POLITECNICO DI TORINO

Master's Degree in Electronic Engineering

Master's Degree Thesis

Design and implementation of a Time to Digital Converter on Field Programmable Gate Array

Supervisor

Prof. Sarah Azimi

Co-Supervisors

Prof. Luca Sterpone

Ph.D. Corrado De Sio , Eng. Eleonora Vacca

Candidate

Arash Amini Bardpareh

October 2024

Summary

This thesis is part of a HONEY project, Progetti di Rilevante Interesse Nazionale (PRIN) in a collaboration with università di Torino (UNITO) and INFN Torino-which focuses on developing innovative hybrid technology that will substantially enhance both beam monitoring and online treatment verification in Charged Particle Therapy (CPT). Charged Particle Therapy is a form of cancer treatment that utilizes beams of charged particles, such as protons or carbon ions, to precisely target and destroy cancer cells. One of the critical challenges in this therapy is ensuring the accurate delivery of the charged particle beam to the tumor while minimizing damage to surrounding healthy tissues. To address this, the HONEY project aims to create a new hybrid system that integrates advanced beam monitoring and real-time treatment verification. The ultimate goal is to develop and construct a prototype that incorporates both of these functions by leveraging high-performance data acquisition and real-time analysis. This prototype is expected to significantly improve the accuracy and efficiency of CPT, ensuring that the particle beam is properly monitored and the treatment is continuously verified during the entire process. The first step in achieving this goal was the implementation of a high-resolution Time-to-Digital Converter (TDC) on an FPGA (Field-Programmable Gate Array), which had to meet specific technical requirements. A TDC is essential for precisely measuring the time intervals between events, which is crucial for ensuring the correct timing in the beam monitoring and verification processes. The design had to be both highly accurate and capable of operating at very high speeds to meet the stringent demands of CPT. To implement the TDC, the tapped-delay line method was selected as the preferred design approach. This method is particularly effective for time measurement as it relies on capturing the time interval between two signals: a "start" signal that marks the beginning of the interval and a "stop" signal that signifies its end. The tapped-delay line consists of a series of delay elements arranged in a chain, and each element introduces a small delay to the signal as it passes through. At the moment the "stop" signal arrives, the state of the delay chain is captured using latches. These latches store the output of the delay elements at the precise moment the stop signal is received. The output of these latches is then a series of bits, where each bit corresponds

to the state of a particular delay element. If a bit is high, it indicates that the signal has passed through that specific element by the time the stop signal was received. By counting the number of high bits (or "ones") after the arrival of the second signal (the stop signal), a digital representation of the time interval can be obtained. This process effectively converts the analog time difference between the two signals into a digital value. The final time interval measurement is obtained by multiplying the digital count by the resolution of the TDC. The resolution is determined by the delay of the individual delay elements in the chain. Therefore, the design's resolution directly depends on the choice of these delay elements. Since the goal was to design the TDC on an FPGA, achieving high resolution required selecting an FPGA element that could produce the smallest possible delay. After evaluating the available options within the FPGA, the carry4 and carry8 blocks inside the Configurable Logic Blocks (CLBs) were chosen. These blocks were selected because they generate the smallest possible delay and are easily accessible by the latches, making them ideal for implementing the delay line. In earlier design iterations, the delay line was implemented on the Pynq-Z2 board, and initial tests were conducted. However, it was found that the number of carry blocks available in a single row within the same clock region was insufficient. As a result, the maximum measurable delay in that configuration was limited, which did not meet the required specifications. To overcome this limitation, the design was transferred to the Kintex Ultrascale+ FPGA. This board offers more carry blocks in a single row and within the same clock region, allowing for a longer delay line. By fully utilizing the available carry blocks on the Kintex Ultrascale+, the design achieved a measurable delay of 3 nanoseconds (ns) with a resolution of 30 picoseconds (ps). This high-resolution timing measurement is a crucial step toward realizing the final hybrid prototype for beam monitoring and treatment verification in Charged Particle Therapy. This achievement, along with the ongoing testing and optimization of the TDC, positions the project to advance toward building a fully integrated hybrid system. This system is expected to significantly enhance the precision of beam monitoring and treatment verification in Charged Particle Therapy, ultimately improving cancer treatment outcomes.

Acknowledgements

I would like to express my deepest gratitude to Prof. Sarah Azimi, Prof. Luca Sterpone, Ph.D. Corrado De Sio, and Eng. Eleonora Vacca, as well as all my colleagues in our group also all our colleagues at INFN and medical physics department at UNITO, for their unwavering encouragement and motivation. Their constant support pushed me to strive for excellence and persevere, a truly invaluable lesson that I will always cherish.

I am also profoundly thankful to my family and friends for their constant presence, and to my girlfriend for her boundless patience, support, and encouragement throughout this journey.

Table of Contents

List of Tables VIII

List of Figures IX

Acronyms XII

1 Introduction 1

2 Background 4

2.1 Field Programmable Gate Arrays 4

2.1.1 Key Characteristics 4

2.1.2 Architecture and Resources 5

2.1.3 Comparison with Other Technologies 5

2.2 Vivado 7

2.2.1 Key Features and Capabilities 7

2.3 Time To Digital Converter 9

2.3.1 Working Principle 9

2.3.2 TDC Architectures 9

2.3.3 Performance Metrics 10

2.3.4 Applications of TDCs 10

2.3.5 Advancements in TDC Technology 11

2.4 ESA-ABACUS 12

3 State-of-the-Art TDC Modules 14

3.1 Overview of Available TDC Modules 14

3.1.1 TDC7200 14

3.1.2 picoTDC 14

3.1.3 CAEN picoTDC 15

3.2 Implementation TDC on ASIC vs FPGA 16

4 Implementation Process 17
4.1 Delay line: 20
4.2 Counter: 37
4.3 bit counter: 37
4.4 Encoder: 37
4.5 Final ALU: 38
4.6 Memory: 38
4.7 Verification of the configuration memory of the FPGA exposed to radiation: 41

5 Conclusion and Future Work 42
5.1 Testing the latest version 42
5.2 Conclusion 42
5.3 Future Work 43

A Test Instruments And Setup: 44
A.1 Oscilloscope: 44
A.2 Pulse Generator: 45
A.3 The last setup: 46

Bibliography 48

List of Tables

4.1 Comparison Of Delay For Different Blocks In KCU105 20
4.2 Delay Summary for Carry4 Block 21
4.3 Delay Summary For Carry8 Block 24
4.4 FMC HPC Connector Overview 34
4.5 Delay Line Offset Values in Different Configurations 35

5.1 TDC Analysis Table 42

List of Figures

1.1 Treatment Technique during CPT 2
1.2 The Proposed DAQ Architecture 3

2.1 CLB 6
2.2 EsaAbacus Front 12
2.3 EsaAbacus Back 12
2.4 EsaAbacus Board 1 13
2.5 EsaAbacus Board 2 13

3.1 CERN picoTDC Architecture 15
3.2 CAEN TDC SCHEMATIC 16

4.1 TDC Data Path 18
4.2 TDC Tapped Delay Line 18
4.3 Timing Diagram 19
4.4 Delay Line Implementation 19
4.5 TDC Tapped Delay Line With Latch 20
4.6 pynq-z2 Board 21
4.7 Asymmetry In Delay Line 22
4.8 Delay Line Implementation Outputs In pynq-z2 With Encoder . . . 23
4.9 KCU105 Board 24
4.10 Delay Line Outputs Part 1 25
4.11 Delay Line Outputs Part 2 26
4.12 Delay Line Outputs Before Reordering - Part 1 and 2 27
4.13 Delay Line Outputs Before Reordering - Part 3 and 4 28
4.14 Delay Line Outputs Before Reordering - Part 5 and 6 29
4.15 Delay Line Outputs After Reordering - Part 1 and 2 31
4.16 Delay Line Outputs After Reordering - Part 3 and 4 32
4.17 Delay Line Outputs After Reordering - Part 5 and 6 33
4.18 FMC Connector 35
4.19 Error In Measurments 36

4.20 TDC Block Diagram 40

A.1 Oscilloscope 44
A.2 Pulse generator 45
A.3 The last setup 1 46
A.4 The last setup 2 47

Acronyms

CPT

Charged Particle Therapy

PRIN

Progetti di Rilevante Interesse Nazionale

UNITO

Università di Torino

INFN

Istituto Nazionale di Fisica Nucleare

FPGA

Field-Programmable Gate Array

TDC

Time-to-Digital Converter

CLB

Configurable Logic Block

ns

nanoseconds

ps

picoseconds

ToF

Time of Flight

PET

Positron Emission Tomography

PGT

Prompt Gamma Imaging

CERN

European Organization for Nuclear Research

ASIC

Application-Specific Integrated Circuit

LVDS

Low Voltage Differential Signaling

LSB

Least Significant Bit

ToA

Time of Arrival

ToT

Time over Threshold

CFD

Constant Fraction Discriminator

DAC

Digital-to-Analog Converter

DLL

Delay Locked Loop

CMOS

Complementary Metal-Oxide-Semiconductor

HDL

Hardware Description Language

Chapter 1

Introduction

The objective of this research is to design and develop a precise, high-performance data acquisition system for monitoring particles during Charged Particle Therapy (CPT) to enhance in-vivo treatment verification. CPT, also called particle radiation therapy, uses charged particles like protons for cancer treatment. Unlike X-rays or gamma rays, these particles allow for more accurate targeting of tumors while reducing harm to surrounding healthy tissues. However, it is essential to monitor beam parameters and particle range within patients to ensure optimal treatment. Further technological advancements are needed to enhance treatment outcomes and introduce new techniques. Current detectors, such as gas-filled ionization and micro-pattern gaseous detectors, face limitations in sensitivity and response time. Solid-state detectors provide direct particle measurements but require improved data processing methods. This project aims to develop a fast and accurate data acquisition and processing architecture to efficiently handle and extract the required information from these measurements.

The process of monitoring beams and extracting relevant information involves several key steps. First, a particle accelerator is employed to generate and accelerate charged particles, such as protons or carbon ions, which are referred to as source particles. These particles form the therapeutic beam, which is then carefully shaped and controlled to align with the specific treatment plan. Parameters like energy, intensity, and direction of the beam are adjusted to ensure precision in targeting the tumor. In addition to serving as a tool for tumor treatment, the information derived from the source particles can be further analyzed to gain insights about the tumor itself.

As the primary particle passes through the tumor, it releases a portion of its energy, which is responsible for destroying the cancerous cells. However, this interaction between the primary particle and the tumor also generates secondary particles. These secondary particles exit the patient's body, and their behavior provides valuable information. By measuring parameters such as the Time of Flight

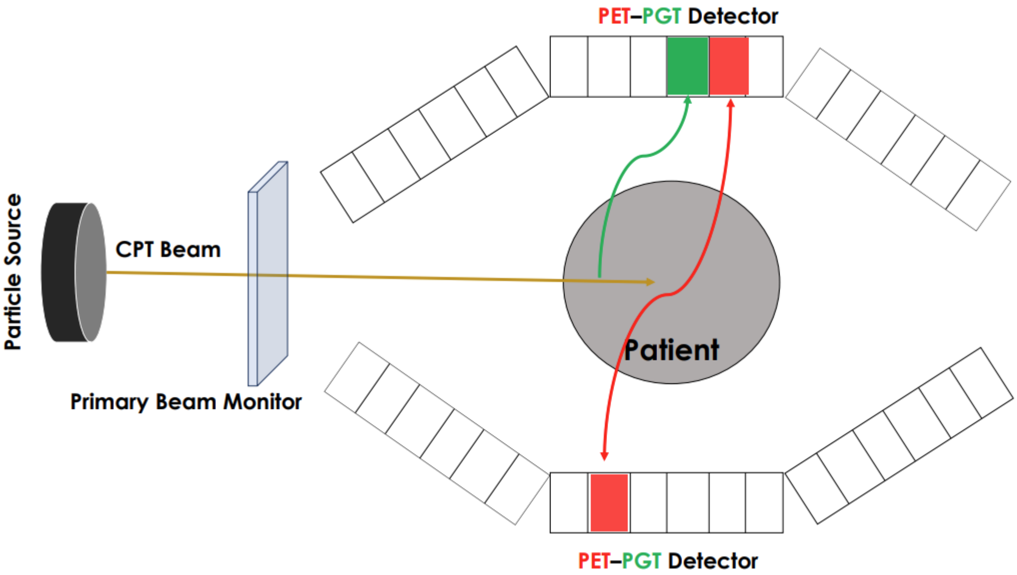


Figure 1.1: Treatment Technique during CPT

(ToF) and energy of both the primary and secondary particles, essential data can be obtained. This data includes details about the tumor's shape, thickness, and dimensions, contributing to a more comprehensive understanding of the tumor and improving treatment accuracy.

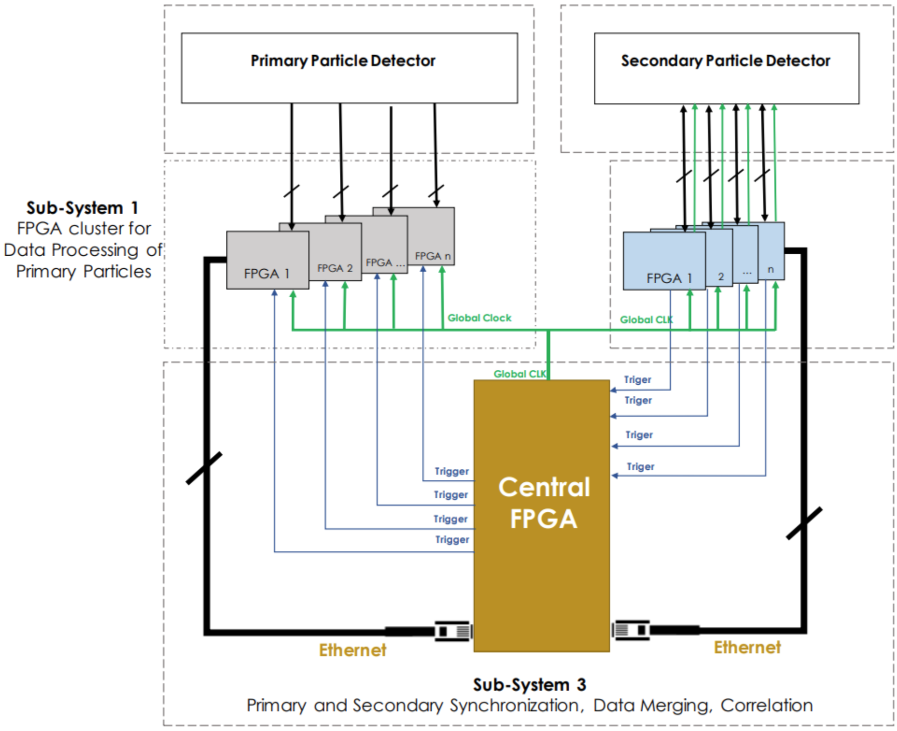

Primary Particle Detector
Secondary Particle Detector
Sub-System 1
FPGA cluster for Data Processing of Primary Particles
Sub-System 2
FPGA cluster for Data Processing of Secondary Particles
FPGA 1
FPGA 2
FPGA ...
FPGA n
FPGA 1
2
...
n
Global Clock
Global CLK
Global CLK
Triger
Triger
Triger
Triger
Trigger
Trigger
Trigger
Trigger
Central FPGA
Ethernet
Ethernet
Sub-System 3
Primary and Secondary Synchronization, Data Merging, Correlation

Figure 1.2: The Proposed DAQ Architecture

To effectively utilize the output from the detectors, the data they generate must be processed with high performance and resolution, which presents a significant challenge. Additionally, another key difficulty lies in synchronizing the two measurements—those of the primary and secondary particles. The time difference between these measurements is critical, as it represents the particle's flight time within the body, particularly through the tumor.

