A novel synchronizer for a 17.9ps Nutt Time-to-Digital Converter implemented on FPGA

Rui Machado
ALGORITMI Center
University of Minho
Guimarães, Portugal
id6010@alunos.uminho.pt

Luis A. Rocha
CMEMS
University of Minho
Guimarães, Portugal
lrocha@dei.uminho.pt

Jorge Cabral
ALGORITMI Center
University of Minho
Gumarães, Portugal
jcabral@dei.uminho.pt

Abstract— **The evolution of Field-Programmable Gate Array (FPGA) technology triggered the appearance of FPGAs with higher operating frequencies and large number of resources. Simultaneously, the evolution of the FPGAs design tools has simplified the development process, reducing the time to market. These factors made FPGA platforms attractive for several applications, including time-of-flight applications that require the implementation of Time-to-Digital Converters (TDC). This work presents a Nutt TDC, based on a coarse counter and a Tapped Delay Line, with 17.9 picoseconds resolution and 5.4 LSB differential nonlinearity (DNL), implemented in a Xilinx Zynq-7000 FPGA, to be used on LiDAR applications and pull-in time measuring in MEMS accelerometers systems.**

Keywords— Time-to-Digital Converter (TDC), Field Programmable Gate Array (FPGA), Time Measurement, Counters Synchronization.

I. INTRODUCTION

Time-to-Digital Converters (TDC) are widely used for time measurement, being of extremely importance in time-of-flight (TOF) applications [1], where sub-nanosecond resolutions are required [2]–[5]. Besides that, TDC are also used in phase-locked-loops (PLL), LiDAR systems, time-over-threshold (TOT) measurements, analogue-to-digital converters (ADC) [6], and positron emission tomography (PET) applications. Both Application Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA) technologies can be used to implement TDC with high resolutions around 3 to 30 picoseconds [2], [7]. The ASIC offers more flexibility on the architecture design while the FPGA, although being limited due to the FPGA logic blocks configuration, enables a faster and much cheaper development process. Recently, a considerable amount of research effort has been made to enhance FPGA based TDC measurement resolution [7]. Since the available resolution is usually limited not by the minimum cell delay presented in the FPGA, but by the difference in the homogeneity of the cells delays, due to process, voltage and temperature (PVT) variations, most of these efforts focus on the improvement of the measurement linearity. Other works [5], [8] presented some novel architectures aiming to improve FPGA based TDC resolution beyond the intrinsic minimum cell delay, although a calibration process is always required to reduce the non-linear delay errors.

The implementation of a TDC architecture, based on a single TDL to measure both the rise and falling edge of the input signal on FPGA, is proposed in this paper. The proposed TDC enables the time measurement between the rise and falling edges in a single input channel with a 17.9 picoseconds resolution, maximum DNL of 5.4 LSB and an INL error between -2.5~5.8 LSB. When using a fine TDC and a coarse counter, the synchronization of the two values is required. A novel synchronizer for metastability errors correction, when the input signal arrives close to the system clock rise edge is also proposed.

The remainder of this paper is structured as follows: Section II starts by briefly describing several FPGA based TDC architectures found on the literature. Section III presents the proposed TDC architecture design along with the implemented synchronization system. The most common calibration techniques, used to improve TDC linearity, are also addressed. In Section IV, the proposed TDC results are presented and assessed with the current state of the art. Finally, in Section V, the main conclusions are drawn and some future research paths regarding FPGA based TDC are proposed.

II. STATE OF ART

The evolution of FPGA technology in recent years have enabled its application on a varying range of applications where in the past, only ASIC solutions prevail. With FPGA being produced in 40nm technology and below, the logic cells delay is decreased to few picoseconds, making FPGA an attractive platform for TOF applications. In recent years, multiple works implementing TDC on FPGA were reported in the literature. These works focus essentially in three different architectures: Tapped Delay Lines (TDL); Phased Clocks; and Ring Oscillators. In the remainder of this section, the most important contributions regarding each of the aforementioned architectures will be presented.

A. Tapped Delay Lines

The tapped delay line is the most common architecture used to implement TDCs on FPGAs [1], [9]. The basic operation principle is delaying the input signal across a series of logic cells, carry cells in the case of FPGAs, which define the minimum achievable resolution [4], [7], [8], [10]. The state of the delay chain is sampled at each system clock rise edge, resulting in a thermometer code which stores the fine timestamp information. Fig. 1 depicts the described TDL architecture.

