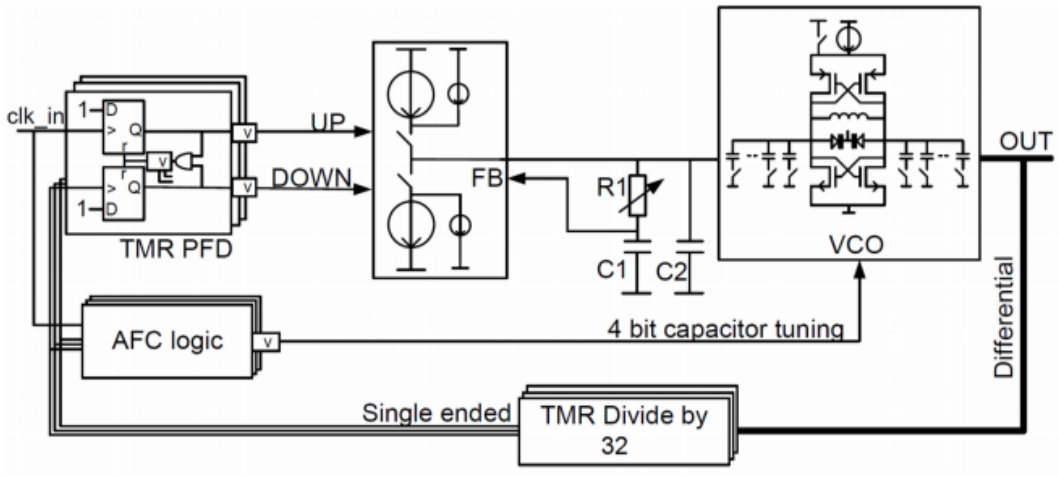

Figure 2.4: Components of PLL used in the picoTDC [7].

frequency and phase lock in approximately 10 milliseconds.

2.2.3 Delay Locked Loop

The Delay-Locked Loop (DLL) plays a crucial role in generating precise timing signals to achieve the required time resolution. It comprises three main components: a chain of 64 delay elements, which can be adjusted by a control voltage, a phase detector that measures the phase error between the clock and the delayed clock from the delay chain, and a charge pump along with a filter capacitor to generate the control voltage for the delay elements based on the phase error detected by the phase detector. Since the DLL is sensitive to jitter on its input clock, any jitter on the clock directly impacts the accuracy of time measurements, potentially deteriorating precision. The dynamics of the control loop in the DLL can be fine-tuned by controlling the current levels in the charge pump circuit. Lower current levels result in slow lock tracing but reduced jitter once locking is achieved. On the other hand, higher current levels facilitate fast locking but may lead to increased jitter.
Before initializing the DLL, the PLL must have achieved stable locking; otherwise, the DLL might enter a false locking state. The correct phase lock will be attained after approximately 10 milliseconds.

2.2.4 The picoTDC data structure of the measurement

The fine time measurement's dynamic range (6 or 8 bits), obtained from the hit signal's state, is expanded by incorporating additional information. This includes the state of the PLL feedback divider (medium time, 5 bits, always in phase with the 40MHz reference)

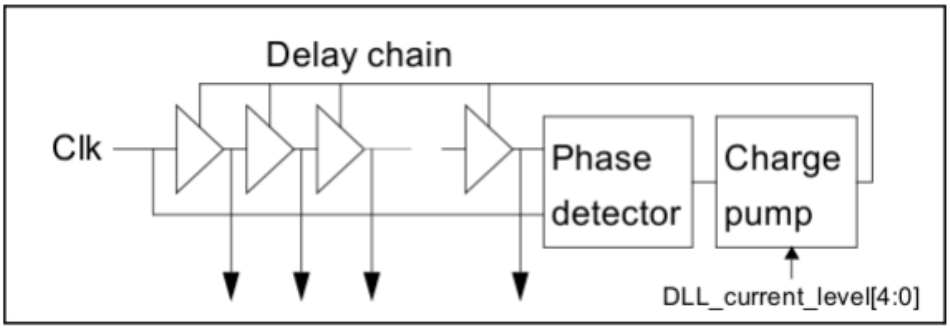

Figure 2.5: "The DLL consists of three key components: a chain of 64 adjustable delay elements, a phase detector measuring phase errors, and a charge pump with a filter capacitor generating control voltages [7].

and a clock-synchronous coarse time counter (running at 40MHz) with 13 bits, which activates when a hit is detected. As a result, the dynamic range reaches 24 bits (with 12ps binning) or 26 bits (with 3ps binning), equivalent to a total range of 205 microseconds. Upon reset, the coarse time counter is set to zero and utilizes natural binary roll-over at 213 for accurate measurements.
Due to the incompatibility between the bunch structure of LHC and the natural binary roll-over of the coarse time counter, an extra 13-bit bunch counter is introduced. This additional counter also runs at 40 MHz, synchronized with the reference clock. It can be independently reset using the bunch count reset signal and can be programmed to roll over to zero at a specific value. This approach ensures that the counter remains synchronized with the bunch count at all times. This value can then be associated with events in triggered mode, providing additional information for analysis.

2.2.5 I^2C Configuration and Status Port

The picoTDC is equipped with a bi-directional 2-wire bus and data transmission protocol (I^2C) that supports Fast-Mode Plus (FM+) speeds. The power supply (VDD) for this bus is typically 1.2V, aligned with the chip's power source. With regards to its I^2C communication, the TDC possesses a 7-bit I^2C address, which can be configured using the I^2C ADDR[6:0] pins. If these pins are left unconnected, the internal pull resistors set the default address to 0x63 (0b1100011).
The internal configuration and status registers within the picoTDC are accessed byte-wise within a 16-bit address space. Similar to I^2C memories, when writing to registers, data bytes should be preceded by the high byte and then the low byte of the starting address. Subsequently, an arbitrary number of data bytes can be included. The internal address is incremented for each received byte, allowing for the writing of multiple configuration words in a single command. All registers, except for the DLL taps adjustment, can be

read back non-destructively. To perform a read command, an I^2C write command with only the two address bytes for the start address of the read is required. This is followed by a read command that retrieves an arbitrary number of bytes, with the internal address being incremented for each byte. Four bytes are combined into an internal configuration word, beginning with address 0x0004. When the highest byte of these four is written, a parity bit is set based on the data, and the parity error status can be examined through the status registers. This enables the detection of configuration corruption during the TDC's runtime by assessing the parity error status bits for the complete configuration. Furthermore, starting from address 0x0160, the status registers are accessible in read-only mode. Writing to these addresses does not have any impact [7].

2.2.6 Read-out

The 4 read-out FIFOs have a depth of 512 words each, serving the primary purpose of allowing events to be read out while others are processed in the trigger matching. When a read-out FIFO becomes full, there are two options for handling this situation, controlled by the config bit "disable-ro-reject."

1 Back propagate: In this mode, the trigger matching process is blocked until space becomes available in the read-out FIFO. When space opens up, the channel buffers and the trigger FIFO are compelled to buffer more data. If this situation persists for extended periods, the channel buffers or trigger FIFO may eventually overflow. In this back propagate mode, the TDC effectively has a FIFO buffering capacity of 16x512 + 512 = 8704 measurements per channel group, approximately 35,000 in total.

2 Reject: Alternatively, when the read-out FIFO is full, event data (excluding event headers and trailers) will be rejected immediately. If the read-out FIFOs remain full for an extended duration, the bandwidth might not be fairly shared between the channels anymore.

In certain scenarios, it can be advantageous to start rejecting hits when a large amount of data starts to accumulate in the read-out buffer. This allows the effective size of the read-out FIFO to be "artificially" reduced through programming, providing more flexibility and control over the data handling process.
The picoTDC is structured into four distinct groups, each housing 16 channels. Each group is equipped with an independent 8-bit parallel interface that utilizes differential signaling. Moreover, there's an extra differential pair with a sync signal, enabling data synchronization. The sync signal can be configured to indicate the beginning of a frame or act as a clock at half the data rate.

2.2.7 Data format

The TDC sends out all data in frames of 32 bits, transmitting the highest byte first. The phase of the data concerning the 40MHz reference can be configured, but the first byte of each frame will always be in sync with this clock. The initial bit of each frame indicates whether it contains hit data (0) or belongs to various management frames (1). The type of other frames can be determined immediately after receiving the first byte, as it is defined by the following 3 bits. In cases where there is no data, the parallel port maintains a constant value of 0xD0, which serves as a unique start for management frames (1101) and generates the idle frame 0xD0D0D0D0. The other possible frame types are: first header (1000), second header (1001), trailer (1010), and channel group separator (1111).

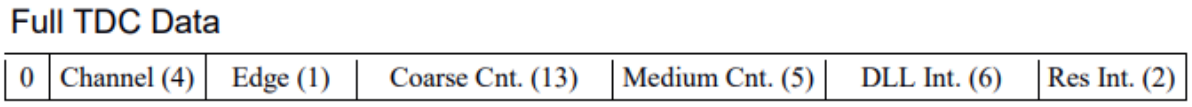

Figure 2.6: Hit data frames commence with the frame type bit (always 0), followed by 4 bits denoting the channel ID, one bit indicating leading edges when high, and ultimately, the full 26-bit time measurement [7].