This time difference provides crucial insights into the tumor's characteristics, such as its location and dimensions. By accurately capturing this time difference, the system can offer valuable feedback to the medical team, enabling them to make more informed decisions about the treatment plan for subsequent therapy sessions. This level of precision is essential for optimizing treatment outcomes and ensuring the therapy is as effective as possible.

For measurement of the time of flight of the particles, the need of developing a time-to-digital converter (TDC) in FPGA arises. All the steps of the implementation is discussed in the following chapters.

Chapter 2

Background

2.1 Field Programmable Gate Arrays

Field-Programmable Gate Arrays (FPGAs) represent an adaptable and powerful technology used extensively by electronic system developers for designing and implementing custom hardware solutions. FPGAs are versatile because they can be programmed and reprogrammed after manufacturing, offering significant flexibility compared to traditional Application-Specific Integrated Circuits (ASICs), which are purpose-built for a single task. The reconfigurability of FPGAs makes them a popular choice in industries requiring iterative design cycles, rapid prototyping, or systems that need continuous updates.

Historically, FPGAs were introduced in the 1980s by Xilinx (now part of AMD), who remains one of the leading manufacturers in this space, alongside companies like Intel (through its acquisition of Altera). While FPGAs were initially used for prototyping ASIC designs, their usage has expanded significantly into production systems due to advances in performance, power efficiency, and cost.

2.1.1 Key Characteristics

- **Reprogrammability:** The key feature that sets FPGAs apart from other hardware solutions is their ability to be reprogrammed. This makes them ideal for applications requiring adaptability, such as telecommunications, automotive systems, or military and aerospace applications, where the ability to update hardware functionality without replacing physical components is highly advantageous.

- **Parallel Processing:** FPGAs enable massive parallelism by allowing multiple tasks to be executed simultaneously on different parts of the chip. Unlike general-purpose processors (CPUs), which operate serially and have fixed architectures, FPGAs can be tailored to optimize specific algorithms for high performance, such as real-time image processing or artificial intelligence (AI) workloads.

- **Performance and Power Efficiency:** Compared to software-based solutions on CPUs or GPUs, FPGAs can achieve significant performance gains for specific workloads due to their hardware-level parallelism and customized data paths. This makes them ideal for time-critical applications such as high-frequency trading, data encryption, or digital signal processing. While FPGAs typically operate at lower clock speeds than high-end processors, they often offer improved power efficiency for specialized tasks by executing operations directly in hardware, avoiding the overhead of software layers.
- **Hardware Description Languages (HDLs):** Users program FPGAs using HDLs such as VHDL or Verilog, which define the hardware's behavior at a register-transfer level (RTL). This approach allows developers to create precise control over logic design and hardware implementation. More recently, higher-level tools like HLS (High-Level Synthesis) have been developed to generate HDL code from more abstract programming languages such as C++.

2.1.2 Architecture and Resources

- **Configurable Logic Blocks (CLBs):** These are the basic building blocks of an FPGA and contain look-up tables (LUTs), flip-flops, and carry logic. CLBs can be configured to perform both combinational and sequential logic functions(see figure 2.1).
- **DSP Blocks:** FPGAs often contain dedicated digital signal processing (DSP) blocks optimized for arithmetic operations, which are critical for applications like real-time signal processing and machine learning.
- **Memory Blocks:** FPGAs include embedded memory (block RAM or distributed RAM), which allows for efficient storage and data manipulation close to the processing elements.
- **Programmable Interconnects and I/O Ports:** FPGAs have extensive interconnect networks, enabling flexible routing between logic elements and external peripherals. High-speed I/O ports allow for efficient communication with other systems or components.

2.1.3 Comparison with Other Technologies

FPGAs vs. ASICs: While ASICs provide optimal performance and power consumption for a specific task, they are non-reprogrammable and costly to develop, particularly for low-volume production. FPGAs, in contrast, offer flexibility and lower upfront costs, though they may not match ASICs in performance for a highly specialized application.

FPGAs vs. GPUs: GPUs excel at parallel processing tasks and are well-suited for AI and graphics workloads. However, FPGAs can outperform GPUs in latency-sensitive applications due to their deterministic nature and customized logic pathways.

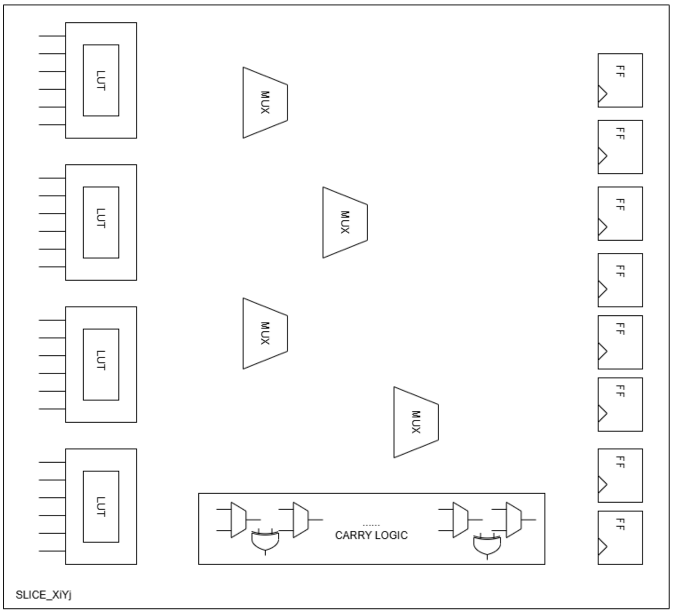


Figure 2.1: CLB

2.2 Vivado

Xilinx Vivado is a comprehensive design suite developed by Xilinx for the development and optimization of digital circuits on FPGA (Field Programmable Gate Arrays) and SoC (System on Chip) devices. Introduced as the successor to Xilinx ISE, Vivado provides a more modern and integrated environment specifically designed to address the growing complexity and performance requirements of today's FPGA designs.

2.2.1 Key Features and Capabilities

- **High-Level Design Abstraction:** Vivado supports RTL (Register Transfer Level) design using traditional hardware description languages like VHDL and Verilog. Additionally, with the use of Vivado HLS (High-Level Synthesis), designers can use C, C++, or SystemC to describe hardware designs, which are then synthesized into RTL. This higher-level abstraction is especially beneficial for software-oriented developers who are more familiar with these programming languages, enabling quicker iterations in design development.

- **Integrated Development Environment (IDE):** Vivado combines synthesis, simulation, and implementation into a single unified platform. Its user-friendly graphical interface enables users to manage the entire FPGA design flow, from design entry to bitstream generation, without switching between multiple tools. The integrated design environment includes a powerful editor, a project manager, and debugging tools for efficient design analysis and optimization.

- **Scalability and Performance:** Vivado was created to handle the increasingly large FPGA devices and complex designs. It is designed for faster synthesis, place-and-route, and optimization algorithms, providing significant improvements in compilation times and overall design performance compared to older tools like Xilinx ISE. This makes it highly scalable for large and high-performance FPGA designs.

- **IP Integration:** Vivado provides access to a rich library of IP (Intellectual Property) cores, both from Xilinx and third-party developers. These IP blocks can be easily integrated into designs, reducing the time and effort required to develop standard functionalities such as communication interfaces, memory controllers, DSP blocks, and more. The IP Integrator tool in Vivado allows for easy drag-and-drop creation of systems using pre-verified IP.

- **Block Design Support:** Vivado includes a block-based design flow where designers can visually create complex systems by interconnecting functional blocks, such as IP cores and custom components. This approach is beneficial in managing large designs, allowing designers to quickly implement and modify the system's architecture at a higher level of abstraction.

- **Advanced Timing and Power Optimization:** Vivado offers advanced capabilities for timing closure and power optimization. It provides detailed timing analysis and constraint-based design methodologies to meet stringent timing requirements. Additionally, the tool offers power analysis and reduction features, helping designers optimize power consumption, which is critical for energy-efficient FPGA designs.

- **Debugging and Simulation Tools:** Vivado includes powerful debugging tools like the Vivado Logic Analyzer and Vivado Simulator, which allow real-time debugging and validation of FPGA designs. These tools enable designers to insert debug probes into their designs, capture and analyze signals on the FPGA, and perform in-depth timing and logic analysis.

- **Support for Zynq and Versal Platforms:** Vivado is fully equipped to handle the development of Xilinx Zynq SoCs and Versal ACAPs (Adaptive Compute Acceleration Platforms), which combine FPGA fabric with processor cores (ARM-based or AI accelerators). This makes Vivado suitable not only for traditional FPGA development but also for creating complex embedded systems.

2.3 Time To Digital Converter

A **Time-to-Digital Converter (TDC)** is an electronic device used to measure very small time intervals with high precision by converting time differences into a digital value. TDCs are typically used in applications where precise time measurements are crucial, such as physics experiments, radar systems, medical imaging, and even telecommunications. These devices can achieve resolution in the range of picoseconds (ps), making them vital in situations requiring extremely accurate time interval measurements.

2.3.1 Working Principle

The basic operation of a TDC involves measuring the time delay between two events, such as the arrival of a signal and a reference clock pulse. The time interval between these events is then converted into a digital value that represents the duration between them in discrete steps, or "bins." The TDC output is a digital word that corresponds to the elapsed time between the events, typically in units as small as picoseconds. The precision of the TDC is determined by the size of these discrete steps, which depend on the specific architecture of the device.

2.3.2 TDC Architectures

There are several common architectures used to implement TDCs, each optimized for different performance characteristics, such as resolution, speed, and power consumption:

- **Counter-Based TDCs**: These TDCs use a high-frequency counter to track the number of clock cycles between two events. While simple, the resolution of counter-based TDCs is limited by the clock frequency, making them less suitable for applications requiring very fine time resolution.

- **Delay Line TDCs**: In delay line architectures, the incoming signal passes through a series of delay elements, and each element introduces a known delay. By determining how many delay elements the signal passes through before a reference event occurs, the TDC can measure time intervals with very fine precision. The resolution is determined by the delay introduced by each element, which can be on the order of tens of picoseconds.

- **Ring Oscillator-Based TDCs**: A ring oscillator is used to create a periodic signal with a known period. The time difference between events is measured based on how many cycles of the oscillator occur between the events. This method can offer high resolution by using oscillators with very short periods.

- **Vernier TDCs**: This architecture leverages two oscillators or delay lines with slightly different frequencies or delays. The difference in time between the two clocks provides a fine-resolution measurement. Vernier TDCs can achieve extremely high time resolutions, sometimes below 10 ps.

2.3.3 Performance Metrics

Several key performance metrics are used to evaluate TDCs:

- **Resolution**: The smallest time interval that the TDC can distinguish, often measured in picoseconds (ps). Higher resolution TDCs can measure smaller time differences more precisely.

- **Range**: The maximum time interval that can be measured by the TDC before it overflows or resets. This is determined by the length of the counter or the number of delay elements in the architecture.

- **Linearity**: The degree to which the digital output of the TDC is proportional to the actual time interval. Non-linearity introduces errors in measurement, especially for large time intervals.

- **Power Consumption**: Important in low-power applications like mobile devices, wireless communication, or embedded systems.

- **Jitter**: The uncertainty in the timing measurements due to noise or other variations. TDCs with low jitter are required for high-precision measurements.

2.3.4 Applications of TDCs

TDCs are used across a wide variety of fields where accurate time measurements are critical:

- **High-Energy Physics**: TDCs are integral to particle detectors in experiments such as those conducted at CERN. They measure the precise time of particle interactions, helping determine particle trajectories and energies.

- **Time-of-Flight (ToF) Measurement**: TDCs are used in time-of-flight systems, which measure the time it takes for a signal (like a laser or sound pulse) to travel to an object and reflect back. This is crucial in applications like LIDAR systems for autonomous vehicles, 3D scanning, and distance measurement in robotics.

- **Medical Imaging**: In medical devices like Positron Emission Tomography (PET) scanners, TDCs help capture the timing information of gamma photons emitted from radiotracers. This allows for highly accurate 3D imaging of metabolic processes inside the body.

- **Radar Systems**: In radar systems, TDCs measure the time delay between transmitted and reflected signals to determine the distance to an object or its velocity. High precision is required for resolving small differences in distance or speed.

- **Wireless Communications**: TDCs are increasingly used in synchronization of base stations and devices in wireless communication networks, particularly for technologies like 5G, where precise timing is required to ensure low latency and high data throughput.

- **Photonics**: In optical communication and quantum photonics, TDCs are used for measuring the timing of photon arrivals with picosecond precision, enabling time-correlated single-photon counting (TCSPC) and other time-sensitive optical applications.

2.3.5 Advancements in TDC Technology

Recent advancements in TDC technology focus on improving both resolution and power efficiency. **Digital TDCs**, often implemented in Field-Programmable Gate Arrays (FPGAs) or ASICs, are becoming increasingly common due to their reconfigurability and ease of integration. Furthermore, advancements in **all-digital TDCs** leverage modern nanometer-scale CMOS technologies to achieve better resolution, power efficiency, and compact designs, making them suitable for integration into systems-on-chip (SoC) and other portable devices.

TDC technology continues to evolve, with a focus on improving accuracy, reducing power consumption, and expanding their range of applications. As devices become more precise, they are likely to see increased use in emerging fields such as quantum computing, next-generation autonomous systems, and advanced medical diagnostics.

2.4 ESA-ABACUS

A front-end board, named ESA-ABACUS, was developed to read out signals from silicon strip detectors using six 24-channel ASICs (Application-Specific Integrated Circuits), known as ABACUS, which were designed at INFN in Torino. Each of the 144 channels contains a Charge Sensitive Amplifier (CSA), providing a wide input dynamic range (from 4 to 150 fC), followed by a leading-edge discriminator.

A Digital to Analog Converter (DAC) on the board provides a common threshold voltage for all channels of each ASIC, while individual channels can be fine-tuned with an internal DAC to correct any variations in their baseline voltage (pedestal). The dead time for each ABACUS channel is between 5 and 10 ns, which ensures a counting efficiency of 100% for input frequencies of up to 100 MHz or higher, depending on the input charge.