The work from Wang and Liu [8] presents an architecture with 4.2ps resolution and an Integral Non-Linearity (INL) of approximately 2 LSB. To improve the TDL linearity, at the cost of the achievable resolution, a decimation calibration methodology was implemented. In [11], a TDC based on multiple TDL was implemented on a Virtex-7 FPGA, achieving a resolution of 3.5ps and 3.5 LSB of Differential Non-Linearity

978-8-8872-3740-5©2018 AEIT

(DNL).

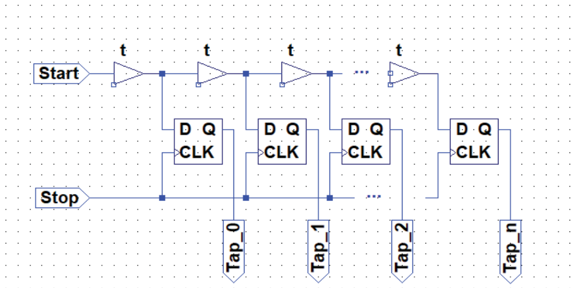

Figure 1 - Tapped Delay Line Architecture (adapted from [8])

B. *Phased Clocks*

Phased Clocks TDC architecture is based on multiple clocks with the same period but shifted in phase to sample the input signal and obtain a fine measurement of time [10], [12]. The non-linearity and resource utilization of this type of TDC is much lower, since modern FPGAs already have integrated PLL blocks, capable of generating multiple clock sources with minimum jitter values. Fig. 2 depicts the basic structure of the Phased Clocks TDC.

Variations of this architecture were implemented in [12] and [13], achieving a 56.6ps and a 1ns resolution, respectively.

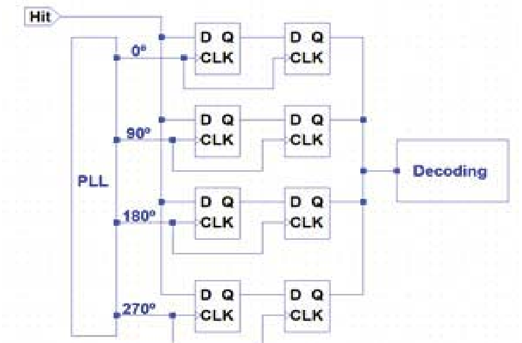

Figure 2 - Phased Clocks Architecture (adapted from [13])

C. *Ring Oscillators*

The ring oscillator topology is based on a dual oscillator architecture with slightly different oscillation frequencies [3], [14], [15]. The input signal starts the slow oscillator while the stop signal, generally the system clock, starts the fast oscillator. A counter is incremented until the two oscillators are in phase, being the value of the counter at that point the fine measurement value. The block diagram of this topology is presented in Fig. 3.

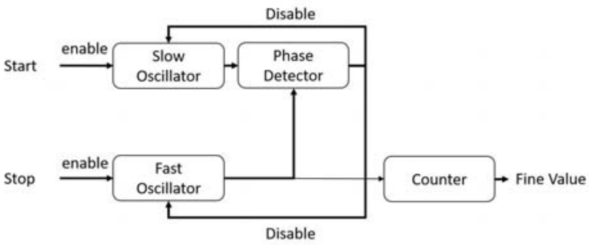

Figure 3 - Ring Oscillator Architecture (adapted from [14])

The work presented by Abbas and Khalil [14] successfully implemented this architecture on a Spartan3AN FPGA with a 23ps resolution and 10ps INL. In [3], another variant of this architecture was implemented in a Stratx III board with a resolution of 31ps.

III. TDC Architecture and Implementation

The implemented TDC is based on a tapped delay line topology, and it was devised to be easily replicated and implemented in TOF, LIDAR, and Pull-in time measurement applications.