The hit data frames start with the frame type bit (always 0), followed by 4 bits of channel ID, one bit marking leading edges when high and finally the full time measurement (26 bits).

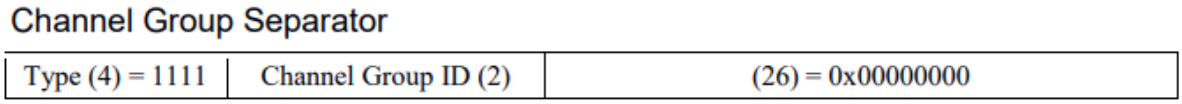

Figure 2.7: To indicate the active channel group transmitting data, a channel group separator frame is inserted before each group switch. This separator is generated only when there is data available within one of the groups [7].

The picoTDC can be configured to transmit all data through readout port 0, which reduces the readout data rate and the number of pins needed to interface with the ASIC. In this mode, the data formatting remains the same as in normal mode. To indicate the channel group currently transmitting data, an additional frame called the channel group separator is inserted before each group switch. This separator is generated only when data is available in one of the groups. The channel groups are served in a round-robin mode to ensure fair sharing of bandwidth. There are two options to determine when the next channel group will be selected:

- Triggered mode: When the full-events config bit is set, a complete trigger event (from headers to trailers) will be read out from each channel group before switching to the next group. This also synchronizes the events between the channel groups.
- Triggered or untriggered mode: In this case, the config word max-grouphits can be set to a value that defines the maximum number of 32-bit data frames a channel group can send before switching occurs. If the currently transmitting group exhausts its data and another group has data to send, switching will also occur under this configuration mode.

2.3 Experimental setup

In this section,i discuss the experimental setup utilized for the tests conducted on the picoTDC. The Figure 2.8 illustrates the nine primary components that form the core of this setup:

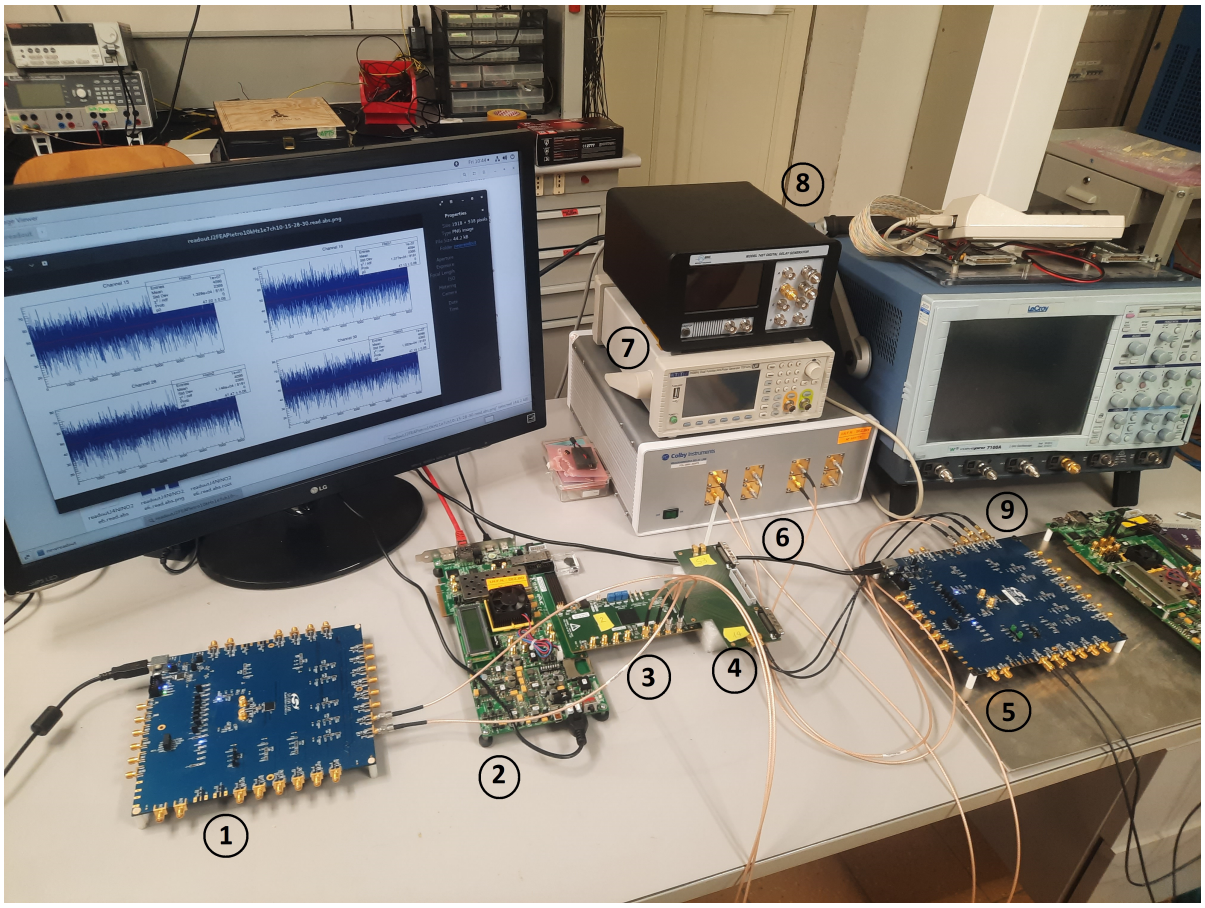

Figure 2.8: A picture of the experimental setup used in the ALICE laboratory (see text for details of the various equipments).

1 SkyWorks board1: This programmable electronic board is employed in this context to supply a clock signal to the picoTDC.

2 FPGA board: this board is responsible for reading the data gathered from the picoTDC.

3 picoTDC: picoTDC demo board provided by CERN.

4 Adapter board: This board facilitates access to the picoTDC channels of the demo board, which uses a FMC connector.

5 SkyWorks board 2: this second card is used for providing input to the picoTDC channel under test.

6 Electromagnetic trombone: High precision delay generator with a range of up to 600ps.

7 Function generator: This component produces specific waveforms or signals.

8 Delay generator: With a capability of inducing delays of several nanoseconds, this generator further increases the possibility to test the TDC under test with calibrated delays. I didn't use this delay generator in my set of measurements..

It's noteworthy that the figure illustrates the experimental setup's configuration when measuring time resolution.

2.3.1 FPGA board

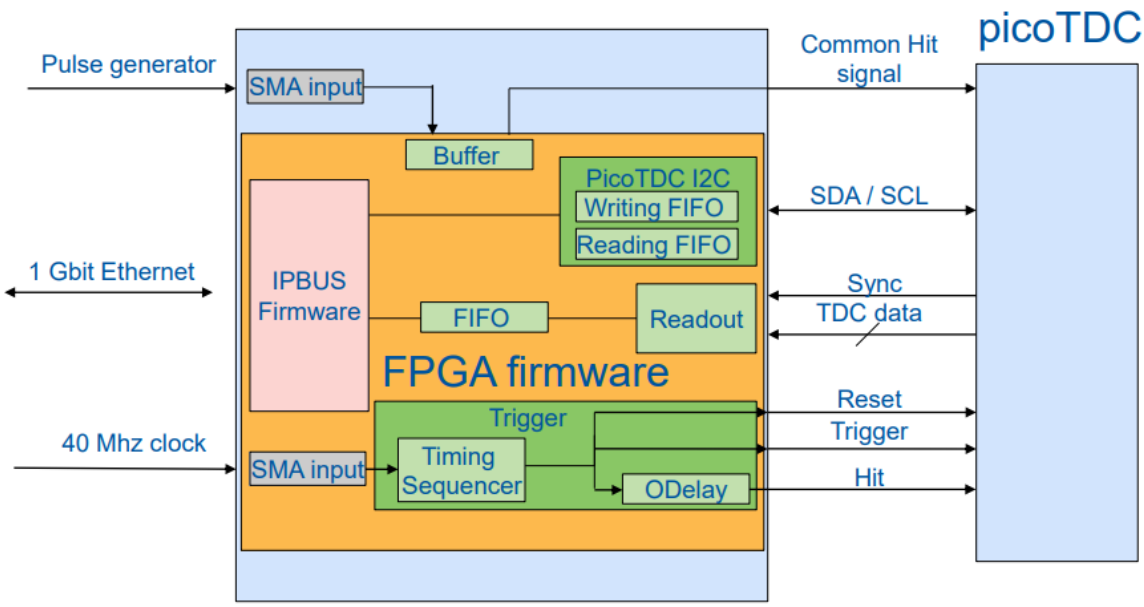

Figure 2.9: Architecture of FPGA board [8].