Once particle signals are detected and discriminated, digital pulses are generated and sent off the chip to be counted by three Kintex7 FPGA boards (KC705 evaluation boards), which sample the ABACUS outputs at 1 GHz. Each channel's particle counts are stored in a counter within the FPGA. These boards also control the threshold settings, and a LabVIEW program was developed to read the counters every 100 ms, adjust thresholds, and store data for later analysis. [1]

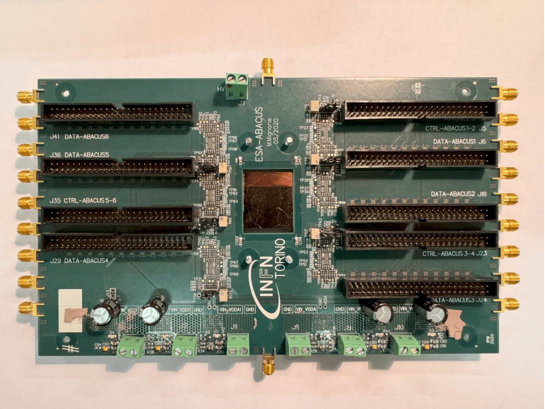


Figure 2.2: EsaAbacus Front

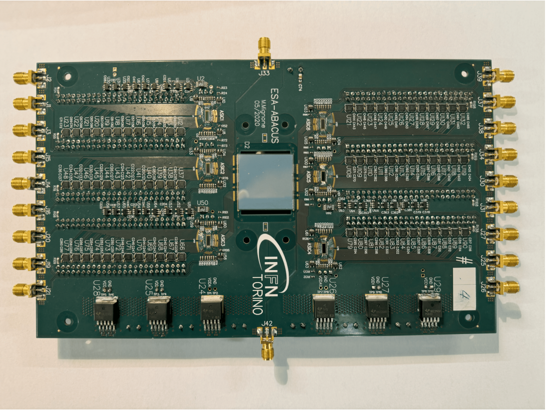


Figure 2.3: EsaAbacus Back

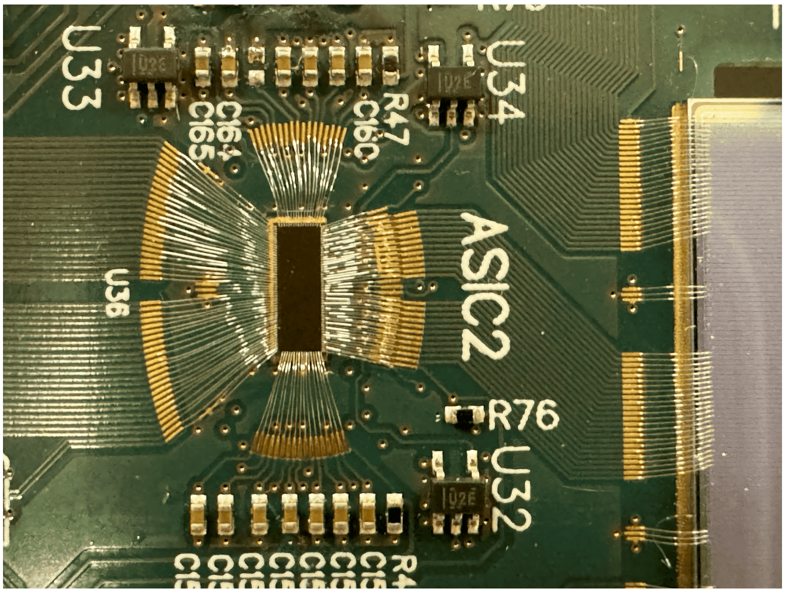


Figure 2.4: EsaAbacus Board 1

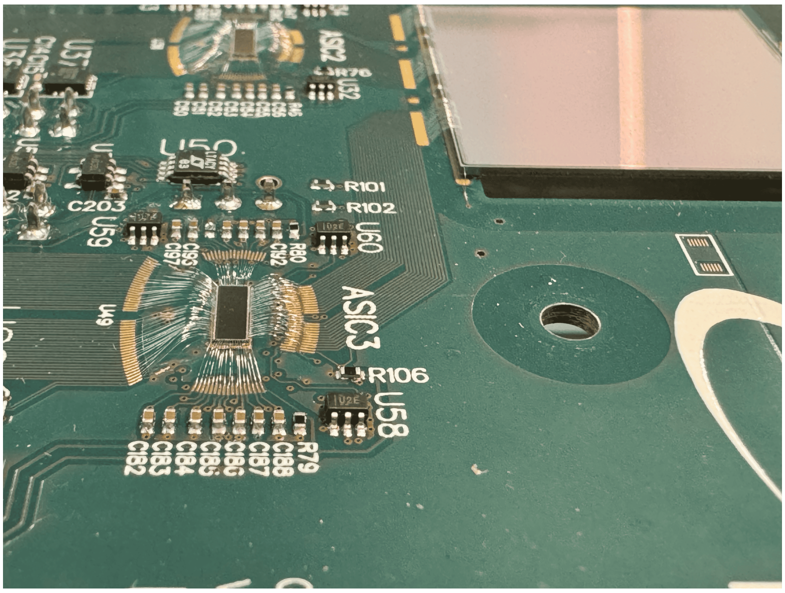


Figure 2.5: EsaAbacus Board 2

Chapter 3

State-of-the-Art TDC Modules

3.1 Overview of Available TDC Modules

3.1.1 TDC7200

The Time to Digital Converter TDC7200 from TEXAS INSTRUMENTS performs the function of a stopwatch and measures the elapsed time (time-of-flight or TOF) between a START pulse and up to five STOP pulses. The ability to measure from START to multiple STOPs gives users the flexibility to select which STOP pulse yields the best echo performance. The device has an internal self-calibrated time base which compensates for drift over time and temperature. Self-calibration enables time-to-digital conversion accuracy in the order of picoseconds. This accuracy makes the TDC7200 ideal for flow meter applications, where zero and low flow measurements require high accuracy. When placed in the Autonomous Multi-Cycle Averaging Mode, the TDC7200 can be optimized for low system power consumption, making it ideal for battery powered flow meters. In this mode, the host can go to sleep to save power, and it can wake up when interrupted by the TDC upon completion of the measurement sequence.[2]

3.1.2 picoTDC

picoTDC a flexible 64 channel TDC with picosecond resolution with 3ps or 12ps time binning developed by CERN is specifically designed to accurately measure the time difference between the emission of primary particles and the detection of secondary particles, which is essential for determining the particle's flight path through the body. PicoTDC is an ASIC implemented in 65nm CMOS process based on Delay Locked loop (DLL) architecture, and it an effective tool for capturing the critical data needed for optimizing treatment and improving the accuracy of tumor targeting. Its ability to provide precise measurements has made it a valuable asset in the context of Charged Particle Therapy. [3]

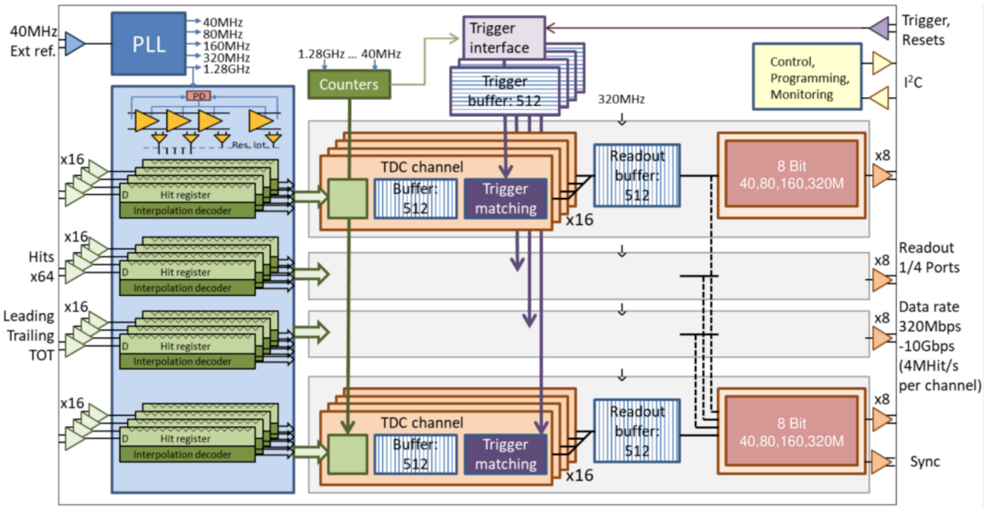


Figure 3.1: CERN picoTDC Architecture

3.1.3 CAEN picoTDC

The CAEN Time-to-Digital Converter (TDC) lineup includes the CERN picoTDC and a flexible FPGA-based architecture, offering enhanced features and adaptability for high-precision timing applications. The TDC is specifically designed for high-resolution, multi-hit time measurements and relies on the picoTDC chip developed by CERN.

One example is the A5203B model, which incorporates an additional mezzanine card with a second picoTDC chip, expanding the module to a total of 128 channels. Each readout channel accepts LVDS (Low Voltage Differential Signaling) inputs and can measure the time of both rising and falling edges with a remarkable least significant bit (LSB) resolution of 3.125 picoseconds. This allows for accurate reconstruction of the Time of Arrival (ToA) of signals, either as an absolute timestamp or as a time difference (ΔT) relative to a common reference pulse (Tref).

Moreover, the PicoTDC chip is capable of acquiring Time over Threshold (ToT) information, which it integrates with the edge timestamps. This ToT functionality enables estimation of signal amplitude, reconstruction of energy spectra, and correction for timing walk. As a result, the system achieves optimal timing resolution without the need for Constant Fraction Discriminators (CFDs), making it an efficient solution for precise timing applications.[4]

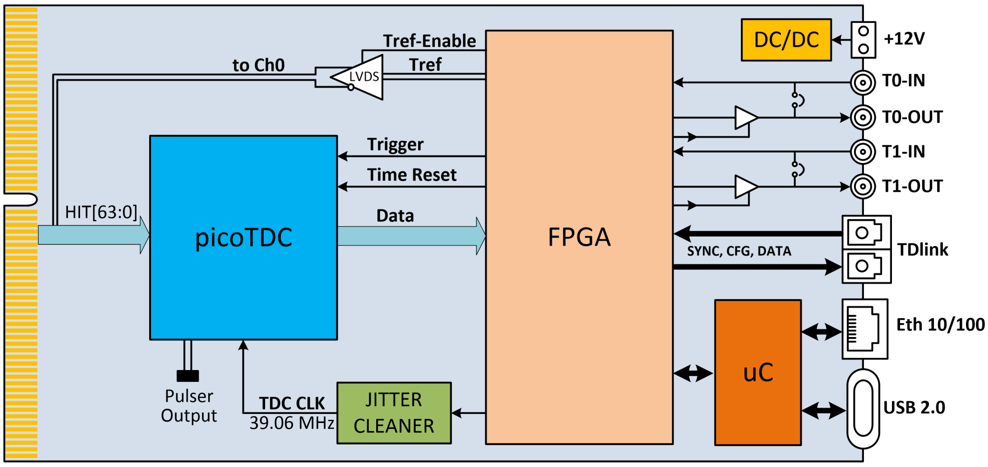


Figure 3.2: CAEN TDC SCHEMATIC

3.2 Implementation TDC on ASIC vs FPGA

Implementing a Time-to-Digital Converter (TDC) on an FPGA is highly advantageous, much like other hardware implementations on FPGAs, due to the significant benefits they offer over ASICs. One of the key advantages is the reconfigurability of FPGAs, which not only reduces design and development costs but also shortens the time to market. Additionally, FPGAs provide flexibility that can address the limitations found in ASIC-based TDCs, such as a restricted number of channels and limited memory capacity.

In our case, the ability to enhance these weaknesses and integrate custom features was a strong motivation for choosing FPGA-based implementation for the TDC. While current reasearchs have been focused on the concept and simulations, the physical implementation has not done yet. in addition, in the whole picture of the project, the DAQ is based on FPGA and we want to take advantage of our resources and have compact and efficient design.

Chapter 4

Implementation Process

In this project, we intend to design and implement a time-to-digital converter (TDC) that achieves sub-nanosecond resolution. While FPGAs are typically suited for synchronous designs with nanosecond-level resolution, our goal is to achieve a timing resolution of at least one hundred picoseconds (ps). This poses significant challenges. The structure of FPGAs introduces limitations, especially when it comes to the implementation of complex asynchronous circuits. Despite these constraints, we are attempting to push the boundaries and develop a high-resolution TDC using FPGA technology.

In order to achieve this goal, First step was studing different architectures for designing the TDC, each has their advantages and disadvantages but the main issue was being implementable in FPGA.
Couple of the available architectures are as follows:

1. Vernier Gated Ring Oscillator Time-to-Digital Converter [5]

2. Tapped Delay Line [6] [7], [8]

3. Time Stretching [9]

The one which has been chosen is the Tapped Delay Line, due to simplicity and being implementable based on available rescources in FPGA. As it can be seen in the figure 4.2 it includes a line which the first signal enters, and it contains delay elements. each delay element is connecter to a flip-flop at its output, the the flip-flops output values have been captured at the time the socond signal arrives(here its the rising edge of the clock) which shows upto where the signal propagets in the delay line. The whole idea of this architecture as can be seen in figure 4.1 is to have three measurment units, two delay lines and a synqronous counter, the first delay line measures the time between first signal and the next rising edge of the system clock, the counter is also starts counting after the same rising edge of the clock, the second delay line measures the time between the second signal and the next rising edge of the system clock . As it can be seen in the figure 4.3 the time difference between the first and second signals can be calculated by the following formula:

$$T = T_1 + T_2 - T_3 \tag{4.1}$$

The dynamic range of the TDC is defined by the counter and the resolution of the TDC is determined by the delay of the delay elements in the delay lins.