The TDC comprises two counting stages, one to do fine measurements using a tapped delay line, which achieves resolutions higher than a clock cycle, and another stage implemented using a coarse counter that keeps track of the system clock ticks, to improve the measurement range. The architecture proposed is presented in Fig. 5, and in Fig. 6, a more detailed view of the synchronizer block is given. The use of two counting stages, fine and coarse, increases the measurement range, but introduces a synchronization issue between counters that must be addressed, in order to assure a correct measure. The output from the TDC is connected to an asynchronous FIFO memory, responsible for storing the measured values, and for safely cross these values to the ARM AXI bus domain. The TDC throughput is dependent on the value to be read and the maximum dead time is equal to one clock cycle. The principle of operation is as follows: upon the arrival of a rise transition on the input signal, the *Hit* signal on Fig. 5, the carry chain delay line elements start to change their state. These values, referenced on the remainder of this document as thermometer code, are sampled by the system clock on the following cycle; simultaneously, the *rise* signal is generated synchronously with the system clock. On the following clock cycle, the chain is sampled again by the system clock. As the *rise* signal is at logic level 1, the previous value is stored in a set of registers called *Sample start thermometer* in Fig. 5. The implemented dual sampling registers technique reduces metastability issues and bubble problems on the sampling stage. Furthermore, since the rise edge signal is only enabled during one clock cycle, the sampled code value remains stable during the entire decoding process. An analogous process occurs when the input signal transits from high to low. The *fall edge* flag, which is generated after a fall transition of the input signal, is also responsible for storing the value from the coarse counter. Finally, the three measurement values are merged in a single 32-bit value and an end-of-conversion flag is generated. A time diagram, presenting a full sampling operation, is given Fig 4.