The Virtex-7 FPGA VC707 is an evaluation board provided by Xilinx having at its core a Virtex FPGA. The acronym "FPGA" stands for Field-Programmable Gate Array, these devices are programmable to execute specific functions, such as combinatorial

and sequential logic, in order to implement custom electronic circuits. It serves as an environment where design ideas can be explored, concepts tested, and the feasibility of hardware solutions verified. This is particularly valuable when experimenting with new techniques or developing prototypes to evaluate the performance of a design before committing to production. The board is suitable for applications requiring data processing from sensors, acquisition devices, or data streams in general. The VC707 evaluation board has multiple interface of communications (PCI express, Ethernet, USB, optical links) with computers. It has also interfaces towards custom hardware via two FMC connectors and several ancillary connectors. In our experimental setup the picoTDC demo board provided by CERN is connected to the FPGA via on one of its two FMC connectors. The I/O pins of the FPGA are connected to the programming interface of the picoTDC (I2C) and to the readout interface. Two of the auxiliary SMA connectors of the FPGA are used to receive the input primary clock (40 MHz) then routed to the picoTDC always via the FMC connectors. Other signals routed to the picoTDC are the ones needed for the trigger and synchronization signals (bunch reset). The interface toward the computer was implemented using the IPBus protocol, exploiting the Ethernet interface of VC707 card.
On the host computer used for programming the picoTDC a Python interface, prepared by CERN,is able to fully configure the picoTDC once the network cable is connected to the FPGA board.

2.3.2 SkyWorks board

Figure 2.10: Si5341-D evaluation board [9].

The Si5341-D-EB is capable of generating clock signals with a wide range of frequencies, ranging from a few kHz to a maximum of 800 MHz. This flexibility makes the oscillator suitable for high-precision measurement instruments. The Si5341-D-EB board offers high temporal stability, with an accuracy on the order of 0.1 parts per million (ppm) [9]. This means that the clock signals generated by this device are extremely accurate and reliable. A notable feature is the low jitter of the generated clock, with typical values below 300 femtoseconds (fs), which means that the generated signals are very stable over time. The Si5341-D-EB allows for multiple clock outputs from a single unit, each of which can be independently configured for fre-

quency and format being the clock jitter the deviation of the clock edge from its ideal location.

2.3.3 Electromagnetic trombone

A trombone delay line is an electronic component used to introduce controlled and variable delays into an electrical signal. This device finds applications in a wide range of scenarios requiring precise management of signal timings. The term "trombone" is employed to evoke the action of extending or shortening the delay, similar to how a musical trombone can be adjusted to vary the length of sound. Trombone delay lines consist of a series of adjustable delay elements arranged sequentially. These delay elements can be realized using diverse technologies such as transmission lines, memory cells, capacitive or inductive circuits, or other components capable of introducing known and controllable delays. The total delay introduced by the trombone depends on the number of activated delay elements. Control over the delay is achieved by activating or deactivating the delay elements within the signal path. The overall delay can be adjusted either manually or electronically through control signals. Some trombone delay lines are equipped with digital control interfaces to allow precise delay adjustments. The XT-200 model from Colby Instruments is a programmable delay line that provides wideband electrical signal delay (phase shift) with excellent resolution from DC to 18 GHz. The delay line combines two electro-mechanical trombones separately and in parallel to offer 0 to 625.0 ps of total delay in each channel of a dual-channel unit. Each channel is independently controlled and can be addressed autonomously [10]. An internal stepper motor produces precise delay steps with a resolution of 0.50 ps when operated in series-connected mode.

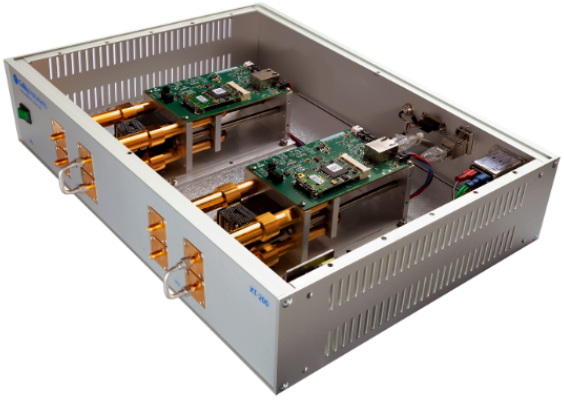

Figure 2.11: Colby Instrument's XT-200 electromagnetic trombone [10]

Chapter 3

Preliminary measurements with picoTDC evaluation board

In this chapter, I present the tests done on the picoTDC demo board provided by CERN, discussing the results obtained.
Separately, I focus on the specific tests performed on the adapter developed by INFN Bologna. All detailed information regarding its usage is presented in the Appendix A.
To gain a comprehensive and accurate understanding of how the picoTDC is configured for data acquisition, we commence by examining the commands used for its initialization.

3.1 Configuration of the picoTDC

To initiate data acquisition, it is necessary to first configure the picoTDC using the software developed by CERN, known as **picotdc_tester_interactive.py** based on Python language. The process begins with the command **tester.quickstart()**, which initializes and configures the picoTDC[11].
Subsequently, a series of commands is used to prepare the picoTDC for data acquisition. Here is an overview of the commands used:

1 **tester.picotdc.channels.disable_all()**: This command disables all channels of the picoTDC, preparing it for a fresh configuration.

2 **tester.picotdc.resolution.set_fine()**: This command sets the picoTDC's resolution to the fine mode. If it were necessary to use the coarse resolution mode, the appropriate command would be **tester.picotdc.resolution.set_coarse()**.

3 **tester.picotdc.readout.reset_FIFO()**: This command resets the FIFO (First In, First Out) buffers, preparing the picoTDC for data acquisition without residual data eventually stored from previous acquisitions.

With these operations completed, the picoTDC is ready for data acquisition.
Now, let's take a closer look at how it is initially configured with default settings. As mentioned earlier, to configure and initialize the picoTDC for data acquisition, it is necessary to use the terminal command **tester.quickstart()** via the software developed by CERN, called **picotdc_tester_interactive.py**. This command automatically sets up the board and the picoTDC for operational conditions. The initialization process is mediated by three functions to ensure that the hardware is configured correctly and ready for testing.
Initially, the **self.initialize()** function is used to program the hardware to a known state. This process begins with "safe mode," which first enables the inputs and outputs of the picoTDC. Subsequently, the channels are enabled, zeroed, and perform basic initialization of power supply using the drivers. Finally, the "power enable" is executed to activate the channels with default settings.
Next, the **self.picotdc.write_default_config()** function is called to configure the channels in the default mode, as defined and tested by CERN.
Finally, the **self.startup()** function performs a more detailed configuration of various parts of the picoTDC. This configuration includes:

1 Default configuration with low-power mode using the "magic word" [source: picoTDC manual] [1].

2 Initialization of the PLL with Automatic Frequency Calibration (AFC).

3 Initialization of the DLL.

4 Activation of all parts of the picoTDC by setting the "magic word" to the correct value.

It's worth noting that during the first phase of this function, all channels are initially disabled, and then only channel 12 is enabled because it is the only channel accessible via the MCx connector without the need of an additional adapter.

3.2 Time Resolution Test for picoTDC Evaluation

An fundamental test to be done to check the accuracy of a TDC is to compare its time interval measurement with respect to a known (and stable) time interval. This can be obtained with a fixed delay line (a cable) or a calibrated source of delay, as it is the trombone. During the test, an electromagnetic delay line was employed to generate extremely precise temporal delays. Figure 2.8 illustrates the experimental setup used to

[1] The "magic word" is used to ensure that the picoTDC is only used when it is configured correctly. During this phase, the "magic word" is initially set to zero, and then, after the correct initialization of everything, it is set to the value 0x5C, which is the correct value to enable the picoTDC.

conduct these tests.
Initially, the picoTDC was connected to the FPGA board, the clock generator (SkyWorks 1 board), and the adapter, allowing access to the device both via the computer and through various necessary connectors, particularly the MCx connectors relevant for this test. After programming the second clock generator, a channel from it was connected to the picoTDC, choosing between 12, 14, or 53, as they are equipped with MCx connectors. Additionally, another channel was routed through the trombone delay line and its output connected to the picoTDC. Once the connection was established, data collection was initiated, recording results on two channel pairs: first with 12 and 14, and then with 14 and 53. The measurement was repeated 20 millions of times. At the end of each acquisition, the delay between the two channels was increased by 100/150 ps, accumulating 4 sets of data for each channel pair. For this test, we configured the second Skyworks board to generate pulses at 23 kHz using LVDS signals with jitter on the order of 100 fs. The data collected during this process were processed using dedicated software that I developed. Initially, the software organized the data so that each subsequent pair represented data from one channel and data from the other. Subsequently, the software calculated the difference between these two points for each pair of measurements. These differences were then used to generate histograms. If the temporal resolution test was conducted correctly, the resulting histogram should follow a Gaussian distribution. Using the standard deviation of this distribution, multiplied by the nominal resolution of the device (3.05ps), we were able to calculate the effective resolution of the picoTDC. Here are now shown in the figure the histograms related to the analysis performed, 4 histograms per channel pair.

