High-Precision Time Measurement on FPGA: An Optimized TDC Approach

High-Precision Time Measurement on FPGA: An Optimized TDC Approach

Arash Amini Bardpareh[1], Eleonora Vacca[1], Corrado De Sio[1], Sarah Azimi[1], Luca Sterpone[1]
Elisa Fiorina[2], Emanuele Maria Data[2], Felix Mas Milian[2]
[1]*Dipartimento di Automatica e Informatica, Politecnico di Torino,* Turin, Italy
[2]*Istituto Nazionale di Fisica Nucleare,* Turin, Italy

***Abstract*—Time-to-Digital Converters (TDCs) are crucial for high-precision time measurements in applications like particle physics and medical imaging. While ASIC-based TDCs offer high accuracy, they lack flexibility. Implementing TDC on FPGA platforms faces significant challenges due to the digital nature of these devices. In literature, TDC designs are often presented in theoretical terms, with limited implementation details. Practical realizations frequently require adapting and refining the original concepts to match the specific device architecture and capability. The custom placement and routing constraints, or manual instantiation of primitive elements, are rarely discussed but are critical for achieving reliable and reproducible results. The omission of these aspects contributes to the gap between theoretical proposals and working hardware implementations. This work addresses this gap by presenting a fully implemented and FPGA-optimized TDC based on a tapped delay line (TDL) architecture, achieving 6 ps resolution and 30 ps precision on an AMD Kintex UltraScale FPGA, discussing detailed low-level optimizations, making it not only a high-performance solution but also a concrete reference for future implementations.**

Keywords—FPGA, Particle Therapy, Time-to-Digital Converter.

I. Introduction

Time-mode circuits are gaining significant attention due to their advantages in high-precision time interval measurements for applications such as LiDAR systems, medical imaging, digital communication, and particle physics [1][2]. When sub-nanosecond precision is required, Time-to-Digital Converters (TDCs) become essential, as they translate time intervals into digital values for further processing and analysis tasks. Traditionally, TDCs have been realized using ASICs to achieve high precision and performance. Despite their advantages, ASIC-based solutions are constrained by high development costs, extended design cycles, and limited post-fabrication flexibility. In response to these limitations, TDC, based on Field-Programmable Gate Arrays (FPGAs), has gained traction as an alternative, providing benefits such as reconfigurability, accelerated prototyping, and reduced costs. However, implementing TDCs on FPGAs presents unique challenges. FPGAs are optimized for sequential logic rather than precise timing measurements, making achieving picosecond-level precision difficult. Variability in logic elements, routing paths, jitter, and metastability introduces errors that must be carefully managed to provide consistent performance and results. Since the resolution of a TDC is determined mainly by the propagation delays of logic elements, which can vary significantly between FPGA devices and even across different regions of the same chip due to manufacturing process variations, extensive calibration and careful architectural optimization are essential to ensure accurate time measurements. The main contribution of the current work is the implementation of a TDC on an FPGA utilized for precise Time-of-Flight (ToF) measurement. The circuit is part of an FPGA-based data acquisition system (DAQ) setup used in the Hybrid online technology for particle therapy (HONEY) research project targeting charged particle therapy (CPT) for cancer treatment.

II. State of the Art

Most FPGA-based TDCs are implemented using the tapped delay line (TDL) method [3][4][5][6], which typically includes a delay line, a counter, and a thermometer-to-binary encoder. This architecture is popular due to its efficient use of FPGA resources. Resolution, precision, dead time, and linearity of a TDC strongly impact its overall performance. Despite various techniques that have been proposed to enhance these characteristics, such as dual-sampling [3], which improves resolution by capturing multiple measurements within a single clock cycle, or wave-union technique [4], which increases the number of effective delay taps to boost precision and linearity without excessive hardware complexity, there is still no methodology focusing on how placement and routing affect TDC behavior when implementing on a FPGA.

This work presents the implementation of an FPGA-optimized TDC designed to achieve high-precision time measurements for oncological particle therapy. The TDC is deployed on an AMD development board, leveraging available resources, with particular attention to I/O port constraints and routing requirements. Unlike previous works that mainly focus on calibration, our approach achieves optimal timing resolution by using a basic TDL (Tapped Delay Line) structure, while also carefully optimizing the physical placement and routing of internal logic blocks, especially with respect to their mutual positioning and their proximity to the I/O. We achieved 6 ps LSB and 30 ps of standard deviation on an AMD Kintex UltraScale FPGA [7], with a measurement range extending to several nanoseconds. The delay line supports up to 2,800 ps dynamic range, extended coupling with an integrated synchronous counter. The design has been implemented for both single-ended and differential inputs, making it adaptable to different input signals.