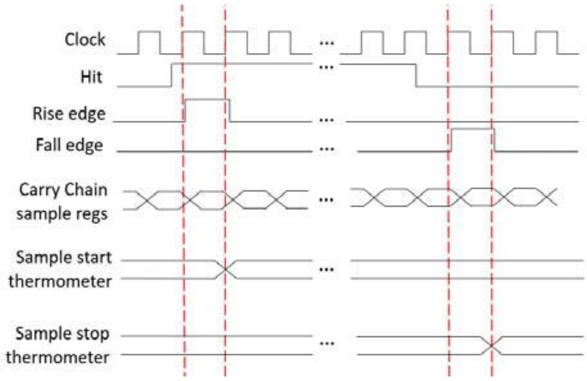

Figure 4 - Start and Stop sampling time diagram

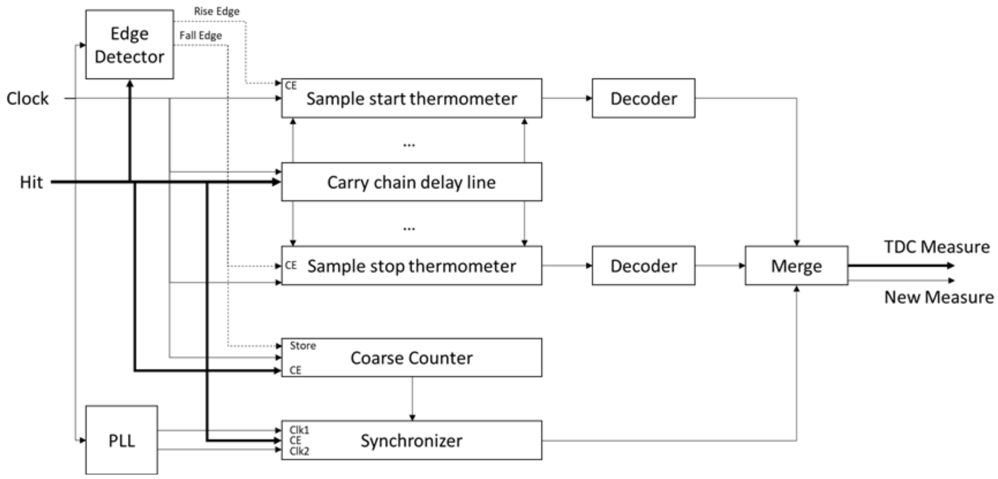

Figure 5 - Proposed TDC Architecture

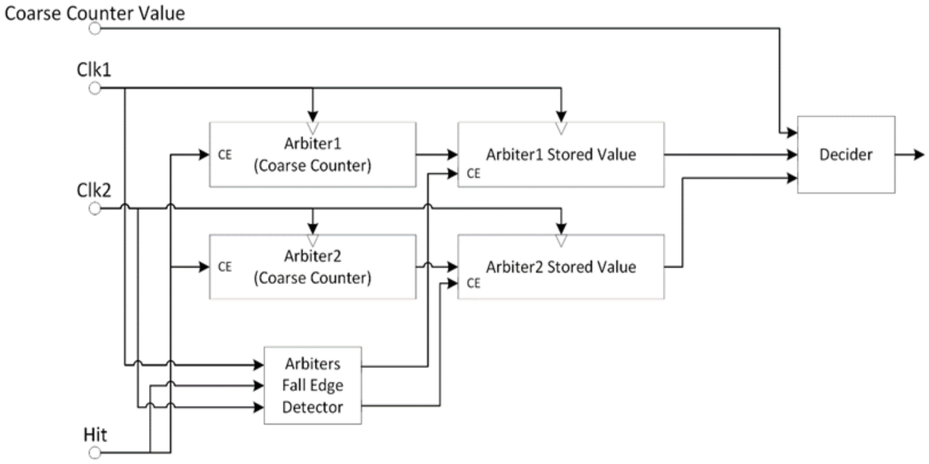

Figure 6 - Proposed Synchronizer Block Diagram

The presented TDC architecture was implemented on a ZYBO development board that includes the ZYNQ XC7Z010-1CLG400C FPGA. The board also has a dual-core ARM Cortex A9 processor. The FPGA in the development board is composed by several configurable logic block (CLB) connected to switch matrix boxes for general routing. Each CLB contains two slices with logic-functions generators, storage elements, carry logic and wide-function multiplexers. The fact that carry logic blocs have a dedicated route to the immediately above CLB, make them the most suitable element to use when implementing a tapped delay line in a FPGA. An overview of the described carry logic path is depicted in Fig. 7.

Implementing a TDC in a FPGA entails a set of practices that are not usually used in the traditional FPGA development flow. As good linearity is crucial to develop a TDL and as the delay of each tap of the TDL is defined by the carry cell and storage element associated, the positioning of these elements must be set with caution, and usually, the automated tools do not perform the best positioning. Moreover, typical HDL code can be simplified by the programming tools, leading to a TDL that does not resemble the one designed. Even direct instantiation of the primitives is subjected to simplifications, and thus, a way to avoid these simplifications is required. In the case of Vivado 2017.4, the compile option *dont_touch = "TRUE"* is enough to ensure that the instantiated primitives will remain in the final bitstream.

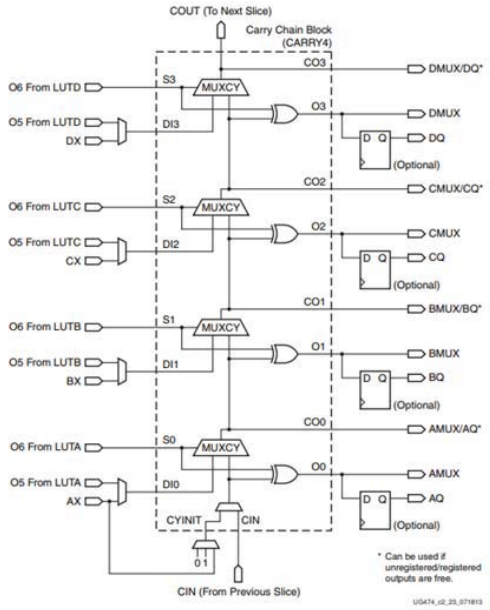

Figure 7 - Base CLB Structure

As FPGA technology evolves and the cell's propagation delay decreases, the influence of paths' propagation delays and clock skews can no longer be neglected. Without proper placement of the delay line elements, serious bubble problems can appear on the thermometer code outputted by the registers of the delay line, making its decoding impossible. Therefore, a set of constraints has to be specified to address this issue. Our practical experience suggests that the delay line should be placed in the slices next to the clock buffer, and the storage element of each tap should be on the same slice as the delay element that it is sampling to minimize bubble issues. In Fig. 8 a detailed slice level view is given. By doing this, it was possible to reduce the bubble problem to delay elements contained on the same slice, leading to a maximum of 2-bit bubble problem. This issue can easily be solved before decoding, and therefore is acceptable.