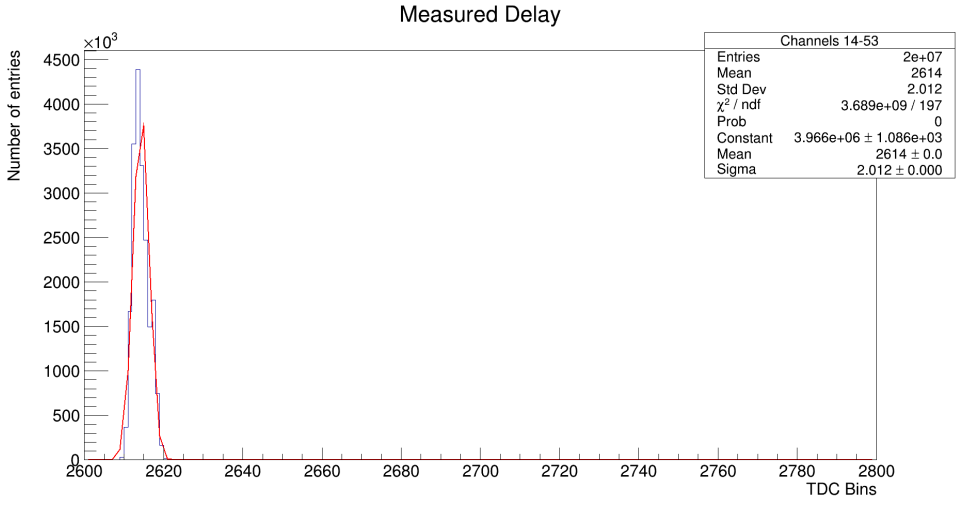

Figure 3.1: Time resolution test: histogram of channels 14 and 53 with Zero delay

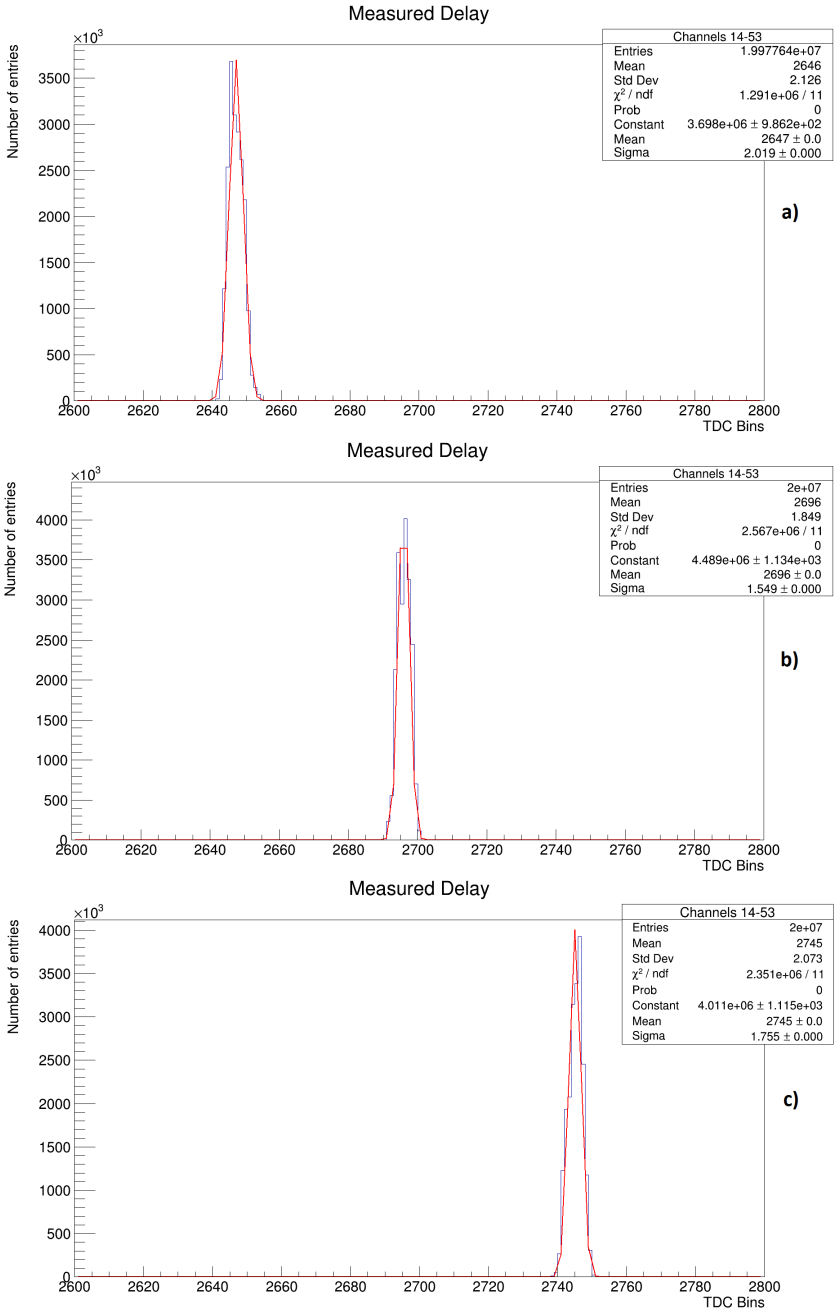

Figure 3.2: These three figures illustrate histograms of channels 14 and 53 from time resolution tests with delay settings of 100ps (figure a), 250ps (figure b), and 400ps (figure c).

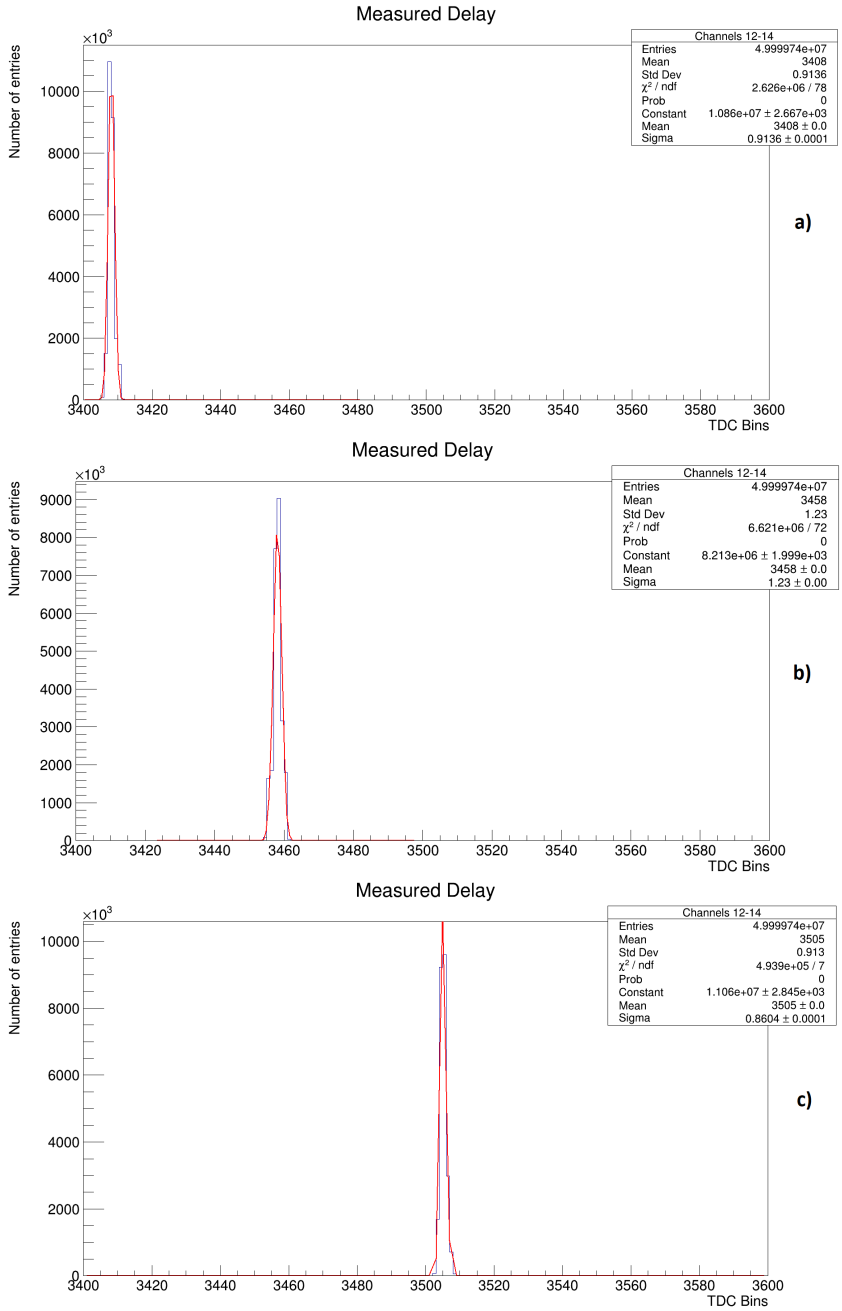

Figure 3.3: These three figures show histograms of channels 12 and 14 obtained through time resolution tests. In graphs a, b, and c, delay settings of 0 ps, 150 ps, and 300 ps are applied, respectively.