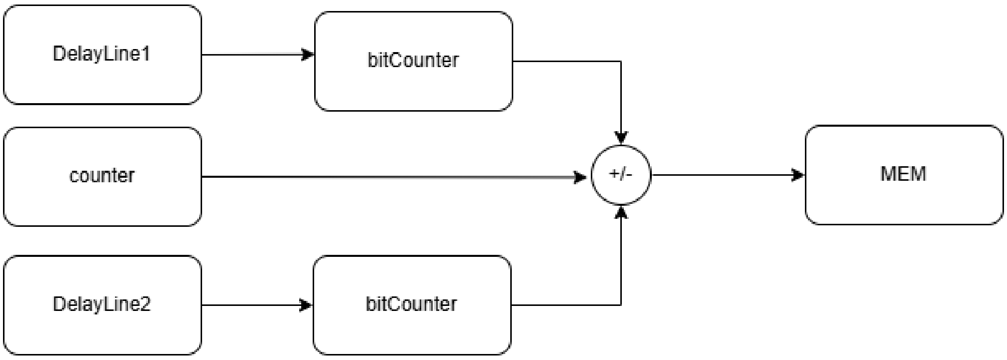


Figure 4.1: TDC Data Path

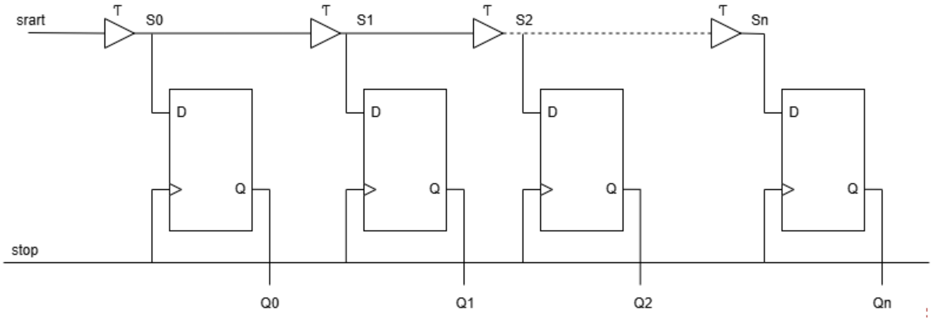


Figure 4.2: TDC Tapped Delay Line

For this purpose, we explored different FPGA resources, such as LUTs and flip-flops, to create the necessary delay elements for a tapped delay lines. However, the delays produced by most of these components were too large to meet our sub-nanosecond resolution requirements. Also being able to have the most symetrical implementation is crucial for the accuracy of the TDC which we considered. Amonge the resorces, the carry logic within CLBs—designed for fast arithmetic (table 4.1), provides the smallest and most consistent delay available in the FPGA. These carry

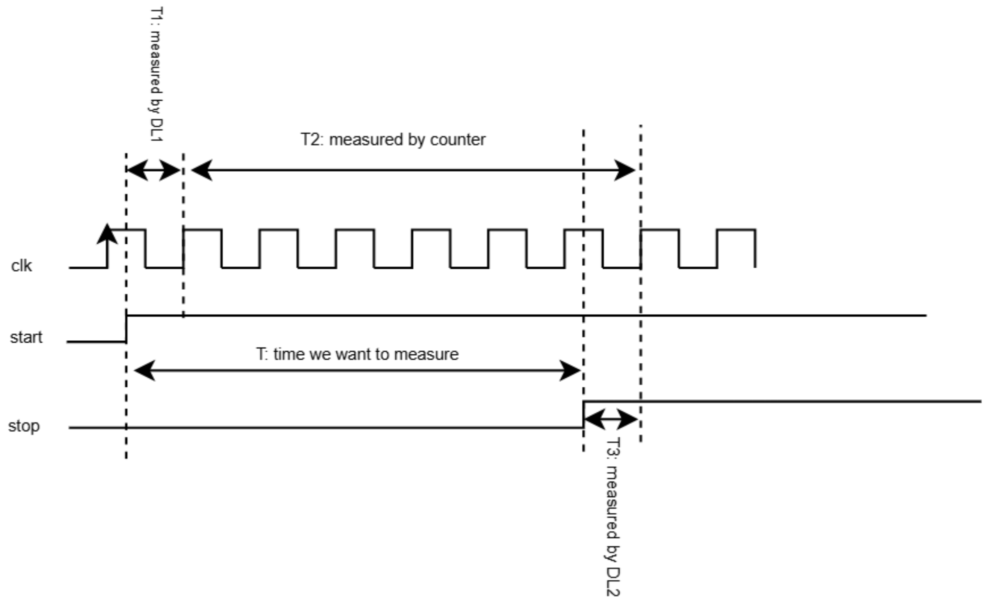


Figure 4.3: Timing Diagram

chains allow logic elements to be connected symmetrically, with minimal delay, making them ideal for precise timing measurements.

By utilizing the carry blocks, we have constructed delay lines that is both efficient and scalable within the FPGA architecture. These carry chains, are able to connected directly to flip-flops, allowing for precise timing control, which is crucial for accurate time-to-digital conversion. The accessibility and reliability of these carry blocks make them the best candidates for achieving the high-resolution TDC we require.

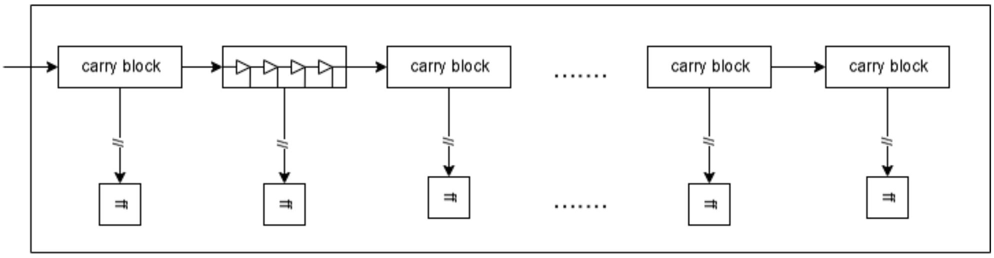


Figure 4.4: Delay Line Implementation

Property	LUT	CARRY BLOCK	FF
CLASS	speed_model	speed_model	speed_model
DELAY (ns)	0.185	0.159	0.260
FAST_MAX (ns)	0.097	0.078	0.126
FAST_MIN (ns)	0.071	0.047	0.099
SLOW_MAX (ns)	0.185	0.159	0.260
SLOW_MIN (ns)	0.136	0.090	0.204

Table 4.1: Comparison Of Delay For Different Blocks In KCU105

4.1 Delay line:

Initially due to our lack of control over the delay of the first or second signal occurance and the next clock cycle, to verify the delay line the delay line was implemented, useing latches instead of flip-flops (figure 4.5). The delay line it self has been simulated and also tested by real signals but instead of clock as second signal of the delay line, an actual second signal, generated by us has been used. Our firts choise of board in early experiments was in pynq-z2 board(Figure 4.6) which has a zynq 7020 FPGA (28 nm xilinx Artix-7 FPGA). We used all available carry4 blocks that can create a line in one clock region. each carry4 block gives 4 outputs to the same CLB flip-flops. the last output of the previous carry4 block is the input of the next carry4 block untill the last point of delay line.

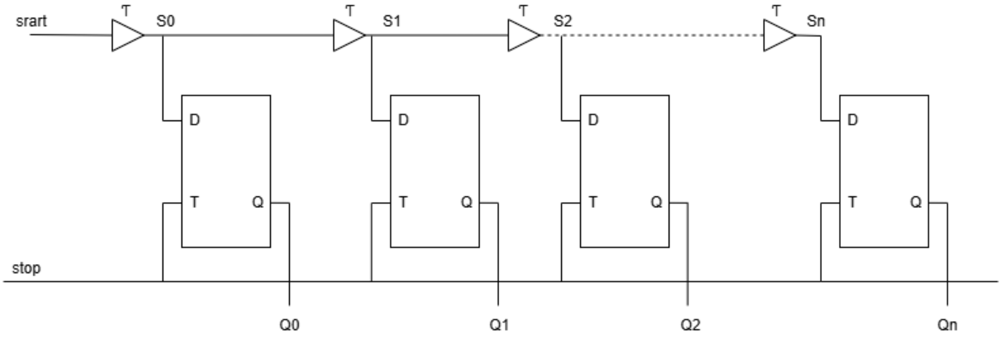


Figure 4.5: TDC Tapped Delay Line With Latch

We connect 32 carry4 blocks in series (each give 4 outputs) to create a 128 measuring point delay line. The vivado report about carry4 blocks is as table 4.2.

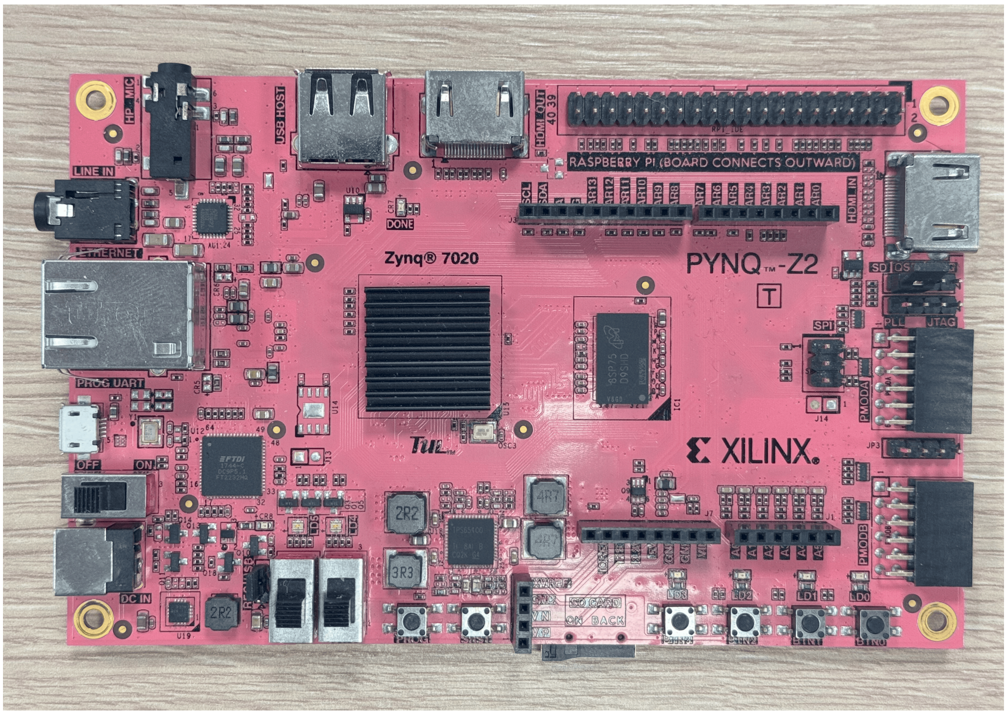


Figure 4.6: pynq-z2 Board

Property	Value (ns)
DELAY	0.281
FAST_MAX	0.111
FAST_MIN	0.084
SLOW_MAX	0.281
SLOW_MIN	0.214

Table 4.2: Delay Summary for Carry4 Block

The table shows the carry4 block can generate delay in range of 84 to 281 ps,when it recives a signal as input and generate that signal at it's output and if we devide these values by 4, because three more outputs can be used in the middle of each carry4 block. the actual delay at each output of the carry4 block could be computed by devivig the valy by four, which is in range of 21 to 70 ps. So with vivado information a view about our resolution could be find, and also the variation based on different conditions of FPGA provide a view about the accuracy of the delay generated by the carry4 block.

It was expected that the value of the latches, flip to one in a uniform way. A thermotobinary encoder has been used to count and convert the number of ones in a latchs outputs to a binary number, and it basically works as a efficient counter of ones. It took the MSB half and see if there is a one in there, if yes consider fist half as all are ones and count the latch outputs at the MSB half, otherwise it counts the ones in LSB half. It could be more efficient by finidng the most optimal number of stages but at the moment the delay line behavior was the part that has to be observed.

In timing simulation after implementation some jumps in the output at some of our measurments have been observed. These missbehavior also has been observed not only at encoder output but also directly at the latch outputs. The missbehavior was due to the fact that, some latches with higher weight gets one before the latches with lower weight, which were not expected, but as the measerments were in ps scale, haveing even some asymmetry like haveing more fanout in last outputs of each carry block (it connects to a latch also goes to next carry4 as input), in addition the fact that the delay which is generated when we are going from one carry4 to another is more than the delay inside carry4 block, can cause this issue (figure 4.7).

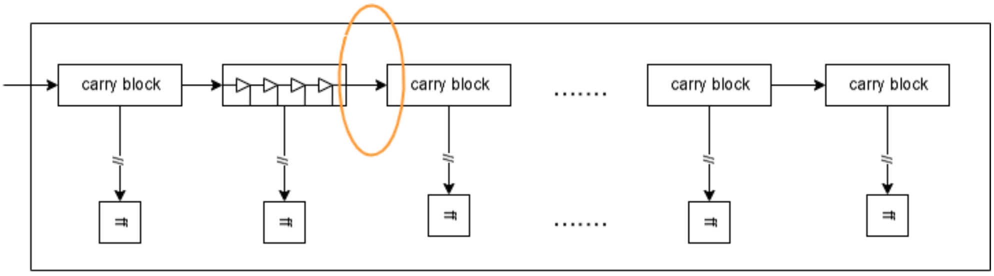


Figure 4.7: Asymmetry In Delay Line

After the simulation, we conducted an experiment to validate the results, and the observed jumps were confirmed. In this test, an additional UART module was added to the design to transmit the results to the PC for visualization. The outcomes can be seen in Figure 4.8.

In this test the delay of the signals was increased 20ps each time, useing continues signals, and the test hs been conitinued approximatly 25s for each step. the signals were generated by a signal generator with 70ps delay precision, and in the graph the most repeated values at output defined in dark blue and less repeated ones in light blue. The variation at output matches the 70ps precision and they are not out of range. also it can be seen that the maximum delay that can be measured by this delay line was 600ps, as at this point all the outputs are one.

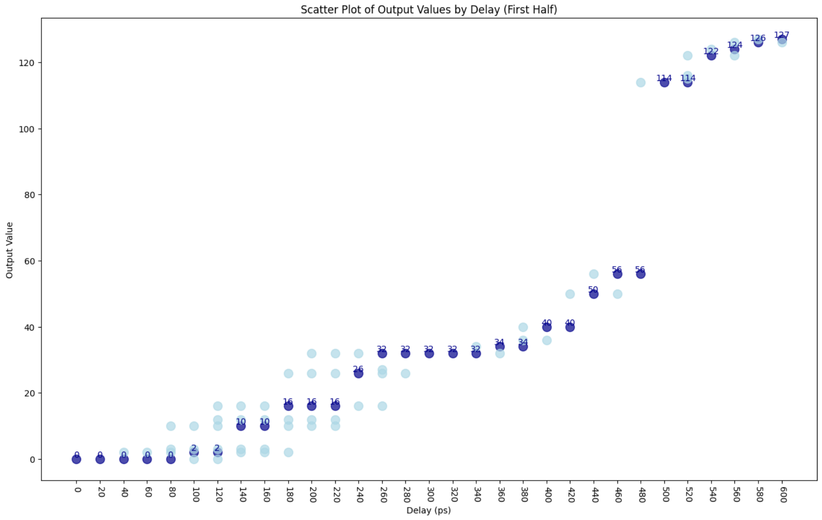


Figure 4.8: Delay Line Implementation Outputs In pynq-z2 With Encoder

As it can be seen in the figure 4.8, the output of the encoder is not as expected, and the jumps in the output are visible. But in most cases the results shows that the delay line in responding and the output is changing when the delay in increamented.

To have a better view and see the exact behavior of the delay line itself, and do not consern about the encoder functionality, the encoder has been removed and all the outputs have been sent to the PC through UART, also The at this point due to matching other future needs (EX. number of the IOs) to avoid redoing same tests and verifications, the board has changed the to kintex ultrascale kcu105(20nm, XCKU040-2FFVA1156E FPGA) Figure4.9. In the new board, there are CARRY8 blocks instead of CARRY4 blocks. The vivado report about carry8 block can be seen in table 4.3.The carry8 block noy only provides 8 sample point, but also as illustrated in the table, it generates less delay, with is totally in range of 47 to 159 ps, so the delay at each sample point can be computer by dividing these values by 8, which is in range of 5.875 to 19.875 ps.