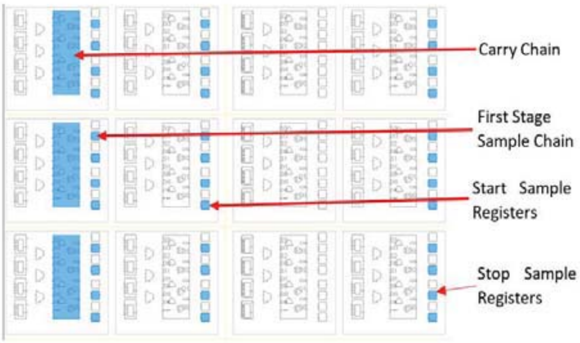

Figure 8 - Carry Chain Placement

When implementing a tapped delay line, it must be assured that the TDL covers, at least, one period of the system clock cycle. As the main delay element is the FPGA carry block, its delay must be determined to be able to build a TDL with the proper length. The FPGA programming tools offer a set of timing reports that can be used to get a rough estimation of these values. Nevertheless, these values are calculated using extremely pessimistic operation conditions, so implementing a TDL using the values obtained by the programming tools will often lead to a TDL that is much shorter than the required one. In our case, the slice carry block had a propagation delay of 114ps. As each slice has 4 carries, a 28.5ps single carry propagation delay was expected. After implementation and experimental measurement, a value of approximately 17.9ps single carry propagation delay was obtained, a value 37% smaller when compared to the simulation tools value. In order to experimentally measure the carry propagation delay, a tapped delay line was implemented according to the schematic of Fig. 1 and following the procedure described above to ensure minimum bubble issues and considerable delay uniformity between carry cells. Than a periodic square wave signal was used as the delay line input and the status of the delay line was sampled and stored multiple times by a system clock with a lower frequency than the frequency of the input signal. The frequency of the input signal was increased until at least half of the period could be sampled in the delay line. The test continued until the implemented delay line could sample an entire input signal period. By dividing half of the input signal period by the number of taps with logical value 1, a good estimation of the single carry cell propagation delay can be obtained.

Another aspect to consider when implementing a TDC based on a Nutt topology is the synchronization between the coarse and fine measurements, otherwise errors equal to one system clock cycle can occur. These types of errors happen when the rise or fall edge of the input signal occurs close to the rise edge of the system clock. Some solutions to this problem were proposed on the literature [16], [17], although none of them completely solve the problem.

This work proposes a novel synchronizer, shown in Fig. 6, to address the aforementioned issue, based on the use of two phased clocks as arbitrators. The two phased clocks are used as clock signals for two identical course counters. If the input signal arrives simultaneously with the main coarse counter clock, the counter may or may not count and the value latched by the carry chain will be zero. In this situation the second coarse counter is used obtain the correct counting value (Fig. 9, *Hit1* scenario). For some specific frequencies, the rise edge of the input signal may arrive at the situation described above, and the fall edge arrives at a point between the rise edge of the main clock and the rise edge of the phased clock, a counting error on the second coarse may exist, and therefore, a third coarse counter is needed. No more than three counters are needed to completely identify every possible counting error condition. In Fig. 9 every error condition is depicted and will be further explained.