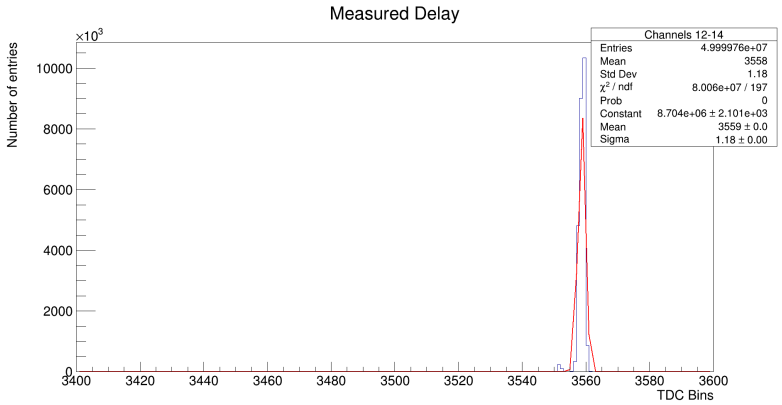

Figure 3.4: Histogram of channels 12 and 14 with 450ps delay

The analysis yielded two tables that allowed us to compare the set values with the measured ones. The data showed a remarkable agreement between the settings and the obtained results. To visualize this relationship more clearly, I show in Fig. 3.5 a scatter plot of the delays set on the delay line on the x-axis and the values derived from the analysis on the y-axis. Their relationship is very close to a line that almost perfectly bisects the quadrant. Furthermore, it's worth noting that the error on the mean value, estimated as the sigma of the value divided by the square root of the number of entries, is negligible.

Delay [ps]	Mean(TDC chan)	Sigma	Delay(TDC chan)	Delay (ps)	σ (ps)
0	2614	2,012	0	0	6,14
100	2647	2,019	33	100,65	6,16
250	2696	1,549	82	250,1	4,72
400	2745	1,755	131	399,55	5,35

Table 3.1: Here are the data obtained from the picoTDC for channels 14 and 53, corresponding to the settings applied on the trombone.

Delay [ps]	Mean (TDC chan)	Sigma	Delay (TDC chan)	Delay (ps)	σ (ps)
0	3408	0,914	0	0	2,79
150	3458	1,230	50	152,5	3,75
300	3505	0,860	97	295,85	2,624
450	3559	1,180	151	460,55	3,60

Table 3.2: The same type of data is presented here, but from channels 12 and 14.

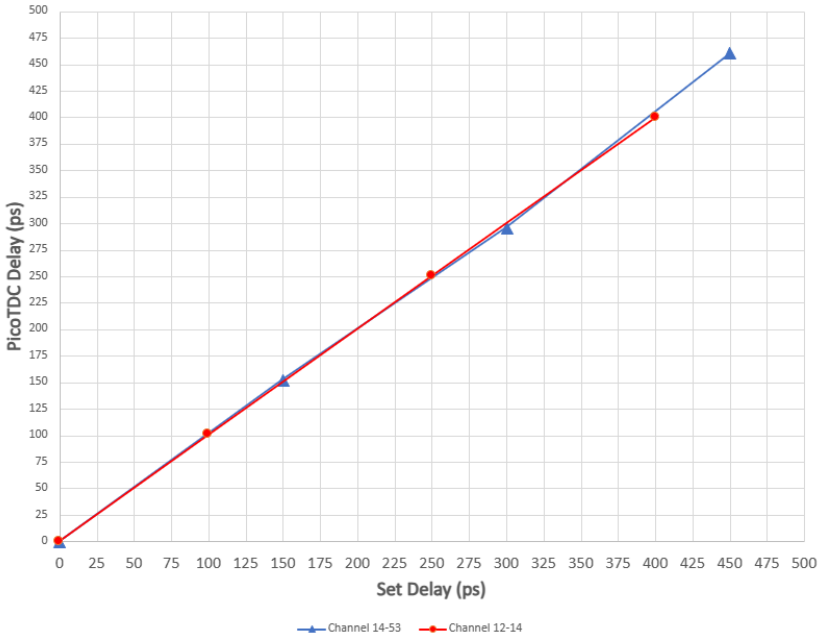

Figure 3.5: Scatter Plot: An illustration of the relationship between delay settings on the x-axis and analysis values on the y-axis.

In both cases (Tables 3.1 and 3.2) the width of the distribution (i.e. the dispersion of the values) is between 0.8 and 2 LSB. A perfect TDC would have LSB/$\sqrt{12}$ that is 0.29 LSB. The fact that the dispersion of the values is larger is per se not a surprise and a contribution to this dispersion is given by the trombone delay line, but we expect this to be negligible (below 0.5 ps) with respect to the values measured. In addition this is the resolution resulting from the pair of two channels, and the pair 14-53 seems worst than pair 12-14. On the single channel, it can be derived however a resolution of sigma/$\sqrt{2}$. Given a LSB of 3.05 ps we were expecting, however, values around that LSB or less. This happens only for two cases over 8 measurements. On the other hand given we are measuring here at sub-10 ps level, we are aware minimal noise disturbances may impact the measurement.

3.3 Evaluation of non linearity of picoTDC

Given the results obtained with the time resolution we further investigated if there are non linearities in the TDC response. This type of test aims to assess the behavior of a device when subjected to variations in input conditions or operational parameters.
To conduct the picoTDC's non-linearity test, we employed the method of code density testing. When pulsed with random hits with respect to its clock we check how they are

distributed over its time acquisition range (25 ns, not considering the coarse counter). To say, if the hits are random and we have 8192 bins of time window and if we send 81920 random hits, we should have on average 10 entries per bins. This method naturally works for a counter based TDC as the picoTDC is. In turn, the time resolution of the TDC will be affected by differential non linearity of the TDC reported by this test.
Initially, we focused on channels 12, 14, and 53, as these channels are accessible through MCx connectors, and furthermore, channel 12 is directly accessible on the demo board (the other two are on the adapter board). Subsequently, we extended the test to encompass all channels of the picoTDC using VHDCI connectors, enabling us to examine the device's response when multiple channels are simultaneously stimulated. It should be noted that during this phase, it was necessary to carefully test the adapter board in order to have a clear understanding of which channels were functioning correctly (see Appendix A). The data collected during the test were analyzed using a specially designed macro within the Root environment. This macro facilitated the processing of data and the generation of specific graphs illustrating the differential non-linearity for each analyzed channel. These graphs highlight the relationship between input conditions and the readings obtained from the picoTDC, thereby identifying any non-linearities present in the channels. The results obtained from this test provide a first evaluation of the picoTDC's non-linearity performance. This information is essential for identifying areas where improvements may be needed. In the conducted analysis, the channel plots of the picoTDC device were initially examined to verify the uniform distribution of bins. In this initial phase, a general overview of event distribution within the bins was obtained. Subsequently, the following formula was applied:

$$DNL = \frac{N_{ev} - \frac{N_{TOT}}{8192}}{\frac{N_{TOT}}{8192}} = \frac{N_{ev}}{\frac{N_{TOT}}{8192}} - 1$$

where, N_{ev} represents the number of events recorded in each bin, N_{TOT} is the total events acquired during the test, and 8192 is the number of considered bins. This formula allowed for the generation of differential non-linearity histograms as shown in Figure 3.6. It is relevant to note that DNL represents a fraction of the Least Significant Bit (LSB), implying that it reflects the device's capability to accurately assign the allocated nominal time of 3.05 ps for each bin. Using the definition given in the formula above, a DNL value less than 0 means that specific bin happens less frequently than expected, and it has an effective duration of less than 3.05 ps. The opposite if DNL greater than 0.
The code density test was separately analyzed on the five bits of the PLL (Figure 3.8), the six bits of the DLL(Figure 3.7), and the last two interpolation bits(Figure 3.9). This in-depth assessment allowed for evaluating the correct calibration of each component in the data conversion process and identifying variations or potential issues in these parts of the device. During the analysis, an interesting pattern emerged: the first bin of each DLL graph for every channel exhibited a low or non-existent event frequency.

This suggests that the initialization of the DLL require further evaluation, as one would expect a similar behavior in each bin without privileges for some. Indeed, in the plots of Fig 3.6, a clear pattern of 32 peaks is visible, a number that, in response to the suspicion, strongly points to the DLL since it has 32 bins. The result shown in Fig. 3.6 is not satisfactory. It can be seen clearly that there is a periodic pattern of non linearity with some bins having a DNL even reaching a value of of three (so a single bin of 3.05 ps, covers instead a much larger time interval up to 12 ps) and other suppressed (well below 0).
The analysis was carried out using a large dataset, comprising 8,192,000 events per channel across a total of 18 channels.