III. The Proposed TDC Architecture

A. Overview

In this section, we present the architecture of a TDL-based TDC implemented on an FPGA, comprehensively describing the design and implementation. The basic functionality of the TDC is to measure the time interval T between the arrival of two distinct signals, S1 and S2. The measure of T is derived from three other measurements, each performed by different functional modules within the architecture, using Eq. 1. The conceptual representation of these three measures is shown in Fig. 1.

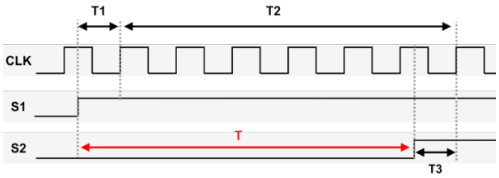

Fig 1. The measurement principle of the TDC architecture.

$$T = T_1 + T_2 - T_3 \quad (1)$$

The proposed two-channel TDC has two independent delay lines and a synchronous counter, as shown in Fig. 2. The delay lines are essential for achieving high-precision measurements in the picosecond range. In contrast, the counter operates at the system clock frequency, typically in the nanosecond range. In detail, T1 and T3 are measured using the two delay lines (DL1 and DL2), while the counter measures T2.

These delay lines capture the fraction of the clock cycle between a signal's arrival and the clock's nearest rising edge. In contrast, the counter starts counting complete clock cycles as soon as the first signal arrives and stops when the second signal arrives.

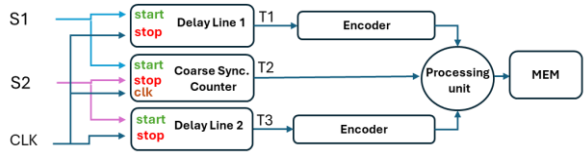

Fig 2. The proposed TDC architecture.

B. Tapped Delay Lines

The goal of the delay line is to measure the time interval between a start signal and a stop signal, with resolution on the scale of picoseconds. As shown in Fig. 3(a), the tapped delay line consists of a chain of logic elements, each introducing a small delay. The start signal propagates through this sequence of delay components, and each component incrementally adds delay to the signal propagation. The output of each delay element is connected to a register, controlled by the stop signal, to capture the state of the delay line. The status of these registers can be *filled* (sampled 1) or *empty* (sampled 0). Concatenating the status of these registers, we obtain a thermometric code. The filled registers are expected to increase linearly as the start signal propagates in the line.

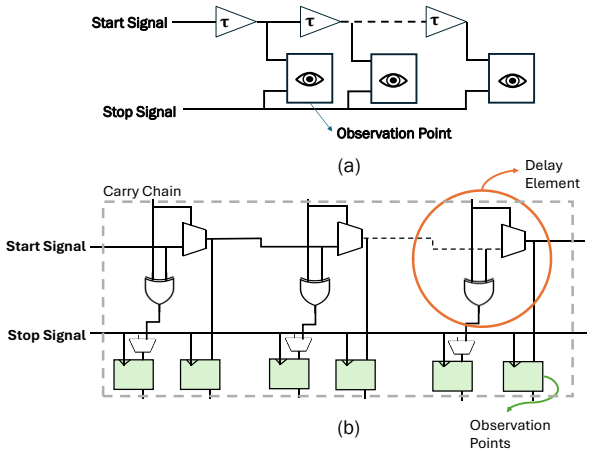

Fig 3. The tapped delay line architecture. In (a) the high-level schematic, while in (b) the correspondent FPGA implementation.

C. Counter

Even if the TDC targets measures in the scale of picoseconds resolution, based on the field it is used in, the required range of measurements can be much larger than this scale, and the delay line cannot be extended to cover the needed measurement ranges. A synchronous counter is employed to extend the measurement range beyond the range provided by the delay line. The counter operates in sync with the system clock, measuring time in discrete steps equal to the clock period. In time measurement between two signals (S1, S2), the counter begins counting when S1 is high at a clock-rising edge and stops when S2 is detected high at a subsequent rising edge.