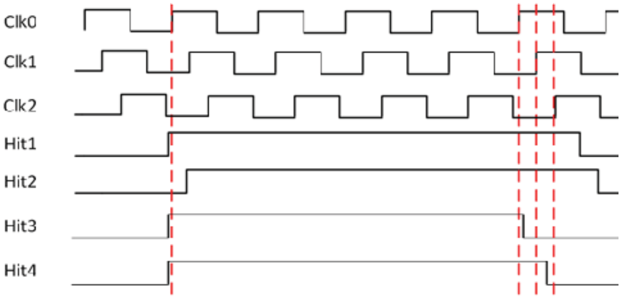

Figure 9 - Metastability measurement scenarios

The system clock, *Clk0* in Fig. 9, is responsible for both the sampling of the delay line and the primary coarse counting increment signal. When in a scenario where the hit signal does not occur close to a system clock rise edge (*Hit2* in Fig. 9), the value from the delay line and the primary coarse counter are directly used in the output value as no metastability situation can occur. In all the other scenarios, *Hit1*, *Hit3*, and *Hit4*, metastability can occur and therefore, the primary coarse counter has an ambiguous value. In these scenarios the second coarse counter is used. Nevertheless, there are situations in which the second counter can have an erroneous value, not due to metastability, but because the other edge of the input signal occurs between the rise edges of the phased clocks. In these scenarios, the third coarse counter is used to determine the true value that should be presented on the second counter. There are two situations where the second and third counter may have the same value. The first case, *Hit1* scenario, all the clocks could capture the hit signal before it goes low and the value sampled by the delay line will be higher than the phase difference between clocks. Therefore, the second coarse counter will have the correct counting value and can be compared to the primary counter directly. In the case of *Hit3* scenario, the second and third clocks will have the same value, but the value sampled by the delay line will be lower that the phase difference between clocks, meaning that the stop signal arrives after the *Clk0* rise edge but before the *Clk1* rise edge. In this case the value of the second counter needs to be incremented by one before comparing it to the primary coarse counter. Finally, when the value of the second counter and the third counter are different, *Hit4* scenario, the stop signals arrives in-between the *Clk1* and *Clk2* rise edges and therefore the second counter has the right counting value and can directly be compared to the primary counter. The same exercise can be executed for the stop signal metastability scenarios.

IV. Experimental Results

In the previous section the main implementation issues were explored and a new approach was proposed to address the synchronization problem when a fine and coarse counting mechanisms are used together. In this section the fine measurement delay line is characterized regarding its resolution, differential nonlinearity (DNL) and integral nonlinearity (INL).

The proposed TDC was implemented on a ZYBO development board. The TDC operating frequency was 200MHz, meaning a 5-nanosecond period, and at the end of each conversion, the value was stored in a FIFO memory. The code density test method was used to measure the real distribution of each delay element width. Ideally the width of every delay cell should be equal, although due to PVT variations, that is not true, which induce nonlinearities on the TDC. The delay cell width is calculated according to equation (1):

$$Wi=Ni*T/N \qquad (1)$$

where *Wi* is the ith delay cell width, *Ni* is the number of times the delay cell was recorded, *T* is the clock period in picoseconds and *N* is the number of measures performed. The density code test histogram for a clock frequency of 5 nanoseconds and 10000 measurements is presented in Fig. 10. The test was made using a Keysight 33600A series waveform generator, configured to output a square wave periodic signal with a 700kHz frequency and 50% dutty cycle. In order to ensure that the input signal had the same probability of being sampled by every tap in the delay chain, the TDC input signal frequency was chosen to avoid correlation with the TDC system clock. The output of the waveform generator was used as the input of the implemented TDC and the measured values were acquired and analysed using a MATLAB script. The result of this analysis is presented in Fig. 10.

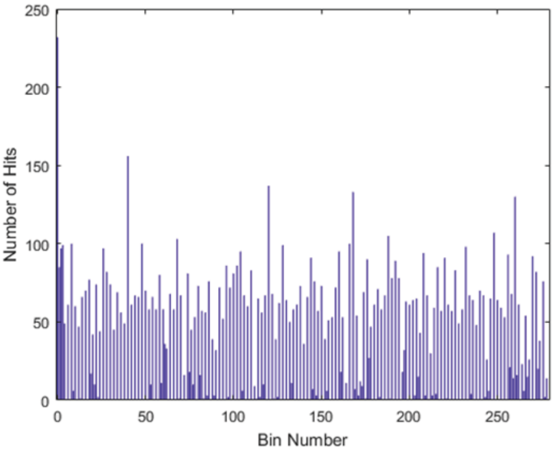

Figure 10 – Code Density Histogram

The resolution of the delay line TDC is obtained by subtracting the maximum measured value and the minimum measured value and then divide the result by the system clock period. A single shot resolution of 17.9 picoseconds was achieved. According to [18], the DNL and INL of the delay line can be obtained using these values, being the DNL defined as the delay cell width minus one standard cell width, and the INL being calculated by adding all the DNL from the first delay cell to the current delay cell. The implemented TDC achieved a maximum DNL of 5.4 LSB and a maximum INL of 5.8 LSB. The DNL and INL along the delay chain is presented in Fig. 11 and Fig. 12, respectively.