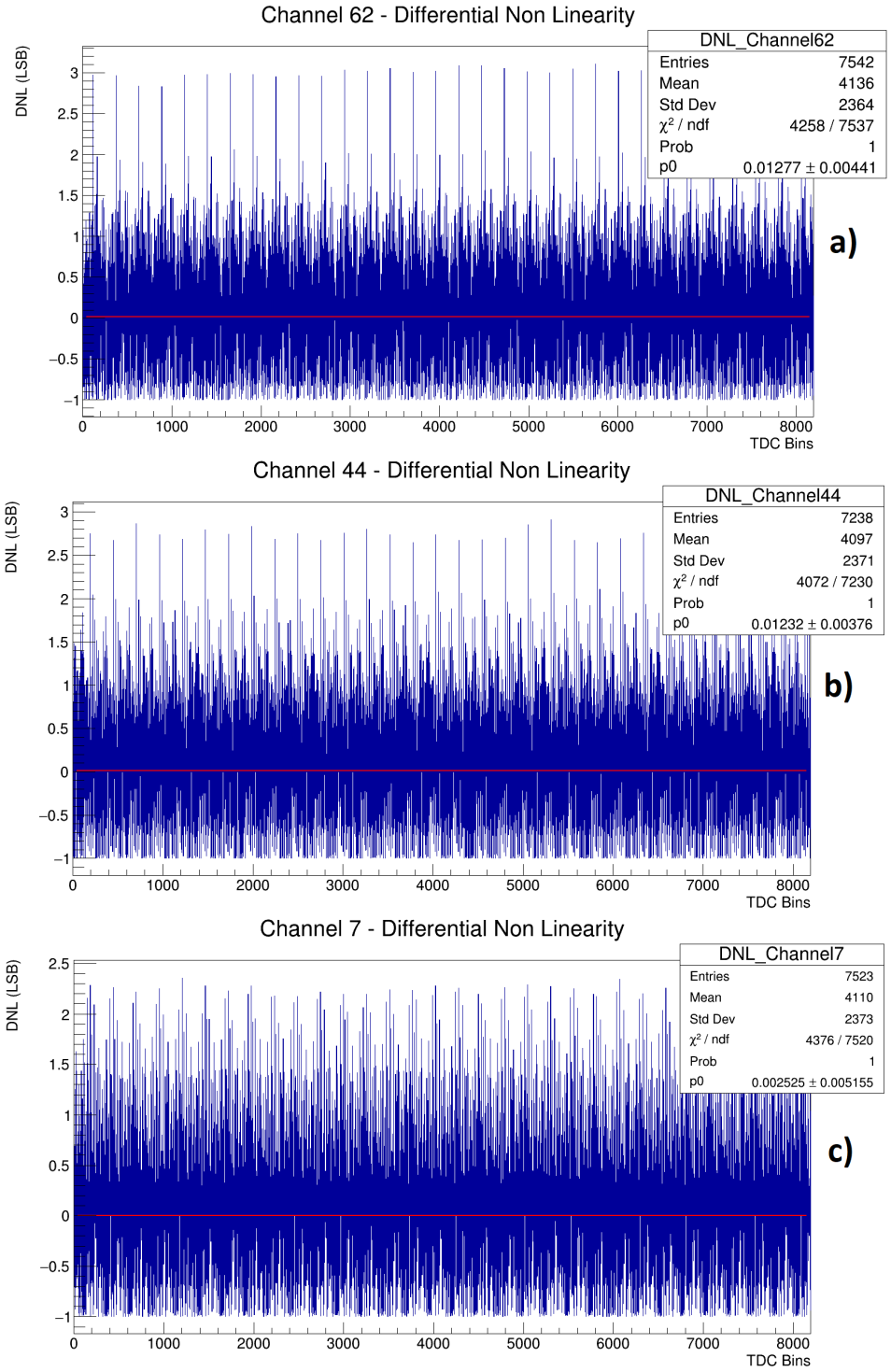

Figure 3.6: Three histograms illustrating the differential non-linearity across three channels. A distinctive pattern of 32 peaks attributed to the DLL is clearly visible.

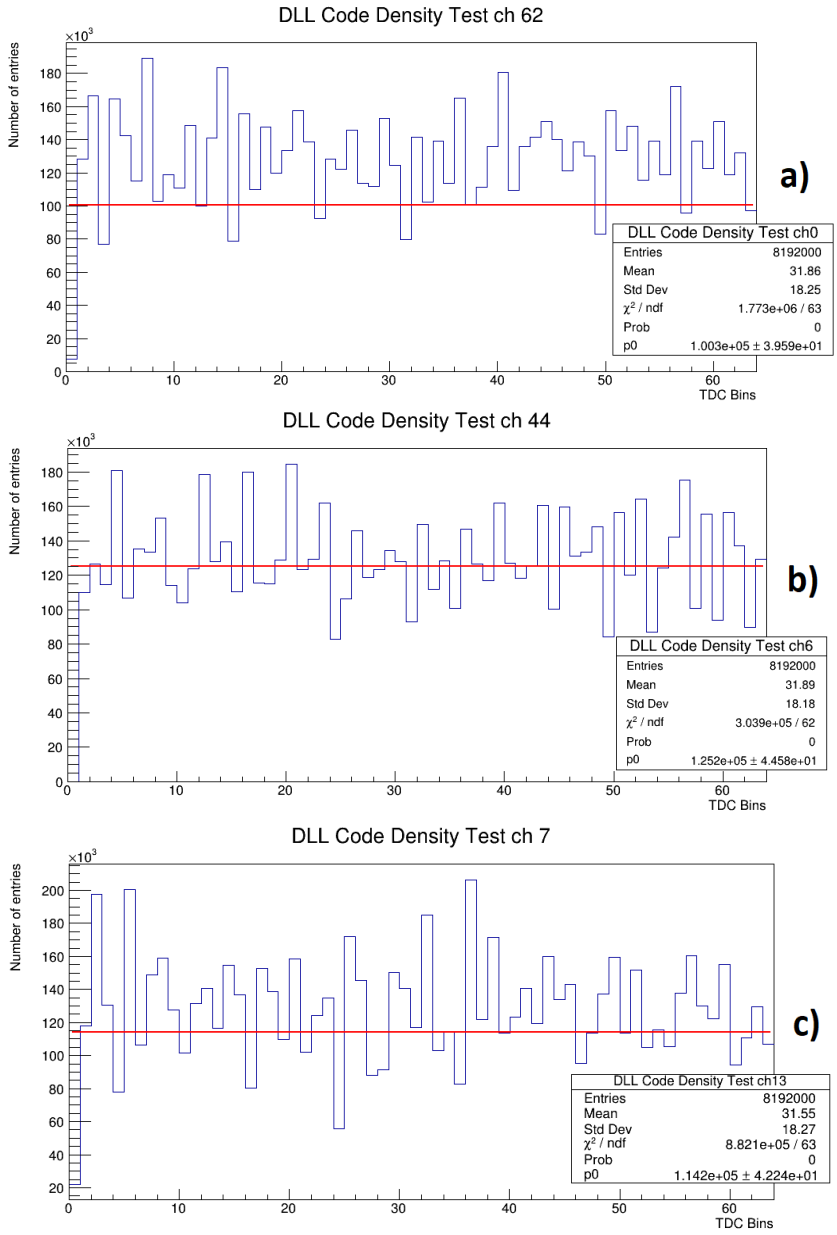

Figure 3.7: These three histograms illustrate the code density test for the 6 bits (64 bins) of the DLL across 3 channels. It is noticeable that the first bin of each channel is either zero or very close to zero. This suggests, given its near absence, that the modulations in the DNL histogram are likely due to this bit. The expected number of entries per bin is 128000 and it can be seen that several bins have a much larger of entries (up to 200000).

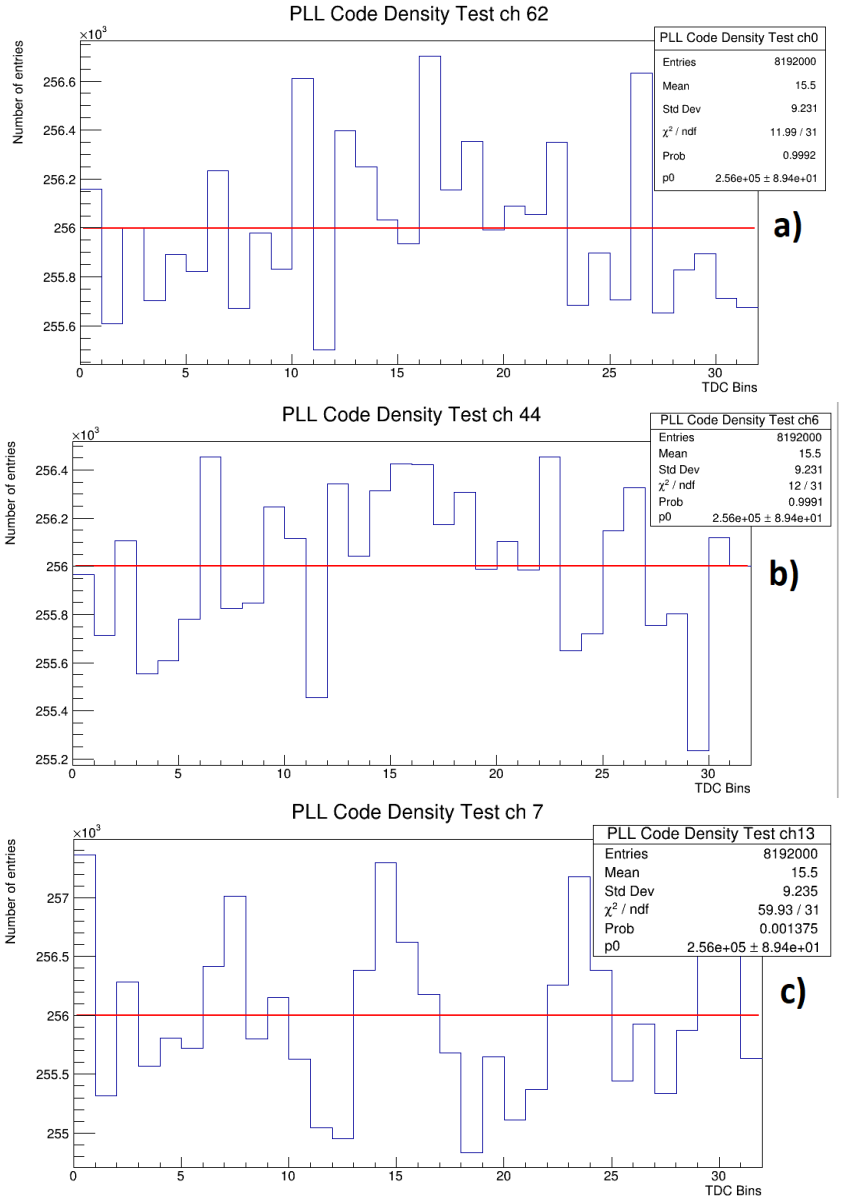

Figure 3.8: These three histograms illustrate the code density test for the 5 bits of the PLL across 3 channels. It can be observed that the oscillations around the mean value are minimal, giving this component a small contribution to the non linearity observed.