Property	Value (ns)
DELAY	0.159
FAST_MAX	0.078
FAST_MIN	0.047
SLOW_MAX	0.159
SLOW_MIN	0.090

Table 4.3: Delay Summary For Carry8 Block

Figure 4.9: KCU105 Board

The result of the test can be seen in figures 4.10 and 4.11. In this test the same signal generator with 70ps of precision has been used as before, but the delay increased with 10ps each test. Also the most repeated values at output are in dark blue and the less repeated ones are in light blue, and the variation at output is in range of signal generator precision. By looking closer look at the figure, the worst variation at outout is in test with signals with delay of 270ps and 280ps. Even if the generator precision has not been taken into accout, and all variations are due to the delay line error, the difference of the most repeated and most differenct less repeated output is 70ps. by this it can be said the precision is less than 70ps. Also in wost case senario by considering the most repeated value at output, the output changed every 40ps and it provides a view about the resolution of the delay line. Figure 4.11 shows by just changing the board, the maximum delay that can be measured by the delay line increased from 600ps to 1080ps.

It can be seen the delay line outputs doesn't increase one by one as it is expected in theory. This could be the reason, the thermtobin encoder was not working correctly when the output value was about in the middle of maximum range (because as mentioned before the encoder first check the second half and if there was a one at the MSB half latches, it considered the first half as all are ones, but in reality it could happen that some of them are zeros and some in MSB half are ones). The higher resolution of the delay line is another points that is visible in the graph.

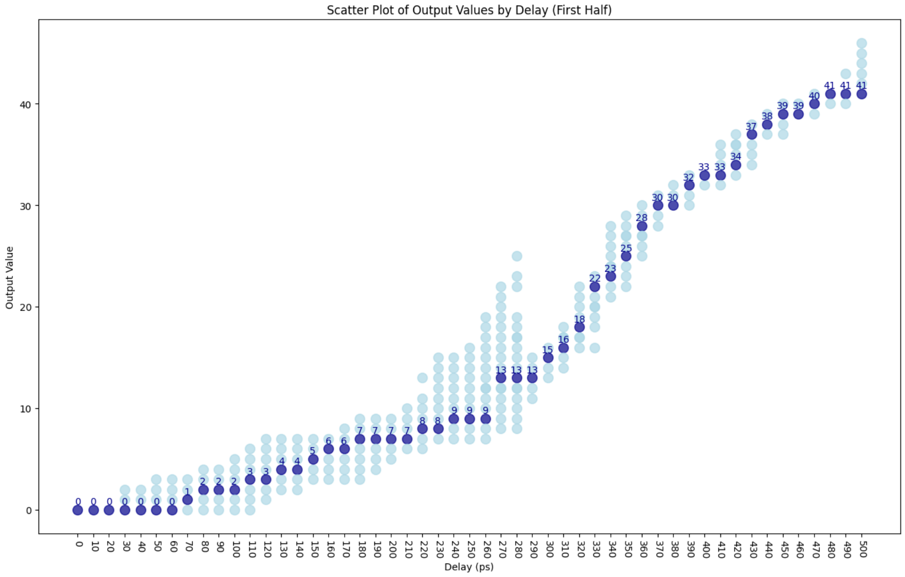


Figure 4.10: Delay Line Outputs Part 1

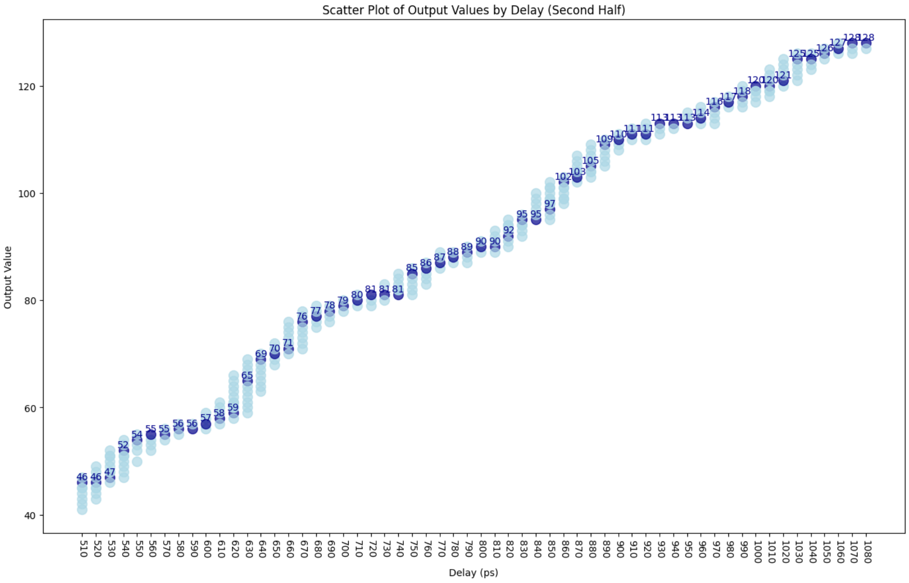


Figure 4.11: Delay Line Outputs Part 2

The problem that will arise from the fact that delay line outputs are not flip in order is not being able to use the efficient encoder, which leads to a very slow encoder. For investigating how to deal with this issue, also haveing better understanding of the delay line behavior, in the next step, the idea of reordering the latches outputs ha been considered, and also due to the fact that the new FPGA with more resources has been used, it provided the longer delay line, so by takeing the full advantage of new board, now the delay line gives 464 sample points (58 carry8 block in same column in same clock region), and by this new implementation the delay line is able to measure upto almost 3ns of delay. In the measurments with same signal generator, and 10ps delay increament at each test, the resolution stays the same, almost about 40ps, so as the precision, which due to the fact that the same device was used was expected. The results before the reordering with 464 sample point delay line can be seen in figures 4.12,4.13,4.14.