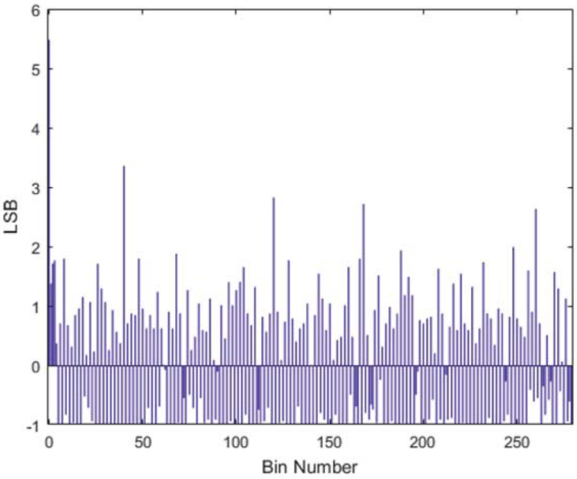

Figure 11 - TDC Differential Non-Linearity

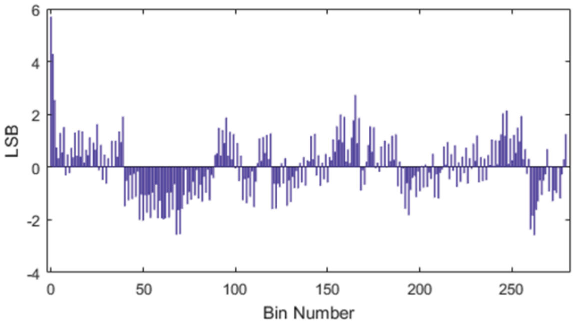

Figure 12 - TDC Integral Non-Linearity

To verify the proposed synchronization methodology efficiency a set of measurements were performed before the proposed synchronizer implementation, for frequencies between 700kHz and 900kHz. Within these measures, approximately 5% of the measures presents an error equal to +/-1 system clock cycle. The proposed synchronizer was implemented and a new set of measurements, for the same frequencies, was performed. The implemented mechanism was able to identify and correct all the metastability situations, and thus, no errors were registered in the set of data acquired after the proposed synchronizer implementation, endorsing the effectiveness of the proposed method.

V. Conclusion

When implementing sub nanosecond measurement systems, although high precision and linearity are fundamental, there are applications requiring a high measurement range as well. This is often achieved by combining coarse and fine counters, that need to be synchronized. Surprisingly, the literature is scarce on this subject being most of the works focused on the fine measurement, not mentioning the synchronization method when more than one counter type is used.

This work addresses this issue, proposing a TDC architecture with a synchronizer that fully covers the metastability issues caused when the input signal arrives near the system clock rise edge. The proposed synchronizer was implemented within the TDC, successfully synchronizing the result from the coarse counter and the tapped delay line.

A complete characterization of the implemented TDL regarding resolution and linearity is also presented in this work. A resolution of 17.9 picoseconds with a DNL of 5.4 LSB was achieved, on a ZYBO FPGA, making the proposed TDC a good candidate for high resolution, high range, single-shot applications.

Furthermore, the proposed architecture is also able to acquire and measure both rise and fall input events using a single tapped delay line, which greatly reduces the amount of needed resources. Being capable of achieving low resource usage also enables the proposed architecture to be easily instantiated in low-cost FPGA platforms. This enables the usage of the presented architecture, e.g. in a LiDAR system with a multiple laser configuration, improving the overall efficiency and throughput of the system. The presented architecture can also be applied in measurement systems based in the pull-in phenomenon, like the one presented in [19]. The high resolution and high measurement range of the proposed architecture enables a more precise way of determining the pull-in time, therefore improving the pull-in system resolution. The proposed TDC is currently being integrated in a pull-in-based measurement system, and a comparative study on the advantages of the TDC in regard to the traditional measurement method is planned as future work.

Acknowledgment

This work was supported by a Portuguese Scholarship from FCT - Fundação para a Ciência e Tecnologia and Bosch Car Multimedia, under the Advanced Engineering Systems for Industry (AESI) doctoral program. (Scholarship ID: PDE/BDE/114562/2016) and COMPETE and FCT: POCI-01-0145-FEDER-007043 within the Project Scope: UID/CEC/00319/2013

References

[1] P. Chen, Y. Y. Hsiao, and Y. S. Chung, “A high resolution FPGA TDC converter with 2.5 ps bin size and -3.79~6.53 LSB integral nonlinearity,” in *Proceedings of the 2nd International Conference on Intelligent Green Building and Smart Grid, IGBSG 2016*, 2016, pp. 2–6.

[2] J. Y. Won, S. Il Kwon, H. S. Yoon, G. B. Ko, J. W. Son, and J. S. Lee, “Dual-Phase Tapped-Delay-Line Time-to-Digital Converter with On-the-Fly Calibration Implemented in 40 nm FPGA,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 1, pp. 231–242, 2016.

[3] K. Cui, Z. Ren, X. Li, Z. Liu, and R. Zhu, “A High-Linearity, Ring-Oscillator-Based, Vernier Time-to-Digital Converter Utilizing Carry Chains in FPGAs,” *IEEE Trans. Nucl. Sci.*, vol. 64, no. 1, pp. 697–704, 2017.

[4] J. Torres *et al.*, “Time-to-Digital Converter Based on FPGA With Multiple Channel Capability,” *Nucl. Sci. IEEE Trans.*, vol. 61, no. 1, pp. 107–114, 2014.

[5] M. Zhang, H. Wang, and Y. Liu, “A 7.4 ps FPGA-based TDC with a 1024-unit measurement matrix,” *Sensors (Switzerland)*, vol. 17, no. 4, 2017.

[6] H. Homulle, F. Regazzoni, and E. Charbon, “200 MS/s ADC implemented in a FPGA employing TDCs,” *Proc. 2015 ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, pp. 228–235, 2015.

[7] Y. Wang and C. Liu, “A 3.9 ps Time-Interval RMS Precision Time-to-Digital Converter Using a Dual-Sampling Method in an UltraScale FPGA,” *IEEE Trans. Nucl. Sci.*, vol. 63, no. 5, pp. 2617–2621, 2016.

[8] Y. Wang and C. Liu, “A 4.2 ps Time-Interval RMS Resolution Time-to-Digital Converter Using a Bin Decimation Method in an UltraScale FPGA,” *IEEE Trans. Nucl. Sci.*, vol. 63, no. 5, pp. 2632–2638, 2016.

[9] P. Chen, Y. Hsiao, Y. Chung, W. X. Tsai, and J. Lin, “A 2.5-ps Bin Size and 6.7-ps Resolution FPGA Time-to-Digital Converter Based on Delay Wrapping and Averaging,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 1, pp. 114–124, Jan. 2017.

[10] N. Lusardi and A. Geraci, “8 - Channels High - Resolution TDC in FPGA,” 2015.

[11] X. Qin, L. Wang, D. Liu, Y. Zhao, X. Rong, and J. Du, “A 1.15-ps Bin Size and 3.5-ps Single-Shot Precision Time-to-Digital Converter With On-Board Offset Correction in an FPGA,” *IEEE Trans. Nucl. Sci.*, vol. 64, no. 12, pp. 2951–2957, Dec. 2017.

[12] Y. Wang, P. Kuang, and C. Liu, “A 256-channel multi-phase clock sampling-based time-to-digital converter implemented in a Kintex-7 FPGA,” in *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 2016, vol. 2016–July.

[13] Z. Li *et al.*, “Development of an integrated four-channel fast avalanche-photodiode detector system with nanosecond time resolution,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 870, no. November 2016, pp. 43–49, 2017.

[14] M. Abbas and K. Khalil, “A 23ps resolution Time-to-Digital converter implemented on low-cost FPGA platform,” *ISSCS 2015 - Int. Symp. Signals, Circuits Syst.*, pp. 0–3, 2015.

[15] S. Berrima, Y. Blaquiere, and Y. Savaria, “A multi-measurements RO-TDC implemented in a Xilinx field programmable gate array,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[16] F. Dadouche, T. Turko, W. Uhring, I. Malass, N. Dumas, and J. Le Normand, “New Design-methodology of High-performance TDC on a Low Cost FPGA Targets,” *Sensors & Transducers*, vol. 193, no. 10, pp. 123–134, 2015.

[17] F. Yuan, *CMOS Time-Mode Circuits and Systems: Fundamental Applications*, vol. 53, no. 9. 2015.

[18] Y. Wang, J. Kuang, C. Liu, and Q. Cao, “A 3.9-ps RMS Precision Time-to-Digital Converter Using Ones-Counter Encoding Scheme in a Kintex-7 FPGA,” *IEEE Trans. Nucl. Sci.*, vol. 64, no. 10, pp. 2713–2718, Oct. 2017.

[19] R. A. Dias *et al.*, “Real-Time Operation and Characterization of a High-Performance Time-Based Accelerometer,” *J. Microelectromechanical Syst.*, vol. 24, no. 6, pp. 1703–1711, Dec. 2015.