D. Data Processing and Calibration

The thermometric output from the delay line represents the measured delay, but it must be converted into a timestamp for further processing because the raw bit pattern only indicates the number of active delay elements and does not correspond to a usable numerical time value. To address this, the encoder module includes a bit counter that counts the number of *filled* bits in the thermometric code. This count is then translated into a digital timestamp corresponding to the input delay. The calibration procedure was initially performed to empirically determine the relationship between the delay line output and corresponding time intervals. Two input signals were applied to the delay line with precisely controlled time offsets, incremented in 10 ps steps. This allowed accurate characterization of the delay introduced by each tap. Additionally, since the TDC is planned to be integrated into a standalone DAQ system, all computations are performed within the programmable device. Thus, to follow Eq.1, instead of using an external processor to compute the final measurement, a processing unit is used, and memory elements on the programmable logic are used to store the result.

E. Implementation Methodology

For implementing the delay line, the components that may be used to generate the delays are limited by the available logic resources within the device's architecture. Additionally, the propagation delay of the chosen logic defines the resolution, and as it is smaller, we have samples of the delay line with higher resolution. To achieve a minimal, regularly structured, and cascading delay element, we selected the carry blocks, interconnected as shown in Fig. 3(b). These blocks provide the smallest pseudo-regular delay in the device fabric, with small variations due to manufacturing process fluctuations, resulting in small variations in delay contribution within the picosecond scale. Additionally, they offer direct access to registers, which are fundamental to implementing the observation points in the delay line. However, we want to highlight that the distances between two carry logic elements may differ accordingly to their locations. For instance, the distance is noticeably larger between clock regions. Additionally, we empirically observed that the clock skew is less probable if the implementation is done in a single clock region. In our implementation, we used 58 CARRY8 primitives, manually mapped on specific carry chains of the architecture, which provide 464 taps. With custom placement constraints, we forced them to be aligned in a single column of resources and to fit in a single clock region. Indeed, using single column alignment allowed a small uniform interconnection delay. With this specification, the delay line, on an AMD Kintex UltraScale KCU105 device, saturates the observation points with an input of about 2.8 ns.

It is worth mentioning that TDC, in our case study, communicates with the front-end logic of sensors using LVDS signals with a 2 ns width. Since only specific I/O pins can be configured for LVDS, their selection significantly impacts the place-and-route phase. Additionally, to capture the 2 ns signal, additional latches are used since the system's clock period is larger than 2 ns. Each delay line receives the signal and measures the time difference between that signal and the next clock rising edge. It has been observed that the placement of the used logics for securing the input signals can also affect the delay line output, introducing additional offset that may reduce the delay line's effective measurement range, eventually failing to support the system clock frequency. With specific placement, a measurement range from 0 to 2.8 ns is achieved, which is the maximum range attained for each delay line.

Furthermore, in our design, the coarse counter is not continuously active but rather enabled and disabled by the input signals S1 and S2 to start and stop, respectively. The interaction between synchronous and asynchronous components must be carefully considered. In particular, we experimentally observed that a mix of synchronous and asynchronous internal components may lead to errors in the final value of the clock counter. We observed that, under specific timing conditions, the counter appeared to be off by an entire clock cycle, either overestimating or underestimating the actual value. This timing mismatch does not derive from functional errors in the hardware itself, but rather from asynchronous signals not being correctly sampled. Such issues must be carefully addressed when interfacing synchronous and asynchronous components, as they can significantly impact measurement accuracy, introducing a significant variability in the measurements, leading to an increase in the measurements' standard deviation independently of the intrinsic performance of the TDC.

Fig4. (a) illustrates the TDC outputs for several measurements, without specific placement constraints, and (b) shows the results after addressing this behavior by finding the optimal placement for each channel relative to one another. Even if there is no timing violation in design, the counter can be started or stopped at one clock edge early or late, producing an output that is ±1 clock period. Indeed, if the S1 enables the counter later or earlier than entering the DL1, or the same scenario for S2 to disable the counter and enter DL2, it will lead to wrong measurements of the passed clock cycle. We eliminated this behavior through iterative placement, co-locating the counter flops with the DLs entry cells and

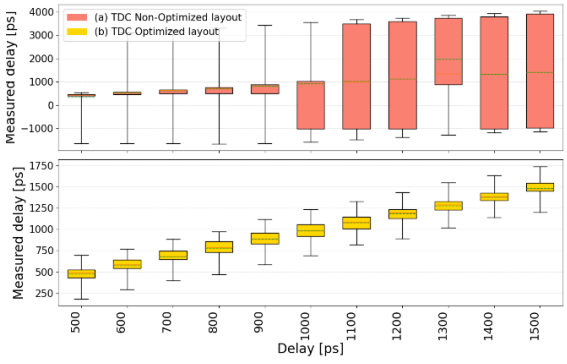

Fig 4. Delay line Input-Output characteristics with and without layout optimization.