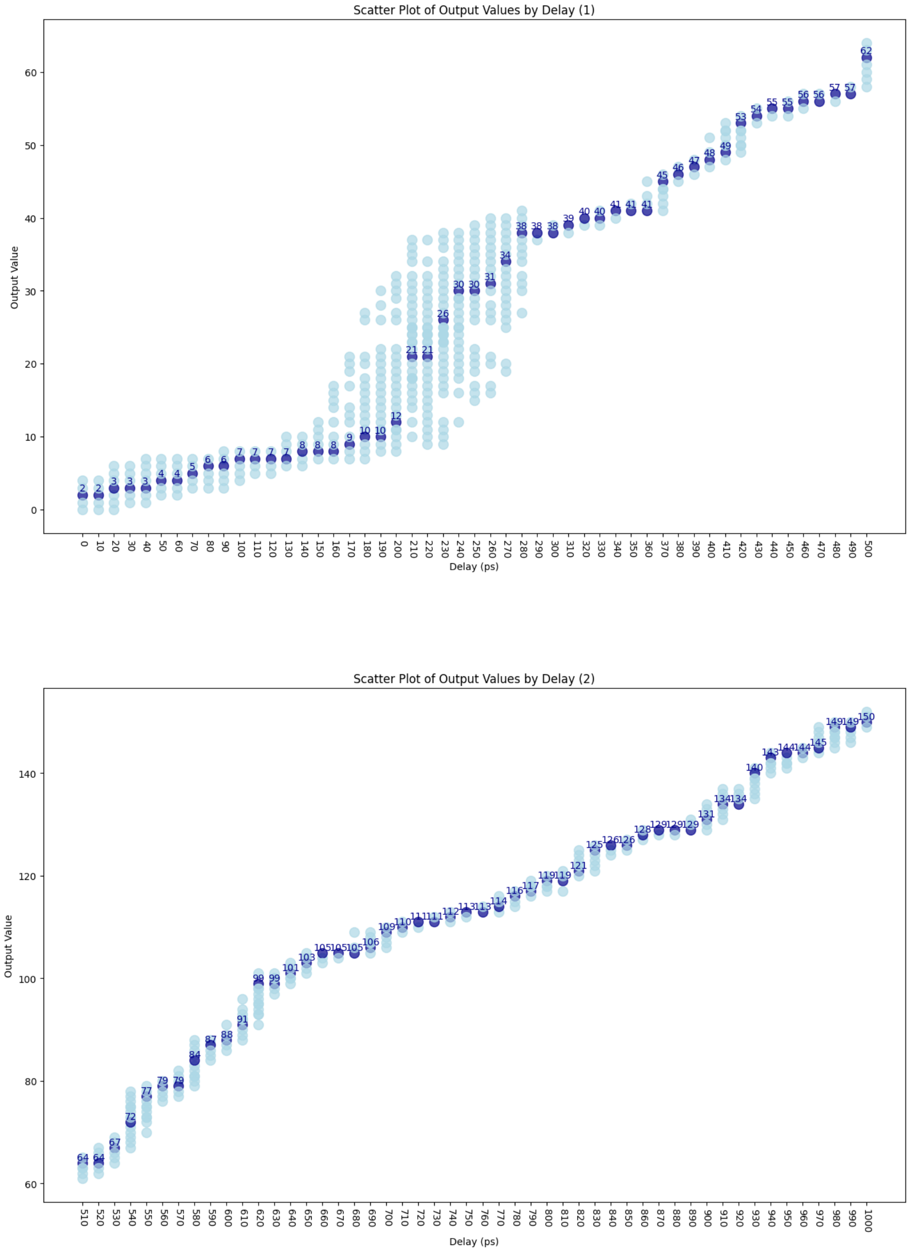


Figure 4.12: Delay Line Outputs Before Reordering - Part 1 and 2

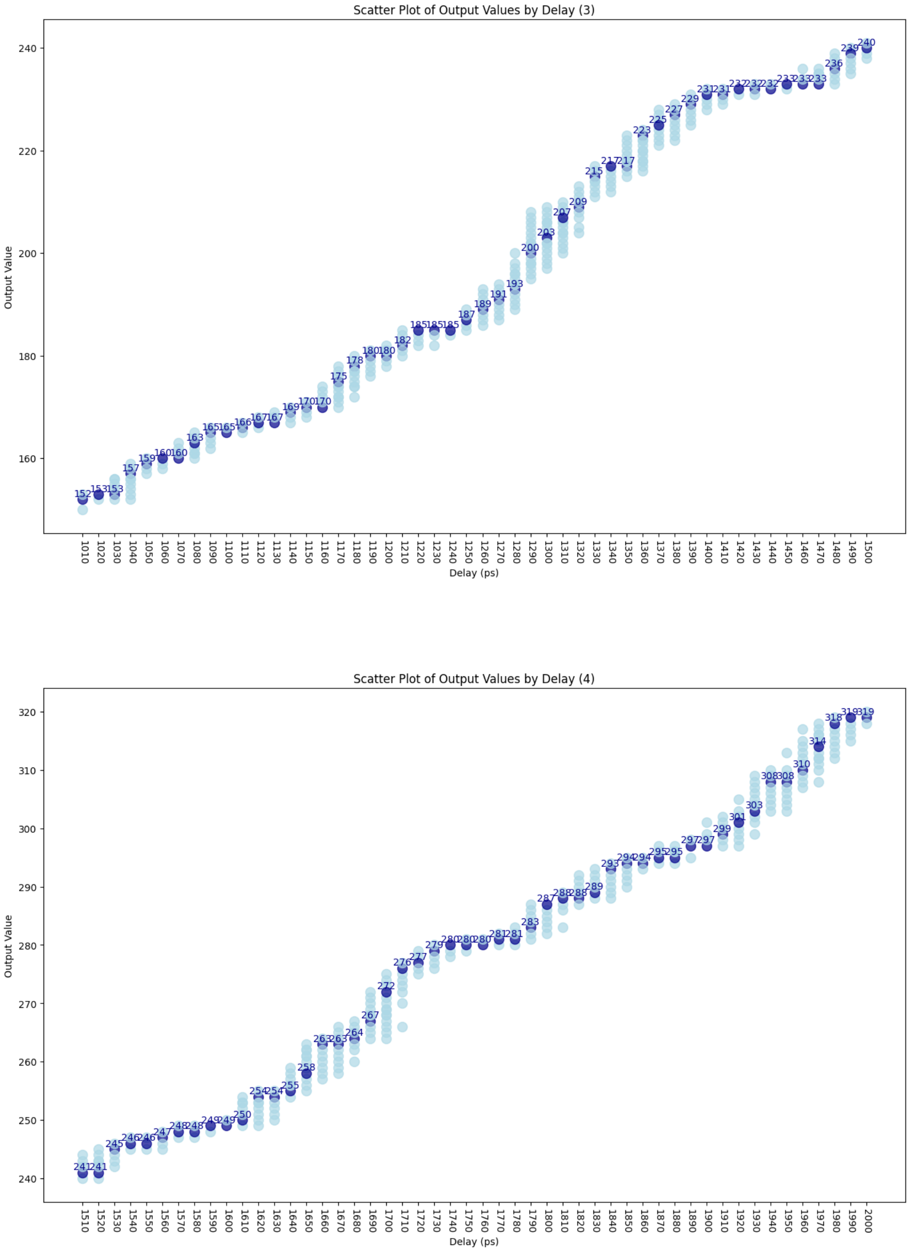


Figure 4.13: Delay Line Outputs Before Reordering - Part 3 and 4

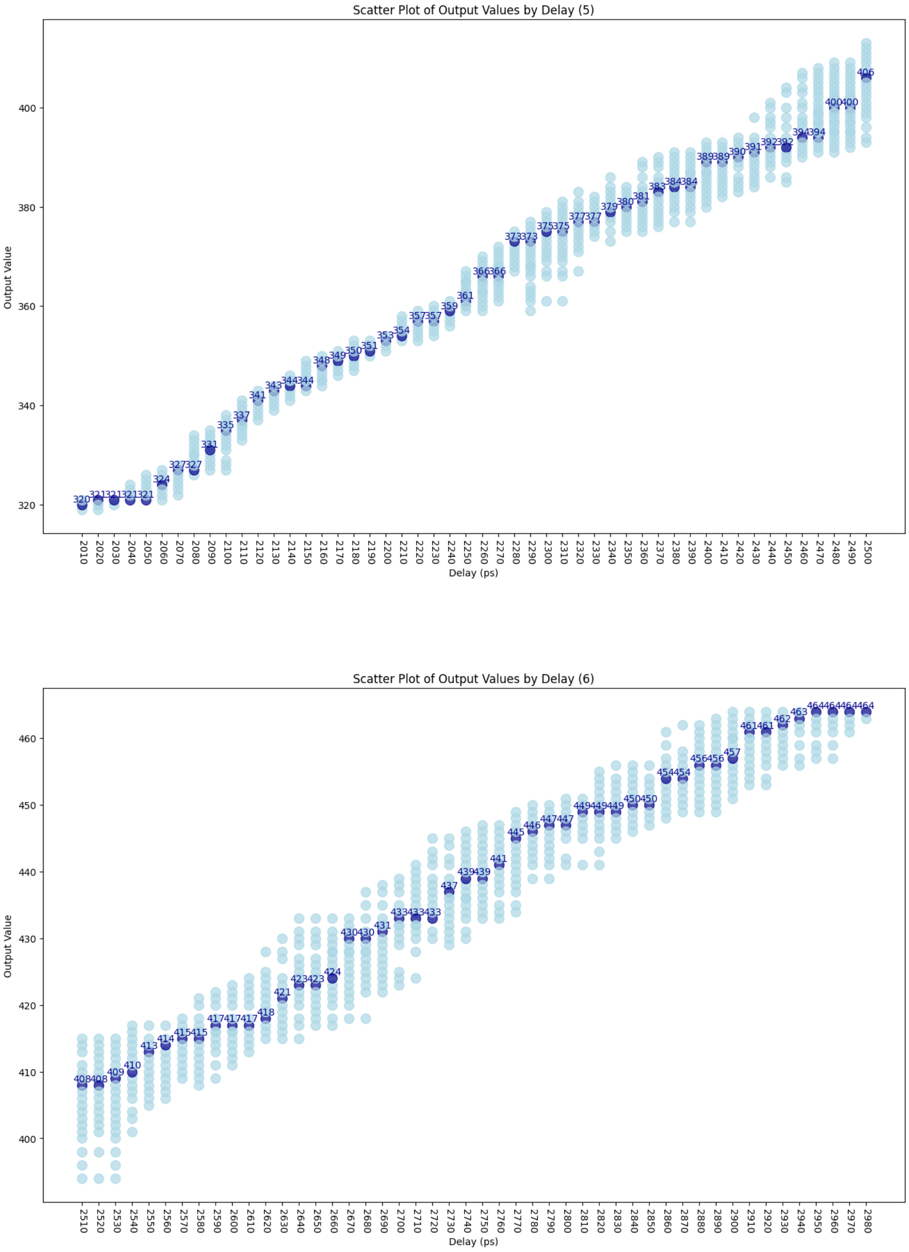


Figure 4.14: Delay Line Outputs Before Reordering - Part 5 and 6

To figure out if the reordering solution will disturb the functionality of the delay line or not, first it should be defined that if the changes in the register outputs always have the same order or its random. To do so, the previous test has been done multiple time and the results have been gathered, the order of changes in outputs has been extract by analizing the results, then they have been compared with each other all in software. In most cases the order of changes were the same and if they were different that wont change the delay line precision. after finding the actual order, the reordring method has been implemented in hard ware and the same test has been performed again.

The implementation of the reordering in hardware was simply done by adding a level of flip-flops, each connect to the previous latches but with the order that has been found. This additional layer may slow down the overall design, but if it allows the use of a more efficient encoder, it ultimately results in a faster Time-to-Digital Converter (TDC). This is because counting all 464 bits individually is significantly slower compared to dividing them into smaller groups and processing them in stages, as previously explained when thermtobin encoder was used and explained. The results of the test after reordering can be seen in figures 4.15,4.16,4.17.

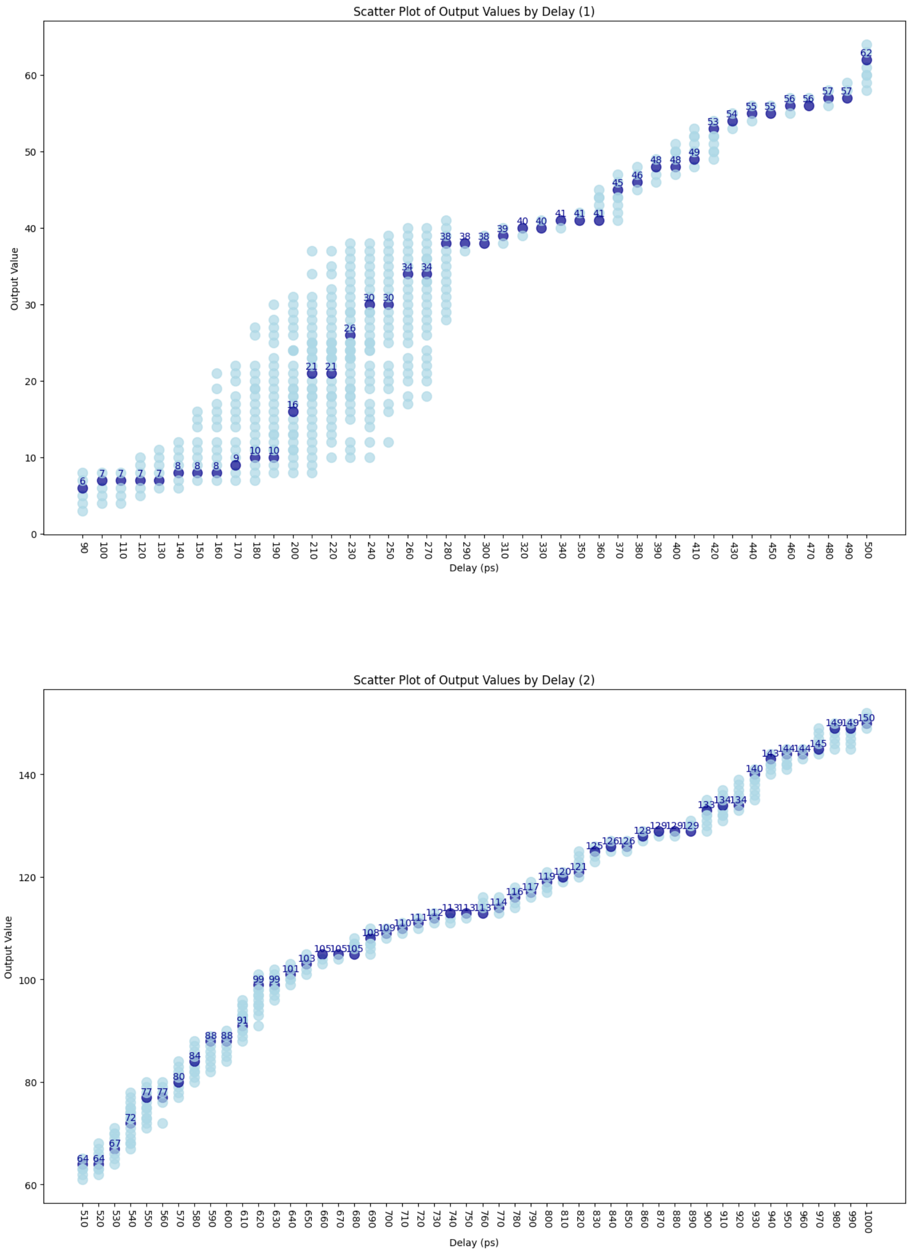
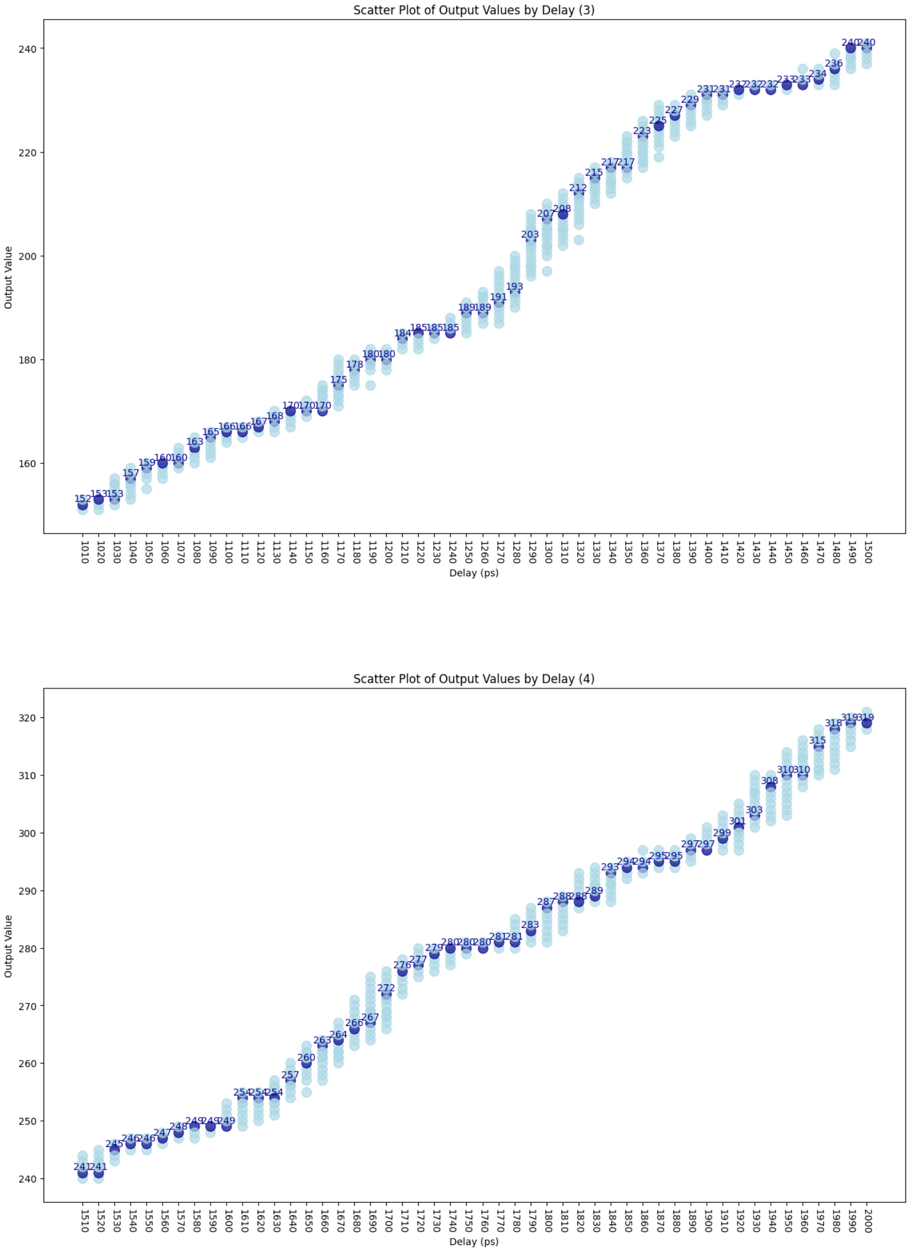


Figure 4.15: Delay Line Outputs After Reordering - Part 1 and 2



Scatter Plot of Output Values by Delay (3)

Scatter Plot of Output Values by Delay (4)

Figure 4.16: Delay Line Outputs After Reordering - Part 3 and 4

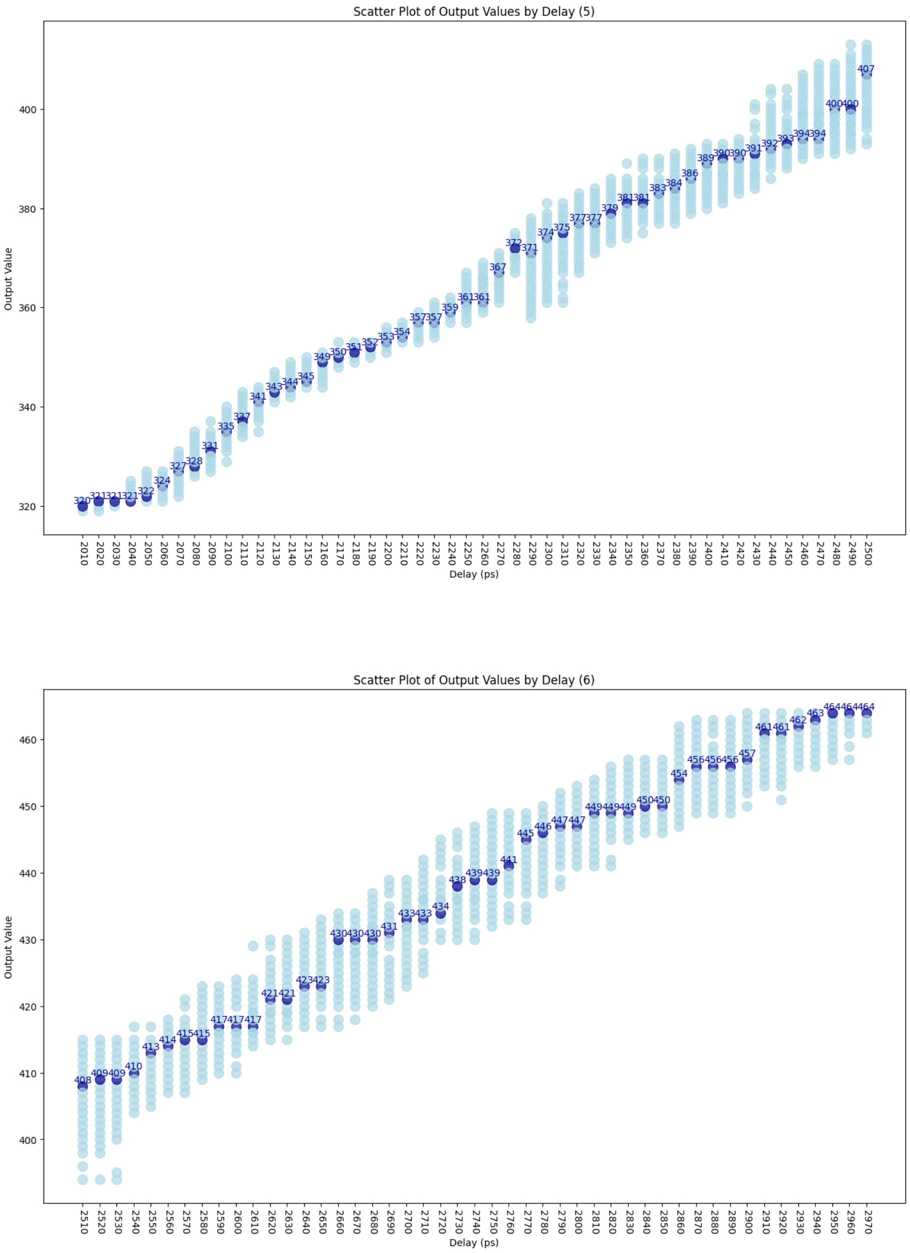


Figure 4.17: Delay Line Outputs After Reordering - Part 5 and 6

From the results, it has been observed that the reordering method did not change the delay line behavior and it could be used in the final version of the TDC. However, at this stage, this process has been halted and the focus has been shifted to other aspects, which was the implementation of the synchronous counter and the integration of the delay lines with the counter. Due to the fact that the outputs of the ESAABACUS was differencial, the design has been moved to different part of the FPGA so that it could be closer to the IOs, which supports differencial inputs. For this purpose, the FMC inputs of the board has been used. There are two FMC slots in KCU105 with specification mentioned in table 4.4. For useing the FMC inputs, an external module has been used which attached the FMC inputs of the FPGA (Figure 4.18) [10].

Connector	**FMC HPC Connector J22**
J22 HPC Connector Subset	- 116 single-ended or 58 differential user-defined pairs (34 LA pairs: LA[00:33]; 24 HA pairs: HA[00:23]) - 8 GTH transceivers - 2 GTH clocks - 2 differential clocks - 159 ground and 15 power connections
J2 HPC Connector Subset	- 68 single-ended or 34 differential user-defined pairs (34 LA pairs: LA[00:33]) - 1 GTH transceiver - 1 GTH clock - 2 differential clocks - 61 ground and 9 power connections

Table 4.4: FMC HPC Connector Overview

After the design has been moved and got fixed in another part if the FPGA, Due to different rounting, the same test has been done again with just the delay line.

It has been realized that the outputs were not exactly the same as the routing changed and the concept of haveing an initial offset for each delay line and the counter has been considered, which should be integrated in the last calibration stage. Also due to the fact that, the requirments of the project, is not to have one but multiple delay lines in one FPGA (multiple channel TDC), the same test has

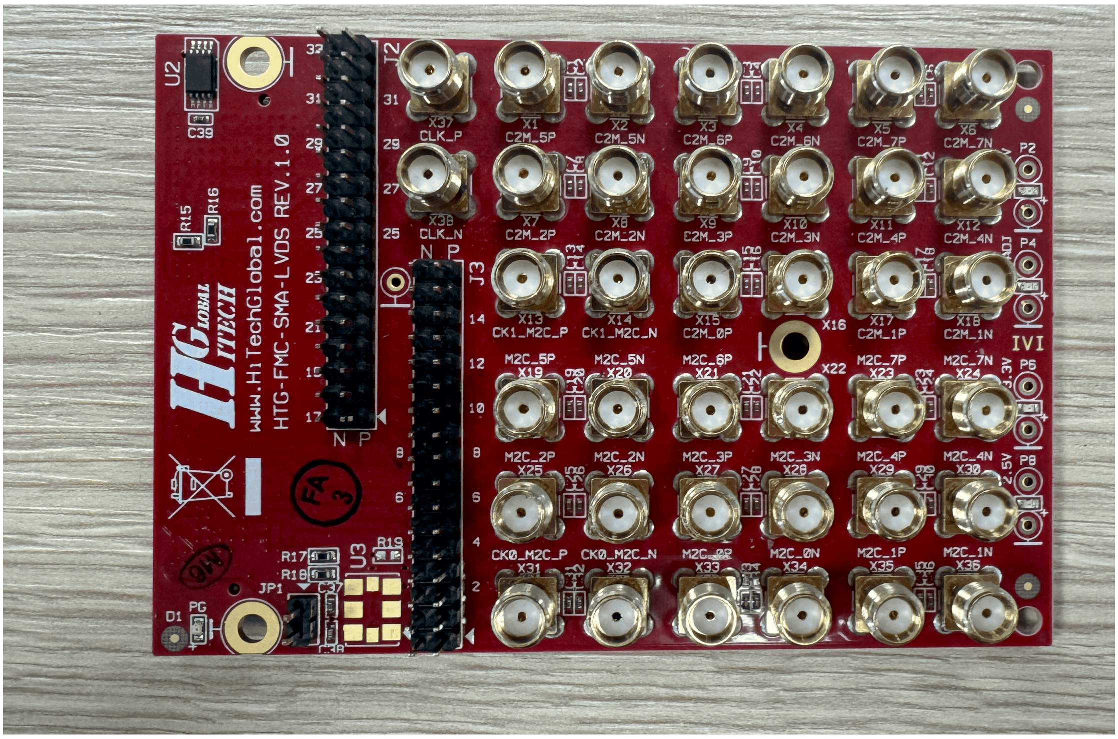


Figure 4.18: FMC Connector

to be done for each channel (contain two delay line and one counter) to be able to have a accurate TDC.

This offset was also detected in simulation 4.5, each time the delay was increased and the detectet offset was the same. the value is not reliable to be used for calibration, but it proves that the offset should be extract in physical tests.

Delay Line Configuration	Offset (ps)
1st DL (alone)	286
2nd DL (alone)	624
1st DL (in complete design)	218
2nd DL (in complete design)	390

Table 4.5: Delay Line Offset Values in Different Configurations

Another fact that has to be considered is which is very difinitive in the TDC presision is rising time and falling time of the clock (Metastability), which was seen also in timing simulation after implementation, that if the signal arrives after the rising edge of the clock, the delay line consider that rising edge and measure very small delay, instead of next rising edge which should be much larger value 4.19.

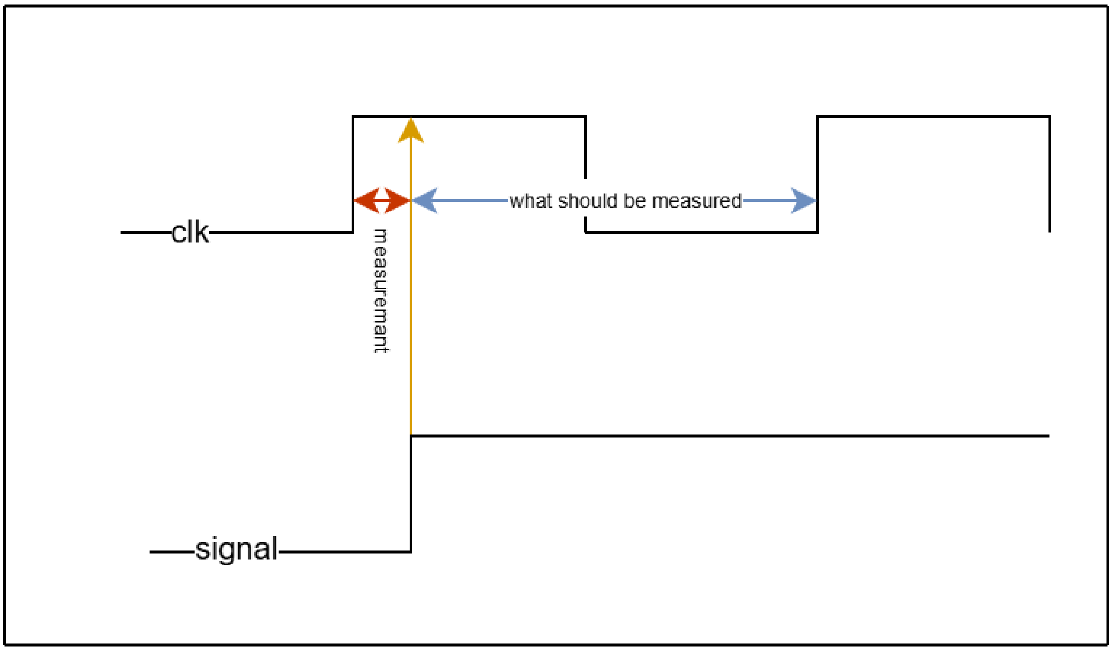


Figure 4.19: Error In Measurments

4.2 Counter:

The counter used in this design is a synchronous counter, which operates based on the inputs provided by the Time-to-Digital Converter (TDC). Specifically, the first input, referred to as the start signal, functions as an enable signal that initiates the counting process, while the second input, referred to as the stop signal, acts as a disable input that halts the counting process. This configuration ensures that the counter operates in sync with the signals from the TDC, allowing for precise time measurement.

It has been established that the maximum delay measurable by the delay line is approximately 3 ns. This limitation, As indicated by the timing diagram (see Figure 4.3), defines that the counter must operate with a clock period shorter than 3 ns to ensure accurate and reliable TDC. Therefore, a clock frequency of 400 MHz was selected for the synchronous counter, corresponding to a clock period of 2.5 ns. This frequency was chosen because the delay lines have been confirmed to measure delays of 2.5 ns effectively, without reaching their full capacity. This margin ensures that the system operates within a safe and reliable range for time delay measurement.

Additionally, the number of bits in the counter is a critical factor, as it defines the dynamic range of the TDC. The dynamic range represents the maximum measurable time interval. By selecting an appropriate output width for the counter, the TDC can achieve a wider dynamic range, allowing for longer time measurements.

4.3 bit counter:

Due to the fact that the implementation of the reordering was postponed, and the focus was on the main functionality of the TDC, a simple brute force bit-counter(behavioral implementation) with high delay was implemented, which still could meet the needs based on the incomming signal rates for the real test in final condition.

4.4 Encoder:

By leveraging the Block RAMs available in the FPGA, a highly efficient encoder has been developed to enhance the overall functionality of the system. This encoder is designed to translate the number of ’ones’ generated by the bit counter into a meaningful output that reflects the corresponding delay. The delay values are derived from extensive previous tests, ensuring that the encoder operates based on reliable and validated data.

The implementation of this encoder is significant because it introduces a streamlined process for generating output signals. Specifically, the encoder is capable of producing its output with a delay of one clock cycle, which, in this context, corresponds to 2.5 ns. This minimal delay is crucial for maintaining the timing integrity of the overall system, allowing it to effectively handle high-speed signal processing while ensuring accurate measurements and responses.

4.5 Final ALU:

The final Arithmetic Logic Unit (ALU), responsible for computing the ultimate delay from the outputs generated by the delay lines and the synchronous counter, has been implemented utilizing two levels of Digital Signal Processing (DSP) blocks. The first level of DSP blocks generates an output from the counter that can be adapted for addition or subtraction with respect to the outputs from the delay lines.

The second level of DSP blocks is specifically designed to perform the addition of the counter output to the delay line outputs. This two-tiered architecture enables the ALU to accurately compute the final delay between the two signals. As a result, the ALU effectively synthesizes the data from both the delay lines and the counter, yielding a precise measurement of the time interval between the input signals. This implementation ensures that the system can operate efficiently and accurately, meeting the stringent timing requirements of the application.

4.6 Memory:

In all our tests, the output data was not stored; instead, it was transmitted through the UART to a PC for analysis of the Time-to-Digital Converter (TDC) behavior. However, it is essential to adopt an efficient method for saving all data throughout the duration of the tests to prevent any loss or overwriting of critical information.

For memory storage, Block RAMs (BRAMs) were utilized, but the approach to saving data was not as straightforward as simply storing values and incrementing memory addresses. This method could lead to memory management issues and limit the duration of tests supported by the TDC, potentially resulting in truncated or incomplete datasets.

To effectively manage data storage, a histogram-based approach was implemented. This method involves organizing the output data into discrete bins, allowing for efficient aggregation of similar values. By categorizing the data in this manner, we can minimize memory usage while still capturing essential information about the signal characteristics.

Before implementing the histogram method, several parameters had to be configured. One of the most critical parameters is the bin size, which determines the range of values that each bin will represent. A smaller bin size provides higher resolution and more detailed data, but at the cost of increased memory usage and complexity in data processing. Conversely, a larger bin size simplifies memory management but may result in the loss of important detail in the signal data.

In addition to the bin size, other parameters such as the number of bins, the maximum and minimum values of the histogram, and the method for handling overflow (if the data exceeds the pre-defined range) were also established prior to data collection. By carefully setting these parameters, we can ensure that the histogram effectively represents the behavior of the TDC over the entire test duration.

This histogram-based storage approach not only optimizes memory usage but also enables a more efficient analysis of the data, allowing for a clearer understanding of the TDC's performance characteristics. By capturing the distribution of measured values, we can identify trends and anomalies that may be critical for further development and optimization of the TDC system.

The finla TDC schematic can be seen in figure 4.20. each time counter increament in represents 2.5ns, dus the final value has been multiplied by 2500 to convert the output into time in ps.

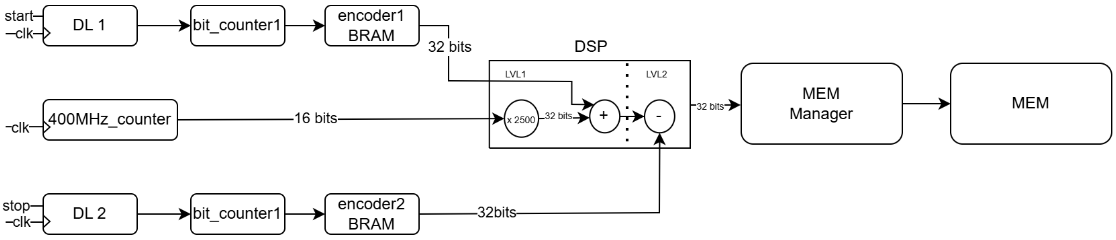


Figure 4.20: TDC Block Diagram

After Implementing both the delay lines and the counter, their outputs were send to PC through UART to observe their behavior seupratly while there were all implmented in FPGA to see any missbehavior. Due to the fact that there is not control over the occurance of the signals with respect to next rising edge of the clock, the range and the values of the delay lines was checked. In the other hand it was known that the output of the counter should be very close to the generated delay.

4.7 Verification of the configuration memory of the FPGA exposed to radiation:

As the design is going to used to measure TOF of the particles, the FPGA is going to be placed in radiation room, so one important step is to see the behavoir of the configuration memory under the radiation. To do so a dryrun has been made just to see if there are any scattered particles which can change the content of the configuration memory. Therefor the Board has been programmed and placed in the room under radiation with different positions and the readback of the configuration memory has been done in which basically the content of the configuration memory has been read time to time. The results shown if the board is going to be placed very close or directly under the radiation source, the configuration memory is going to be changed. Based on the results of this test the safe placement for the setup has been defined. [11], [12], [13], [14], [15], [16], [17], [18]

Chapter 5

Conclusion and Future Work

5.1 Testing the latest version

In the last test, the whole design was implemented upto memory part the the output of the TDC was has been tested. This test has not been done completley with enough reapitition, dus the results are not reliable and it was not possible to be able to make any conclusion and define the TDC metrics from them. It just mensioned here because it was the last test that has been done and most of the duartion spent on fixing and makeing the design to work and being able to get the results on PC. Eventhough a couple of the results can be seen in table 5.1 as an example.

Actual Delay (ps)	Most repeated Value (ps)
500600	498218(2382)
20001600	1999640(1960)

Table 5.1: TDC Analysis Table

The last stage has to be tested multiple time to be able to make conclusion based on the output and modify the design in case of any issue. If same behavior was detected, the calibration should be done to remove the offsets for the delay lines. Then the resolutiuon and precision of the TDC can be calculated based on the output of the TDC.

5.2 Conclusion

The implementation of the TDC on FPGA is a challenging task, dus due to the fact that the most critical part which was the Delay line has been implemented and tested and the results were acceptable, its not impossible. Eventhough the TDC measuring 3ns of time interwall was achived, it has been seen that even a small change in the placement or adding or removeing any other modules might effect the output and change the previous aquired results and due to the fact that there should be multiple channel TDC implemented in an FPGA, each step should be done for each channel. In this stage it could be extracted that how many channels could be implemented based on different type of input (LVDS, LVCMOS, . . .) in one FPGA. The delay due to routing might not effect most other designs as much as it effects the TDC, because any extra or less routing delay with even some ps difference, can effect the output of the TDC, leading to reduceing the precision.

5.3 Future Work

The next step it test and verify the TDC, with and without the memory, to varify the memory stage, also find out the maximum duaration of the data aquisition to not lose any data. Then that the calibration process should be done to remove the unavoidble offsets, and find out the metrics of the TDC on FPGA. After that implemention of the TDC with multiple channel and test the outputs of the each channel should be done, which not only needs all the previous steps for each channel, but also the outputs of them working together needs to be checked and verified (as mensioned every change might effect the outputs and the calibration). Also the effect of the temprature of TDC outsputs should be analiezed. The same tests should be done with TDC in ASIC to verify and compare the results and find out the advantages and disadvantages of the FPGA based TDC .

Appendix A

Test Instruments And Setup:

The instruments which have been used for the test are as following.

A.1 Oscilloscope:

For obseving the signals the KEYSIGHT DSOS254A has been used, Figure A.1.

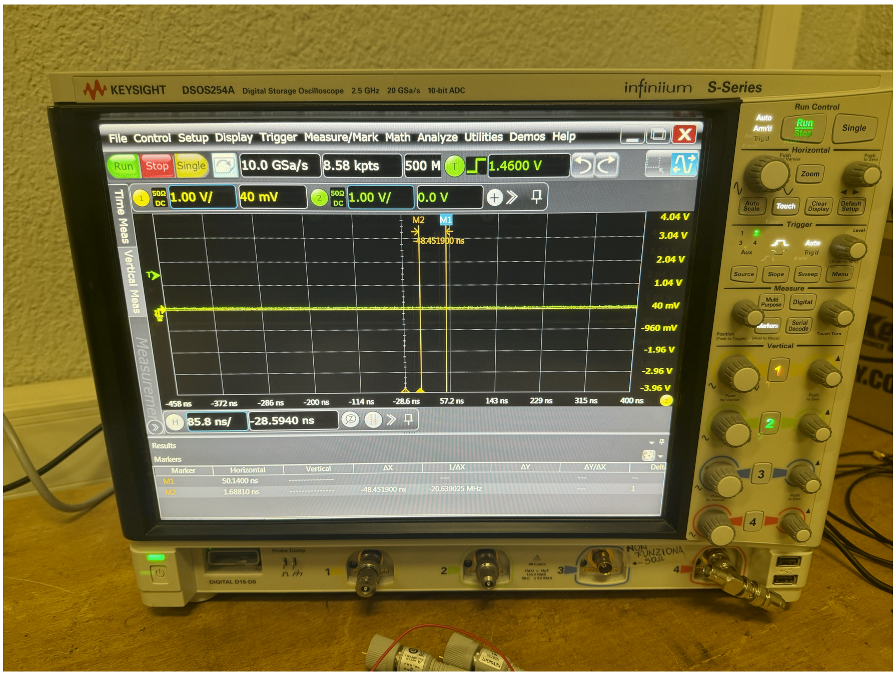

KEYSIGHT
DSOS254A Digital Storage Oscilloscope 2.5 GHz 20 GSa/s 10-bit ADC
infiniium S-Series
File Control Setup Display Trigger Measure/Mark Math Analyze Utilities Demos Help
Run Stop Single
10.0 GSa/s
8.58 kpts
500 M
1.4600 V
1.00 V/
40 mV
1.00 V/
0.0 V
85.8 ns/
-28.5940 ns

Figure A.1: Oscilloscope

A.2 Pulse Generator:

The PULSE RIDER PG-1072 has been used for tests with two outputs, Figure A.2.

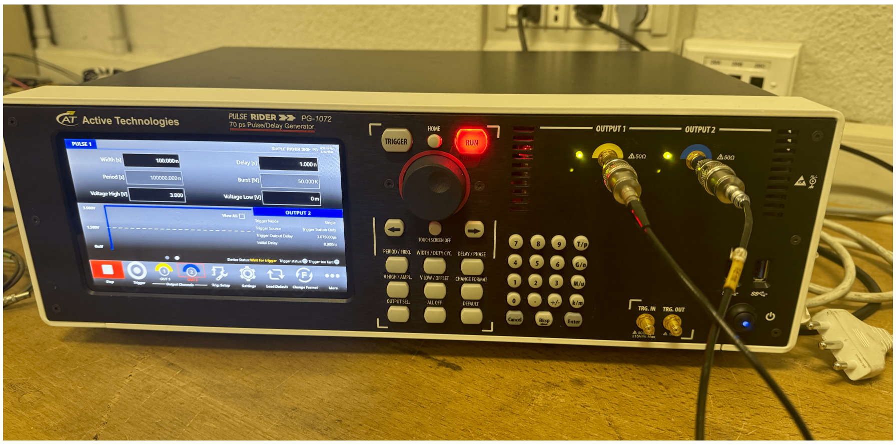


Figure A.2: Pulse generator

A.3 The last setup:

The last setup is shown in Figures A.3 and A.4, in which the TDC was connected to ESA ABACUS.

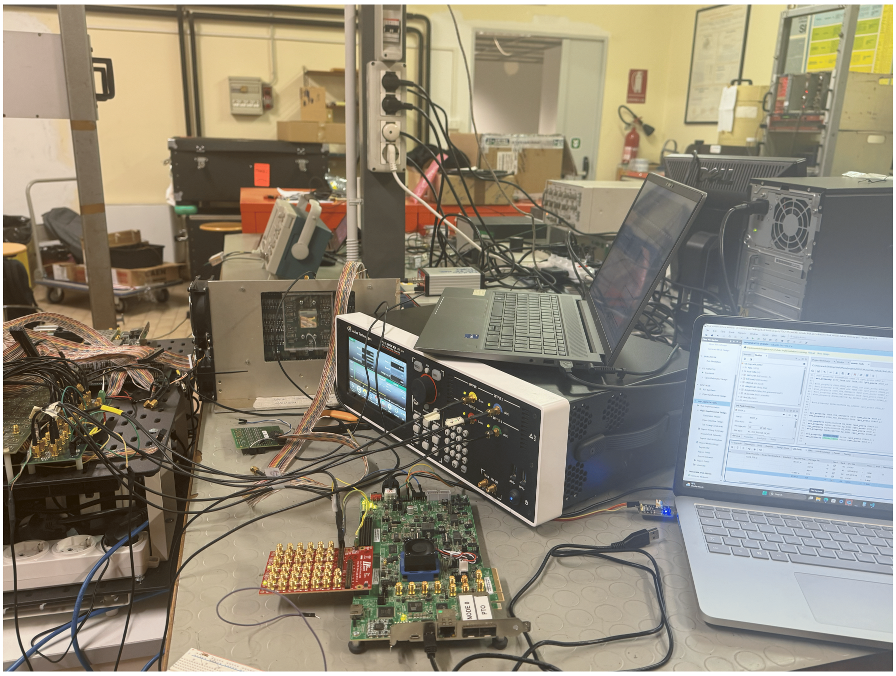


Figure A.3: The last setup 1

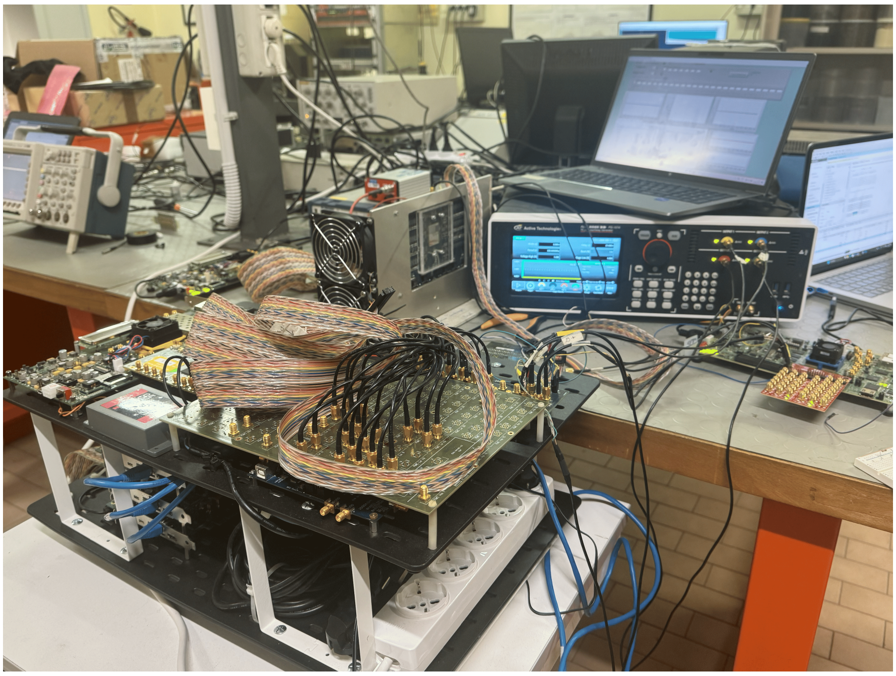


Figure A.4: The last setup 2

Bibliography

[1] E.M. Data et al. «A novel detector for 4D tracking in particle therapy». In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1068 (2024), p. 169690 (cit. on p. 12).

[2] Texas Instruments. *TDC7200 Time-to-Digital Converter for Time-of-Flight Applications in LIDAR, Magnetostrictive and Flow Meters Datasheet.* `https://www.ti.com/lit/ds/symlink/tdc7200.pdf` (cit. on p. 14).

[3] Samuele Altruda, Jorgen Christiansen, Moritz Horstmann, Lukas Perktold, David Porret, and Jeffrey Prinzie. «PicoTDC: a flexible 64 channel TDC with picosecond resolution». In: *IOP Publishing* 18.07 (2023), P07012 (cit. on p. 15).

[4] CAEN. *A5203B PicoTDC 32/64 Channel Time-to-Digital Converter.* `https://www.caen.it/products/a5203` (cit. on p. 16).

[5] Majid Memarian Sorkhabi and Siroos Toofan. «A high resolution, multi-path gated ring oscillator based Vernier Time-to-Digital Converter». In: *2011 Semiconductor Conference Dresden.* 2011, pp. 1–4 (cit. on p. 17).

[6] Fabio Garzetti, Nicola Corna, Nicola Lusardi, and Angelo Geraci. «Time-to-Digital Converter IP-Core for FPGA at State of the Art». In: *IEEE Access* 9 (2021), pp. 85515–85528 (cit. on p. 17).

[7] Poki Chen, Ya-Yun Hsiao, Yi-Su Chung, Wei Xiang Tsai, and Jhih-Min Lin. «A 2.5-ps Bin Size and 6.7-ps Resolution FPGA Time-to-Digital Converter Based on Delay Wrapping and Averaging». In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.1 (2017), pp. 114–124 (cit. on p. 17).

[8] Jinmei Lai, Yanquan Luo, Qi Shao, Lichun Bao, and Xueling Liu. «A high-resolution TDC implemented in a 90nm process FPGA». In: *2013 IEEE 10th International Conference on ASIC.* 2013, pp. 1–3 (cit. on p. 17).

[9] Seongheon Shin and Hyung-Joun Yoo. «A pipelined time stretching for high throughput counter-based time-to-digital converters». In: *2016 International SoC Design Conference (ISOCC)*. 2016, pp. 57–58 (cit. on p. 17).

[10] HiTech Global. *8-Port SMA / 34 Differential Pair FMC Module (Vita57.1) Datasheet.* `https://www.hitechglobal.com/FMCModules/FMC_SMA_LVDS.htm` (cit. on p. 34).

[11] Corrado De Sio, Sarah Azimi, Luca Sterpone, David Merodio Codinachs, and Filomena Decuzzi. «PyXEL: Exploring Bitstream Analysis to Assess and Enhance the Robustness of Designs on FPGAs». In: *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. 2023, pp. 1–4. DOI: `10.1109/SMACD58065.2023.10192116` (cit. on p. 41).

[12] S. Azimi, C. De Sio, A. Portaluri, D. Rizzieri, and L. Sterpone. «A comparative radiation analysis of reconfigurable memory technologies: FinFET versus bulk CMOS». In: *Microelectronics Reliability* 138 (2022). 33rd European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, p. 114733. ISSN: 0026-2714. DOI: `https://doi.org/10.1016/j.microrel.2022.114733`. URL: `https://www.sciencedirect.com/science/article/pii/S0026271422002578` (cit. on p. 41).

[13] S. Azimi, C. De Sio, and L. Sterpone. «Analysis of radiation-induced transient errors on 7 nm FinFET technology». In: *Microelectronics Reliability* 126 (2021). Proceedings of ESREF 2021, 32nd European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, p. 114319. ISSN: 0026-2714. DOI: `https://doi.org/10.1016/j.microrel.2021.114319`. URL: `https://www.sciencedirect.com/science/article/pii/S0026271421002857` (cit. on p. 41).

[14] Corrado De Sio, Sarah Azimi, and Luca Sterpone. «On the Evaluation of SEU Effects on AXI Interconnect Within AP-SoCs». In: *Architecture of Computing Systems – ARCS 2020*. Ed. by André Brinkmann, Wolfgang Karl, Stefan Lankes, Sven Tomforde, Thilo Pionteck, and Carsten Trinitis. Cham: Springer International Publishing, 2020, pp. 215–227. ISBN: 978-3-030-52794-5 (cit. on p. 41).

[15] E. Vacca, S. Azimi, and L. Sterpone. «Failure rate analysis of radiation tolerant design techniques on SRAM-based FPGAs». In: *Microelectronics Reliability* 138 (2022). 33rd European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, p. 114778. ISSN: 0026-2714. DOI: `https://doi.org/10.1016/j.microrel.2022.114778`. URL: `https://www.sciencedirect.com/science/article/pii/S002627142200302X` (cit. on p. 41).

[16] Eleonora Vacca, Corrado De Sio, and Sarah Azimi. «Layout-oriented radiation effects mitigation in RISC-V soft processor». In: *Proceedings of the 19th ACM International Conference on Computing Frontiers*. CF '22. Turin, Italy: Association for Computing Machinery, 2022, pp. 215–220. ISBN: 9781450393386. DOI: 10.1145/3528416.3530984. URL: https://doi.org/10.1145/3528416.3530984 (cit. on p. 41).

[17] Eleonora Vacca, Giorgio Ajmone, and Luca Sterpone. «RunSAFER: A Novel Runtime Fault Detection Approach for Systolic Array Accelerators». In: *2023 IEEE 41st International Conference on Computer Design (ICCD)*. 2023, pp. 596–604. DOI: 10.1109/ICCD58817.2023.00095 (cit. on p. 41).

[18] Eleonora Vacca, Sarah Azimi, and Luca Sterpone. «ZOR: Zero Overhead Reliability Strategies for AI Accelerators». In: *2024 22nd IEEE Interregional NEWCAS Conference (NEWCAS)*. 2024, pp. 248–252. DOI: 10.1109/NewCAS58973.2024.10666350 (cit. on p. 41).