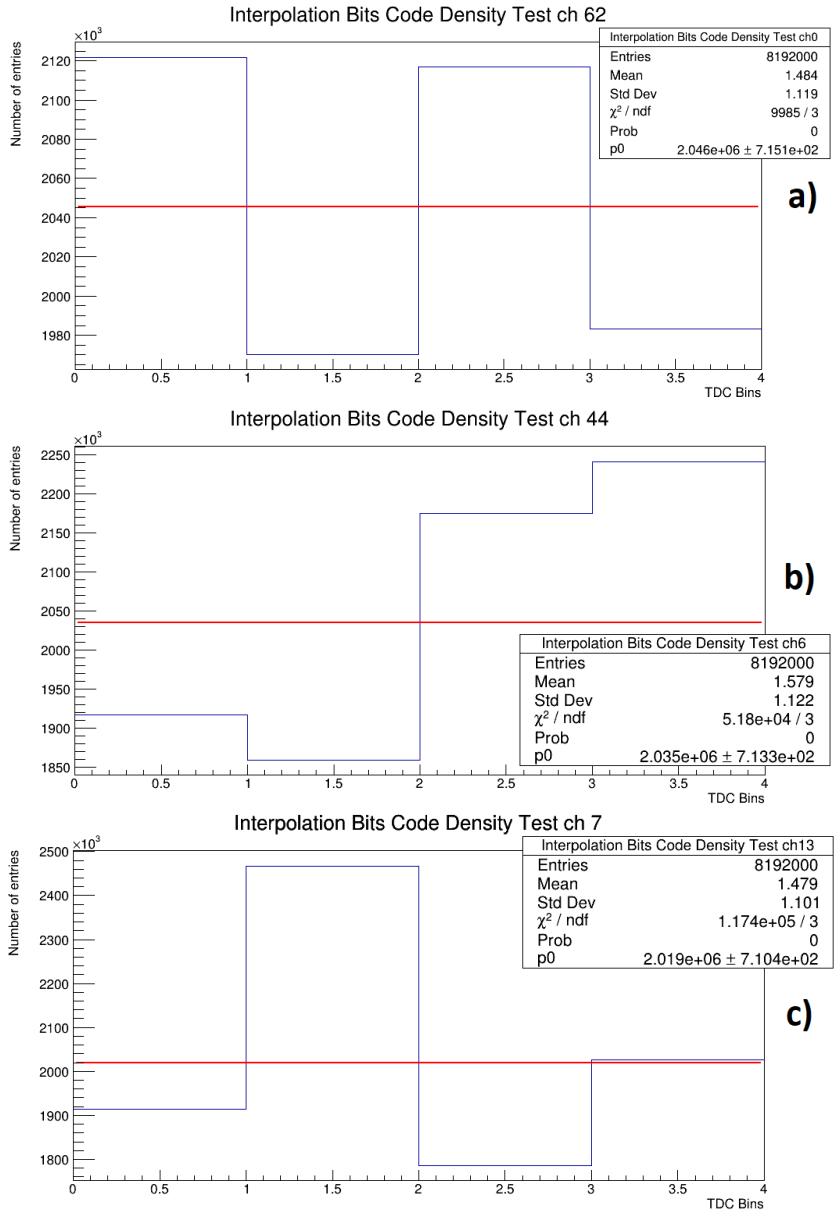

Figure 3.9: These three histograms illustrate the code density test for the 4 bins of interpolation across 3 channels.

Another factor that emerged in the evaluation of non-linearity histograms is that the fluctuation around zero is very wide. Therefore, for further analysis, the DNL values were plotted in another histogram, creating a Gaussian figure (see Figure 3.10). Given the excessive width, it is clear that something is not yet working properly: or the setup was not immune to some outside noise (affecting in turn the internal calibration of the

DLL), or something is missing in the programming procedure of the picoTDC. Despite the good resolution already reached on short delays measurements, further studies are needed.

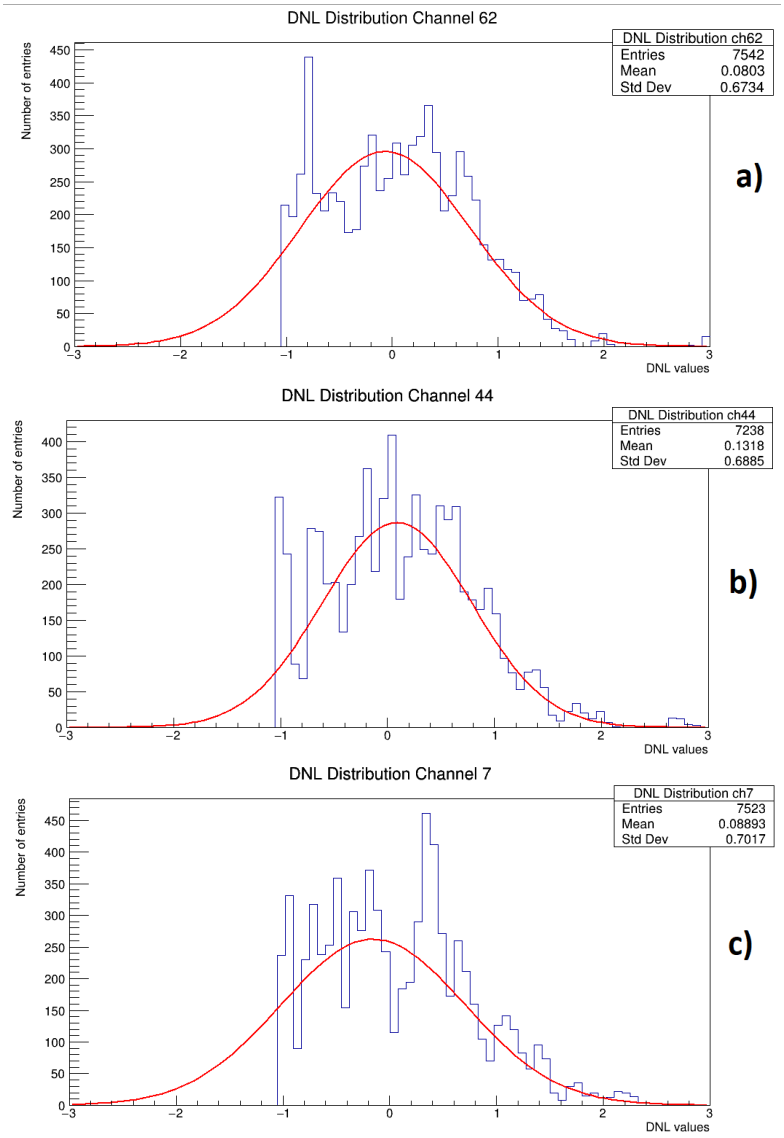

Figure 3.10: These three histograms illustrate the distribution of DNL across 3 channels. The significant width of the Gaussian curves raises suspicion of a potentially significant issue with the stability of the DLL.

3.4 Test of compatibility with ALICE TOF front-end cards

To assess the usability of the picoTDC within the ALICE TOF detector, an additional test was conducted on channels 10, 15, 28, and 30 of the picoTDC. This test utilized pulses provided by the NINO FEA (Front-End Electronics Analogue card). The NINO FEA, based on the NINO ASIC, receives the signals from the Multigap Resistive Plate Chambers (MRPC), which are the sensors used to measure the crossing of charged particles [12]. It handles the amplification and discrimination of signals from the MRPC chambers.
To acquire this data, the experimental setup was configured similarly to the non-linearity test, with the difference being that the FEA served as the pulse generator this time. By connecting the differential LVDS output of the NINO FEA to the input of the picoTDC adapter, measurements were taken on the mentioned channels. The results of these measurements were subsequently graphically represented through a code density test, as shown in Figure 3.11. No significant interface issues were observed between the NINO FEA and the picoTDC.

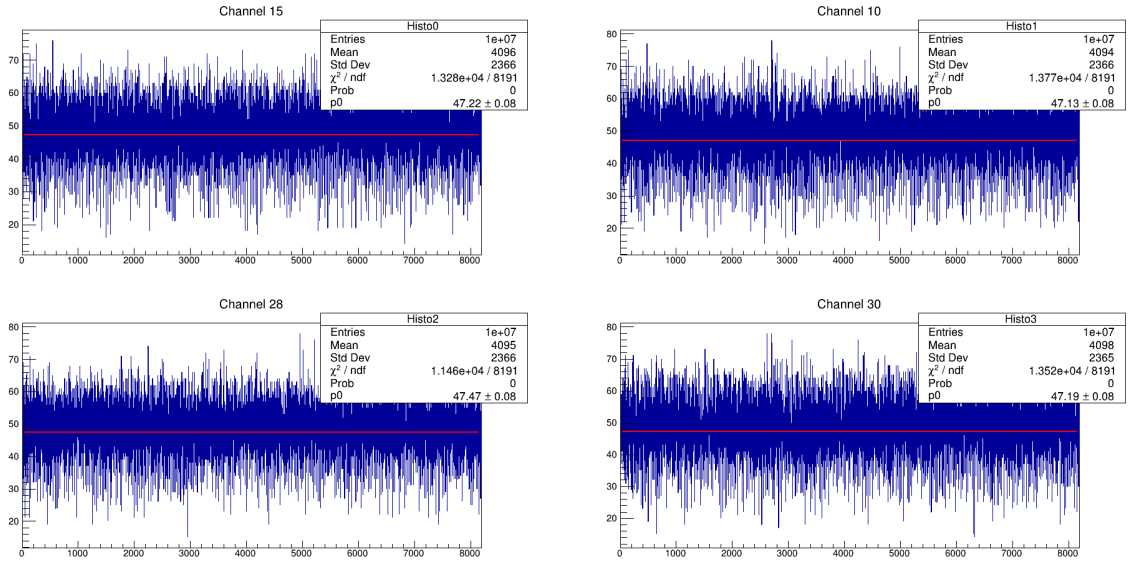

Figure 3.11: Code density test of the picoTDC's response to FEA pulses.

Conclusions

The picoTDC appears to perform well in terms of time resolution, considering very short time delays (up to 450 picoseconds), even if below the stated specification of the chip. Measurements of differential non-linearity (DNL) have highlighted some issues. DNL measurements have revealed non-linearity problems across the entire range of bins. These effects are also likely responsible of the larger than expected distribution measured for fixed delays (up to two LSB).
Regarding the adaptability of the picoTDC to the TOF FEA, no problems were detected: the picoTDC might be used as an alternative to the currently used HPTDC. This was expected, even if the LVDS level of TOF signal is higher than what specified in recommended levels for the picoTDC.
The methodologies for using the trombone have been successfully developed, but the evaluation of chip's differential non-linearity through the code density test currently presents limitations. It is evident that further efforts are required to optimize the chip's operations.
Therefore, I conclude that more in-depth work is needed to improve the chip's configuration, with particular attention to the DLLs, which appear to be improperly calibrated. Regarding the observed pattern in the DNL, it could be attributed to issues in DLL configuration through chip programming or interference from noise associated with the experimental setup.

Appendix A

Test of the adapter board

The adapter developed by INFN Bologna serves the purpose of facilitating the connection between the picoTDC and the experimental setup. This small board is composed of various connectors of different types to provide the necessary versatility for its intended use. In Figure A.1, a photo of the board is provided. Starting from the left side and moving to the right (located on the rear side), you will find the FMC connector. This connector is used to connect the adapter board to the picoTDC demo board. In the central section of the adapter, marked with numbered yellow labels, there are two channels equipped with MCx connectors, specifically channels 14 and 53. Moving to the right in the figure, both above and below, two VHDCI connectors (referred to here as J2 at the bottom and J4 at the top) are visible, each of which has 24 differential channels. One of these connectors is visibly connected to its respective cable. These are the cables used in ALICE TOF to connect FEA and TRM cards. At the center of the right side of the adapter, there is also an IDC 2x17 connector, which has 34 input pins, of which 32 are utilized (resulting in a total of 16 differential channels, while the remaining two pins are in a floating mode).

A series of hardware tests were conducted to evaluate the performance of the adapter used for data collection in the context of picoTDC. During the data collection phases in the tests carried out on the picoTDC, significant issues were observed regarding the channels' response to the provided pulses. These problems raised questions related to channel reading, necessitating a thorough investigation. To address this issue, the scope of the tests was expanded to include the picoTDC adapter board. In the course of data collection, it was observed that some channels did not respond adequately to the sent pulses. These channels exhibited inadequate responses, including detecting an excessive or insufficient number of pulses or, in some cases, remaining entirely inactive. To accurately assess the extent of these problems, a detailed mapping of the channels was performed. This procedure involved sequentially providing input to each channel using a pulse generator. During this process, it became evident that some channels were consistently inactive or exhibited limited activity, while others were subject to significant

Figure A.1: "INFN Bologna's Adapter Board" - A specialized adapter board developed by INFN Bologna to connect the picoTDC to various instruments.

noise. It's important to note that these tests were conducted on both available picoTDC demo boards. An interesting pattern emerged: while some issues were consistent between the two boards, in other cases, there was no direct correspondence between the observed anomalies. This led to the conclusion that the cause of the malfunctions initially attributed to the adapter board may not be predominant. However, concerning the other three connectors, it was observed that some channels exhibited fluctuations between connectors or, in some instances, did not appear on either of the two boards. We have provided in tables A.1 and A.2 the mapping of the VHDCI connectors on which this investigation focused. In the table regarding connector J4 (Table A.1), discrepancies were found in some channels, which we highlighted in red. As evident, the channels did not match between the two boards. For example, where one chip had channel 35, the other had channel 56. In Table A.2, there were differences in the number of channels and the presence of gaps in the channels that we found as present in the other connector. It's possible to presume that these two missing channels are channels 35 and 40, which were not found in connector J4. This might indicate that these channels are unable to read the signal or that there are connectivity or configuration issues associated with these channels. Further investigation is required to precisely determine the cause of these discrepancies and work toward resolving any issues with the missing or non-responsive channels. Obviously the integration tests with the ALICE TOF FEA cards, that require the VHDCI cable and the adaptor were conducted on correctly working channels.

PicoTDC demo board 1		PicoTDC demo board 2	
Connector VHDCI J4			
0	No signal	0	Working
1	Noisy*	1	Noisy
2	Working	2	Working
3	Noisy	3	Working
4	Noisy	4	Working
5	Working	5	Working
6	No signal	6	Working
7	Working	7	Working
8	No signal	8	Working
9	Noisy*	9	Noisy
31	Working	31	Noisy
35	Working	56	Noisy
36	No signal	36	Working
37	No signal	37	Working
38	No signal	38	Working
39	Noisy	39	Working
40	Noisy	62	Noisy
41	Working	41	Noisy
42	Noisy	42	Noisy
43	Working	43	Noisy
44	Working	44	Working
45	Noisy	45	Noisy
50	Noisy	50	Noisy
54	Working	54	Working

Table A.1: In this table is shown the mapping of the VHDCI J4 connector channels for the two demo boards. Here, 'noisy' means there is a signal even when the driver is turned off, 'no signal' when the signal is not displayed, and 'working' when the signal is actually present. The channel number have been taken from the adapter's schematic.

PicoTDC demo board 1		PicoTDC demo board 2	
Connector VHDCI J2			
10	Working	10	No signal
11	No signal	11	No signal
12	No signal	12	Working
13	Noisy	13	No signal
14	No signal	14	No signal
15	Working	15	No signal
17	Working	17	No signal
18	No signal	18	No signal
19	Noisy	19	No signal
20	No signal	20	No signal
21	No signal	21	No signal
22	No signal	22	No signal
23	No signal	23	Working
24	Working	24	No signal
25	Working	25	No signal
26	No signal	26	No signal
27	No signal	27	Working
28	Working	28	No signal
29	No signal	29	No signal
30	Working	30	No signal
32	No signal	32	Working
34	No signal	34	Working
56	Working	–	–
62	Working	–	–

Table A.2: In this table is shown the mapping of the VHDCI J2 connector channels for the two demo boards.

Bibliography

[1] P. Antonioli, *Generalità convertitori analogico digitali*, Corso di laurea magistrale in Physics, Università di Bologna, **2022**.

[2] P. Antonioli, *Generalità ADC-TDC*, Corso di laurea magistrale in Physics, Università di Bologna, **2022**.

[3] M. C. Corporation, *Data Acquisitio Handbook*, **2012**.

[4] `https://transistorized.net/post/stdal/post56.htm`.

[5] B. Razavi, *IEEE Solid-State Circuits Magazine* **2018**, *10*, 9–15.

[6] J. Christiansen, *HPTDC High Performance Time to Digital Converter*, CERN/EP - MIC, **2002**.

[7] M. Horstmann, *PicoTDC Picosecond Time to Digital Converter*, CERN EP-ESE-ME, **2021**.

[8] M. H. et al., picoTDC: Pico-second TDC for HEP, `https://espace.cern.ch/PicoTDC/Shared%20Documents/Presentations/DIRC_2019_samuele.pdf`.

[9] S. Inc, `https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/user-guides/Si5341-D-EVB.pdf`.

[10] Colby Instrument XT-200, `https://www.colbyinstruments.com/xt-200`.

[11] `https://gitlab.cern.ch/dporret/picotdc-demo/`.

[12] F. A. et al., *Nuclear Instruments and Methods in Physics Research Section A: Accelerators Spectrometers Detectors and Associated Equipment* **2004**, *533*, Proceedings of the Seventh International Workshop on Resistive Plate Chambers and Related Detectors, 183–187.