equalizing the trace lengths from the package pins. The encoder is composed of a bit counter and a *Map*, which serves as a bit count-to-time converter. The bit counter analyzes the delay line output, counting the number of logic '1' values among the taps. The count of 1s indicates how far the signal has propagated through the delay line before being captured. The output of the bit counter is fed into the *Map*, which maps the counted number of 1s into the corresponding time value. The contents of the *Map* and the mapping from the bit counter output into the timestamp were determined through an extensive calibration process. During calibration, the widths of the bins associated with the taps, as shown in Fig. 5, are measured by testing the delay line with numerous input signals. This process allows each bit counter output state to be accurately mapped to its corresponding delay value. The process of counting ones could be done efficiently by finding the place of the last 1 in the line and using the index to count them, but in the real scenario, the 1's in the delay line do not propagate linearly as expected. Once the state of the DLs is captured, due to violations in setup or hold time, a problem known as *bubbles* may, and often, occur. For example, if "11110000" is expected, the output would be "11100011". To overcome this issue, the bit counter is implemented using LUTs and fabric adders in multiple stages. In each stage, multiple blocks function in parallel to exactly count the bits and comply with the system's measurement pace. The Encoder's Map has been implemented with BRAM. The final placement of the device is shown in Fig. 6.

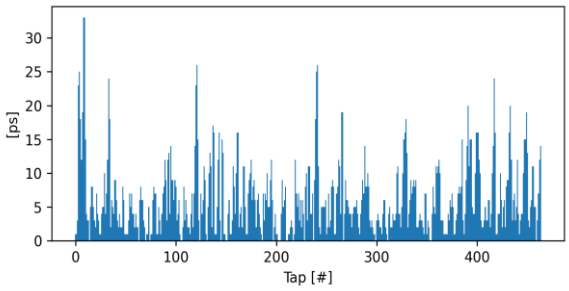

Fig 5. The bin width of the Delay Line

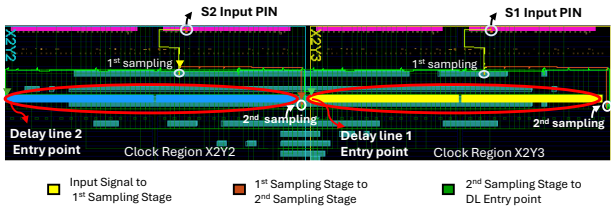

Fig 6. Proposed layout solution of the implemented TDC, showing the custom symmetry-oriented place & route of the two delay lines.

IV. Experimental Validation

TDC functionality has been evaluated as part of the HONEY (Hybrid Online technology for particle therapy) research project. In charged particle therapy (CPT), where protons target tumors while minimizing damage to healthy tissues, precise monitoring of beam parameters and particle range is essential. As primary particles pass through the tumor, they release energy while generating secondary particles. Measuring the Time of Flight (ToF) and energy of these particles provides crucial data on the tumor and efficiency of the treatment. To achieve this, the TDC is dedicated to

measuring (i) time intervals between consecutive primary particles and (ii) time intervals between primary and secondary particles.

A. Hardware Setup, Design, and IO Optimizations

The TDC is implemented on the AMD KCU105 FPGA development board. A key design requirement in our case study is the interfacing of the proposed TDC with the ESA-ABACUS [8] board, which employs Low Gain Avalanche Detectors (LGADs) for high-energy particle detection and outputs digital signals in the LVDS25 standard. However, most modern FPGAs do not support LVDS25, featuring only a limited number of I/O pins using LVDS18. Consequently, an external conversion board is required to translate LVDS25 signals to LVDS18 for compatibility with the FPGA.
The proposed TDC operates at 400 MHz; post-implementation details are provided in Table I.

Table I. Post-implementation details of the proposed TDC.

Resource	Available [#]	Utilization [#]	Utilization [%]
LUT	242,400	1,459	0.6
REG	484,800	3,292	0.68
CARRY8	30,300	285	0.94
BRAM	600	1	0.17
DSP	1920	1	0.05

B. Experimental Results

For the validation process, the TDC was tested using pairs of signals (S1 and S2) generated by a pulse generator. The signals had a width of 2 ns and followed the LVDS18 voltage standard. Measurements were taken at increasing delays, starting from 150 ps and advancing in 50 ps steps up to 4,000 ps. At each delay, 100,000 signal pairs were generated and measured. In Fig. 7, the TDC input-output characteristic is reported. The plot reports the average measured delay between channels for various T, and the standard deviation of measurement caused by jitter, system noise, and errors as error bars. We analyzed the distribution of measurements for each T. Both, the mean and median of the measurements approach the input values with a maximum error of about 25 ps, and an average error of about 17 ps for the mean and 11 ps for the median. The standard deviation varies from 30 to a maximum of 100 which also includes the disturbance introduced by external board, cables, and noise.

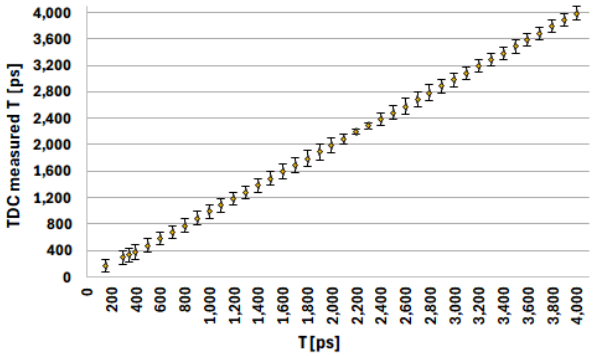

Fig 7. TDC input-output characteristics curve

For instance, Fig. 8 shows the result of measuring 2200 ps in 100,000 measurements.

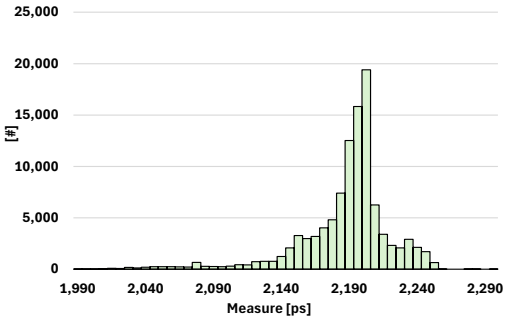

Fig 8. Distribution of 100,000 TDC measurement of 2,200 ps delay

V. Conclusions

We proposed an FPGA-based TDC as a solution for real-time time-of-flight measurements, comprehensively providing implementation details and choices. The experimental results validate its performance, making it suitable for use in particle therapy equipment. This work highlights FPGA-based TDCs as a promising alternative for cost-effective, high-accuracy time measurement in various fields.

Acknowledgment

This research is part of HONEY (Hybrid ONline technology for particle therapy), a PRIN (Progetti di Ricerca di Interesse Nazionale) project funded by the Italian Ministry of University and Research (MIUR), CUP number I53D23001280006 . The project is conducted in collaboration with Politecnico di Torino, INFN, and the University of Torino, with valuable support from CNAO (National Center for Oncological Hadron Therapy).

References

[1] V. Sesta, A. Incoronato, F. Madonini, F. Villa, "Time-to-digital converters and histogram builders in SPAD arrays for pulsed-LiDAR," Measurement, Volume 212, 2023, 112705, ISSN 0263-2241.

[2] S. Russo, N. Petra, D. De Caro, G. Barbarino, A. G.M. Strollo, "A 41ps ASIC time-to-digital converter for physics experiments," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 659, Issue 1, 2011, Pages 422-427, ISSN 0168-9002, doi:https://doi.org/10.1016/j.nima.2011.08.031.

[3] Jinmei Lai et al., "A high-resolution TDC implemented in a 90nm process FPGA," IEEE 10th International Conference on ASIC, Shenzhen, 2013, pp. 1-3.

[4] C. Liu, Y. et al, "A 3.9 ps RMS resolution time-to-digital converter using dual-sampling method on Kintex UltraScale FPGA," 2016 IEEE-NPSS Real Time Conference (RT), Padua, Italy, 2016, pp. 1-3.

[5] F. Garzetti et al., "Time-to-Digital Converter IP-Core for FPGA at State of the Art," in IEEE Access, vol. 9, pp. 85515-85528, 2021.

[6] M. Fishburn et al.,"A 19.6 ps, FPGA-Based TDC With Multiple Channels for Open Source Applications," in IEEE Transactions on Nuclear Science, vol. 60, no. 3, pp. 2203-2208, June 2013.

[7] Xilinx, KCU105 Evaluation Board for the Kintex UltraScale FPGA: User Guide, UG917 (v1.7), May 2023. [Online]. Available: https://www.xilinx.com/support/documents/boards_and_kits/kcu105/ug917-kcu105-eval-bd.pdf

[8] E.M. Data, et al," A novel detector for 4D tracking in particle therapy, Nuclear Instruments and Methods" in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment.