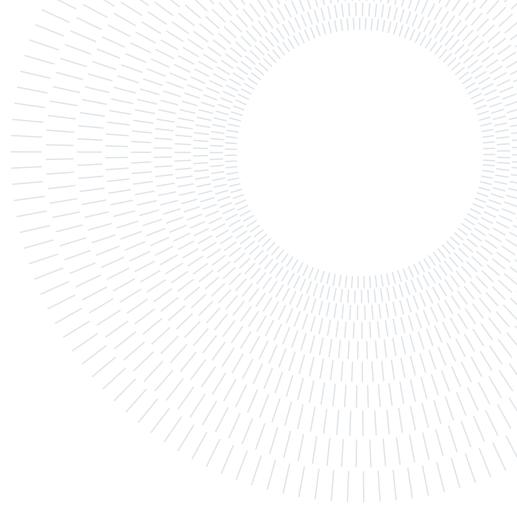




POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE



High-Performance Time-to-Digital Converter IP-Core for Xilinx Ultrascale/Ultrascale+ FPGAs

November 23, 2022

TESI DI LAUREA MAGISTRALE IN
ELECTRONICS ENGINEERING - INGEGNERIA ELETTRONICA

Mattia Consonni, 10531216

Advisor:
Prof. Nicola Lusardi

Co-Advisors:
Dr. Fabio Garzetti

Academic Year:
2021-2022

Abstract: The increasing demand for very precise time measurements in scientific research applications, ranging from the biomedical field to the industrial one, has led to the need for instruments, called Time Interval Meters (TIMs), characterized by features such as a resolution and single-shot precision in the picoseconds (ps) order, and a sampling rate in the order of hundreds of megahertz (MHz) to correctly process the detected physical events. Several TIM solutions are already present in the literature, implemented both in Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs). To cope with features such as fast-prototyping and low time-to-market, the FPGA approach has been chosen for this thesis work, and a fully-digital TIM, a.k.a. Time-to-Digital Converter (TDC), has been implemented. The presented TIM is a multi-channel Tapped Delay-Line (TDL) based TDC IP-Core for Xilinx Ultrascale/Ultrascale+ (XUS/XUS+) FPGAs. The IP-Core approach allows high-portability since the TDC can be promptly tested on different systems with a simple drag-and-drop operation. To exploit the benefits of the scaling, the 20-nm XUS technology has been investigated in this work, allowing achieving high measurement rates, low Dead-Time, high-resolution, and excellent single-shot precision. In particular, a single-shot precision in the order of 3 ps has been achieved, along with a maximum measurement rate of 200 Msps per channel, thus satisfying the high count-rate capability requirement of state-of-the-art detectors. From the design point of view, the TDC has been implemented and fully tested on a Kintex Ultrascale FPGA, hosted in the KCU105 Evaluation Board. To better investigate the power/precision and area/precision trade-offs, two different TDL solutions have been tested. The first solution is more performant in terms of single-shot precision at the expense of higher area occupancy, and it is based on the CARRY primitives (i.e., CARRY8) available in the fabric of the Xilinx FPGA. The second solution is instead less performant but more power/hardware-efficient, and it relies on Digital Signal Processor (DSP) primitives (i.e., DSP48E2). Finally, a hybrid architecture exploiting both CARRY and DSP resources has been tested, leading to results that show the best features of both TDL types (i.e., a single-shot precision of 3.8 ps, a power consumption of 0.49 W, and a maximum number of channels equal to 74). This architecture has been compared with the one exploiting just CARRY chains, which is the most performant in terms of precision (i.e., 2.8 ps), but it is characterized by a higher power consumption (i.e., 0.596 W) and area occupancy (i.e., maximum number of channels equal to 62).

Key-words: Time-to-Digital Converter (TDC), Tapped Delay-Line (TDL), Digital Signal Processor (DSP), Field-Programmable-Gate-Array (FPGA)

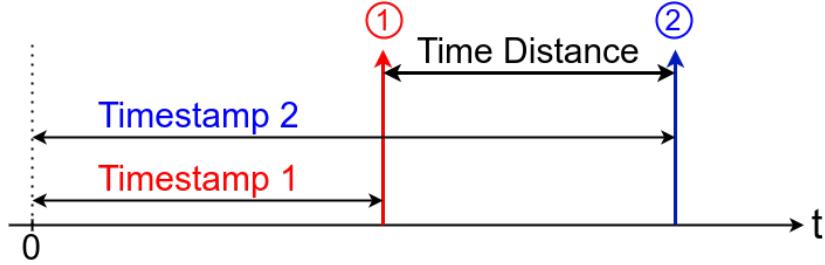


Figure 1: Graphical representation of the concepts of “Timestamp” and “Time Distance”.

1. Introduction

Concerning the scientific research field, recent years have been characterized by continuous growth in the study of temporal evolution in chemical-physical phenomena [24][2][3]; this has led to the need of employing high-performance instruments called Time Interval Meters (TIMs), with the task of measuring very small time intervals, with increasing precision and resolution. Nowadays, the events under study in these applications are characterized by a very high repetition rate and by a temporal resolution in the order of the picoseconds. Therefore, the TIM has to have features such as a resolution and measurement precision in the picoseconds order and a sampling rate of hundreds of megahertz (MHz), to correctly process and measure the physical events detected on the sensor front-end of the acquisition chain.

In this thesis work, a fully-digital TIM, a.k.a. Time-to-Digital Converter (hereinafter simply called TDC), implemented in Field Programmable Gate Array (FPGA) will be presented. This device is in charge of converting the time measurement into a digital output code. To satisfy the aforementioned requirements, the TDC has been implemented and fully-tested in a highly-scaled technology, i.e. the 20-nm Xilinx Ultrascale technological node, allowing exploiting the benefits of the scaling and achieving high resolution, excellent measurement precision, and high sampling rates. Moreover, to investigate the trade-offs between precision and area/power, two different implementations have been tested, the first one characterized by high single-shot precision, and the second one characterized by hardware and power saving (see Section 5.5).

The concept of “Time Measurement” is defined as the time elapsed between an absolute time reference, taken as “zero” on the time axis, and the occurrence of a certain event of interest. In this case, our measurement is a “Timestamp”; however, we are usually more interested to carry out a time measurement as the relative “Time Distance” between two events, the first one being the “START” signal, and the second being the “STOP” signal. Anyhow, a “Timestamp” is nothing but a particular case of time distance between a chosen absolute time reference, and the event under study. On the other hand, the time distance between two events is the time difference between their timestamps, each one calculated referencing to a common absolute “zero” time origin. Figure 1 shows the concept just explained.

1.1. Applications

TIMs are employed in many different industrial and academic fields, from the R&D to the final product, from the most exotic experimental setups to standard test-benches. For example, Time-of-Flight Positron Emission Tomography (TOF-PET) is the most relevant application in the biomedical field that requires TIMs. Concerning the industrial and automotive fields, instead, Laser Rangefinder techniques for 3D imaging represent the main example. Other academic applications are time-resolved spectroscopy experiments in which the Time-Correlated Single Photon Counting (TCSPC) technique is one of the most prominent. Let’s now describe these three main applications in detail.

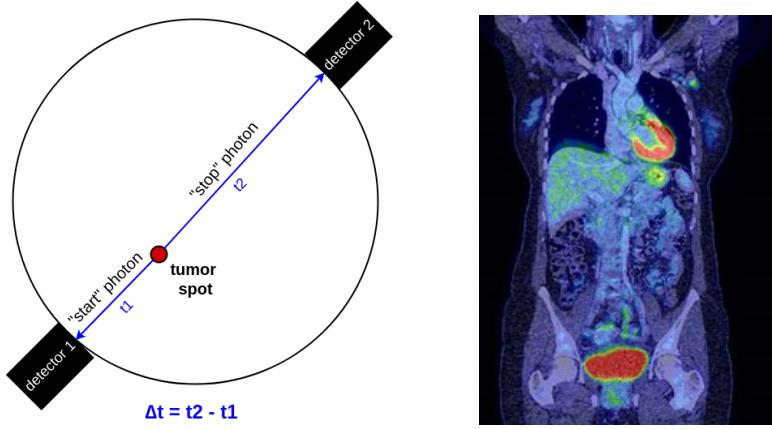


Figure 2: TOF-PET working principle (on the left) and resulting image (on the right).

1.1.1. Time-of-Flight Positron Emission Tomography

Time-of-Flight Positron Emission Tomography (TOF-PET) [28] is a 3D technique used for medical imaging to detect the presence of tumors in the human body. A substance called “radiotracer”, which is characterized by the capability to release a great number of positrons, binds to cancerous molecule. Once there, the positrons are emitted by the radioactive decay of the radiotracer and collide with the electrons present in the tissues. When there is a collision between a positron and an electron, characterized by the same electrical charge but with opposite sign, the so-called annihilation process occurs; this results in two 511 keV gamma photons traveling on the same line but in opposite directions. Referring to Figure 2 (on the left), these two gamma photons arrive with a certain delay (t_1 and t_2) to the ring of detectors surrounding the patient, and thanks to the TIM it is possible to measure the delay between the START event (e.g., the photon detection on detector 1) and the STOP event (e.g., the photon detection on detector 2); in this way, the time distance $\Delta t = t_2 - t_1$ can be used to derive the spatial position. Therefore, it is possible to retrieve the position of the tumor. We can see in Figure 2 (on the right) the resulting image.

1.1.2. Laser Rangefinding

Laser Rangefinding [23], also known as laser telemeter, is a technique using a laser beam to determine the distance to an object. As we can see in Figure 3 (on the left), the measured quantity is the time distance between the instant at which the laser pulse is sent toward the object and the instant at which it is detected by the photodiode (located in the same position as the sender) after being reflected by the target. Once the time information ΔT is obtained, and by knowing the speed of the light v , it is possible to retrieve the distance L at which the object is located with respect to the user, as shown by the following equation:

$$2 \cdot L = v \cdot \Delta T \quad (1)$$

and, therefore:

$$L = \frac{v \cdot \Delta T}{2} \quad (2)$$

The described process can be repeated an $N \cdot M$ number of times by arranging detectors (pixels) on an $N \cdot M$ matrix, so that a 3D image can be obtained. The first two dimensions (2D) are nothing but the two axes of the matrix itself, while the third dimension is the one given by the time measurements. The process just explained is called “3D imaging”. We can see in Figure 3 (on the right) the resulting 3D image in a LiDAR application.

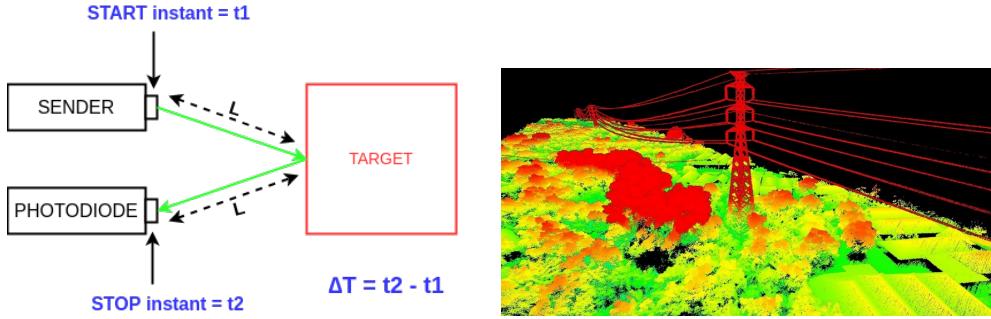


Figure 3: Laser Rangefinder working principle (on the left) and resulting 3D image in a LiDAR application (on the right).

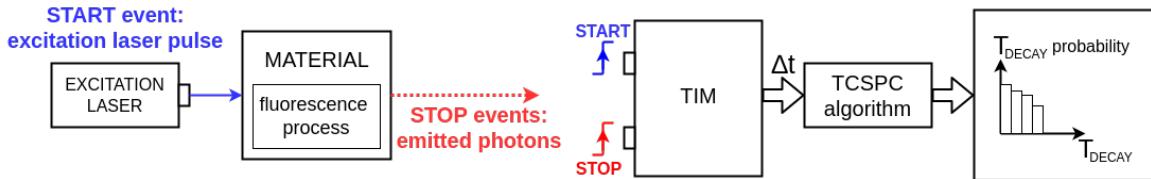


Figure 4: TCSPC working principle.

1.1.3. Time-Resolved Spectroscopy

Time-Resolved Spectroscopy is the study of dynamic processes, on times scales between seconds and femtoseconds, in chemical compounds or other types of materials. It exploits spectroscopic techniques that can be applied to any process leading to changes in the properties of the material under study. Usually, fluorescence processes are studied [37]. These kinds of experiments usually consist of the excitation of the substance under study using a light source (e.g., a laser pulse or a Synchrotron light source). After the excitation, the fluorescence phenomenon, which is an emission of photons from the material, takes place. To measure the time interval elapsing from the instant in which the stimulus is given to the material and the instant in which the material emits photons, a TIM has to be employed. In scientific research, there are a lot of time-resolved spectroscopy techniques, the main ones being: Laser-Induced Breakdown Spectroscopy (LIBS) [56], Time-Resolved InfraRed spectroscopy (TRIR) [30], Time-Correlated Single Photon Counting (TCSPC) [58]. The TCSPC method is one of the most common, and it consists of the excitation of the material under analysis thanks to the illumination by a pulsed laser. This causes the fluorescence process in the material, which is a repetitive emission of single photons with a statistic of some hundreds of picoseconds to some tens of nanoseconds. To get the decay shape of the fluorescence signal, repetitive measurements of the time intervals between the excitation pulse and the single photons emitted have to be carried out [58]. Therefore, a TIM with a resolution in the order of picoseconds is needed. As shown in Figure 4, the START signal of the TIM is the excitation laser pulse, while the STOP signals are the photons emitted by fluorescence. In this way, the photon arrivals per time can be collected in a histogram that represents the so called Probability Density Function (PDF) of the time decay profile (T_{DECAY}) of the material under analysis.

1.2. Typical measurement setup

In all the aforementioned applications, the typical measurement setup consists of a TIM measuring the time interval Δt between the two physical events. Generally, the acquisition chain starts with two detectors, with the task of converting the incoming physical events into electrical signals; then, these two electrical signals get processed by Time Discriminators to adapt the logical levels of the detectors' output to the ones required by the TIM's circuitry. In the end, the TIM measures the time difference Δt , giving a digital code in output.

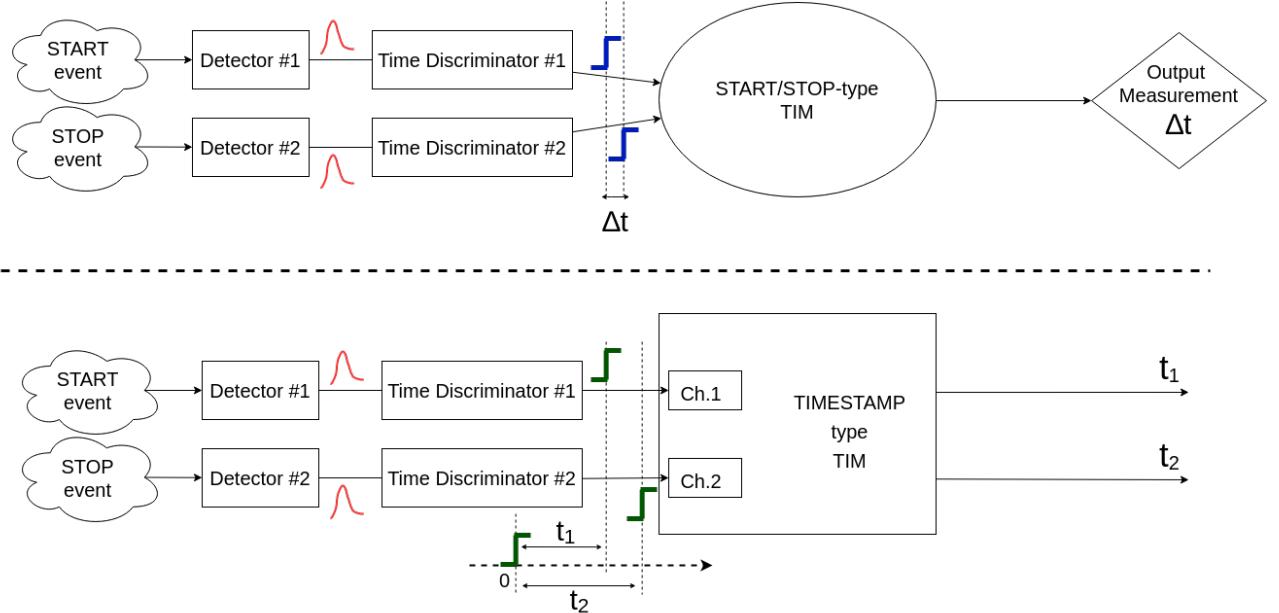


Figure 5: Measurement Setup employing a START/STOP-type TIM (at the top) and a TIMESTAMP-type TIM (at the bottom).

As said in Chapter 1, the time interval measurement can be represented either as the direct time difference Δt between the START and the STOP events or as the difference between their timestamps. In the first case, START/STOP-type TIM (Figure 5 at the top), each channel of the TIM is composed of two inputs (one for the START and one for the STOP). On the other hand, in the second case, TIMESTAMP-type TIM (Figure 5 at the bottom), each channel of the TIM has just one single input, with the scope of detecting the incoming physical event and measuring its timestamp.

If a system capable of measuring N physical events is required (a.k.a. multi-channel), a TIMESTAMP-type TIM would simply be characterized by a number of channels and outputs equal to N . Instead, a START/STOP-type TIM would need N START/STOP pairs. However, for the sake of compactness, all the STARTs are generally merged into a single START signal, thus requiring just one single connector for it and N connectors for the STOP signals.

The TIM that will be presented in this work is a fully-digital TIMESTAMP-type Time-Interval-Meter, a.k.a. TIMESTAMP-type TDC, implemented in FPGA. A TIMESTAMP-type TDC, just like a START/STOP-type TDC, converts the measured time interval into a digital output code.

1.3. Figures Of Merit (FOMs)

In this Section, the main Figures Of Merit (FOMs) characterizing the TDC's performance will be presented [13].

1.3.1. Resolution

Resolution is defined as the minimum time interval that can be measured by the TIM. In the case of TDC, the time duration of the Least Significant Bit (LSB) of the digital output code is commonly used as resolution. High-performance TDCs can reach resolutions in the order of hundreds of femtoseconds [11]. Figure 6 graphically shows the concept of LSB.

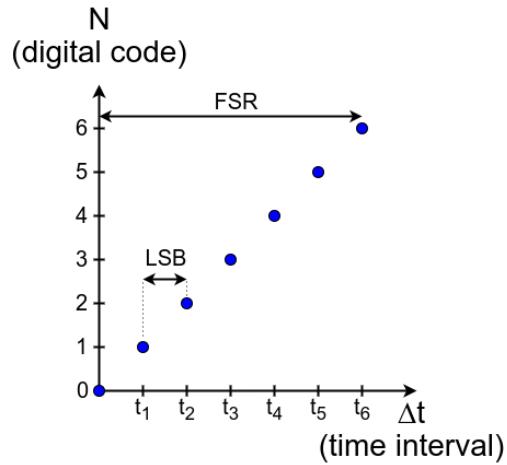


Figure 6: LSB and FSR with $N = 7$ quantization levels.

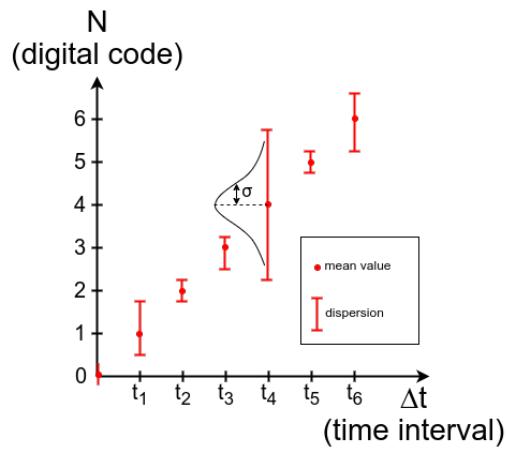


Figure 7: Single-Shot Precision.

1.3.2. Full-Scale Range

Full-Scale Range (FSR) is the maximum time interval that can be measured by the TIM. Figure 6 shows the typical trade-off existing between FSR and LSB (resolution); in fact, with the same number of quantization levels N (meaning that n bits are used in order to have $N \geq 2^n$), a greater FSR results in a greater LSB, since: $LSB = \frac{FSR}{N}$.

1.3.3. Precision

Single-Shot Precision, or simply precision, is the capability of the TDC to give the same output digital code given the same time interval in input. Based on this definition, the information about precision is obtained by studying the statistics of a repeated set of measurements of the same time interval. For the Central Limit Theorem, without external disturbances, the statistical distribution will be a Gaussian curve, whose standard deviation σ will be the index of the precision. Figure 7 graphically shows the explained concept. Therefore, a TDC with a low precision will result in a wide gaussian curve, which is equivalent to a high standard deviation and a wide statistic dispersion. Conversely, an instrument characterized by high precision will result in a narrower gaussian, and thus in a lower σ value.

In some cases, if the measured time interval is not characterize by a Gaussian shape (e.g., Time-over-Threshold measurements from scintillators-collected light pulses [34]) the Full-Width at Half-Maximum (FWHM) and the Full-Width at Tenth-Maximum (FWTM) are taken as the index of precision.

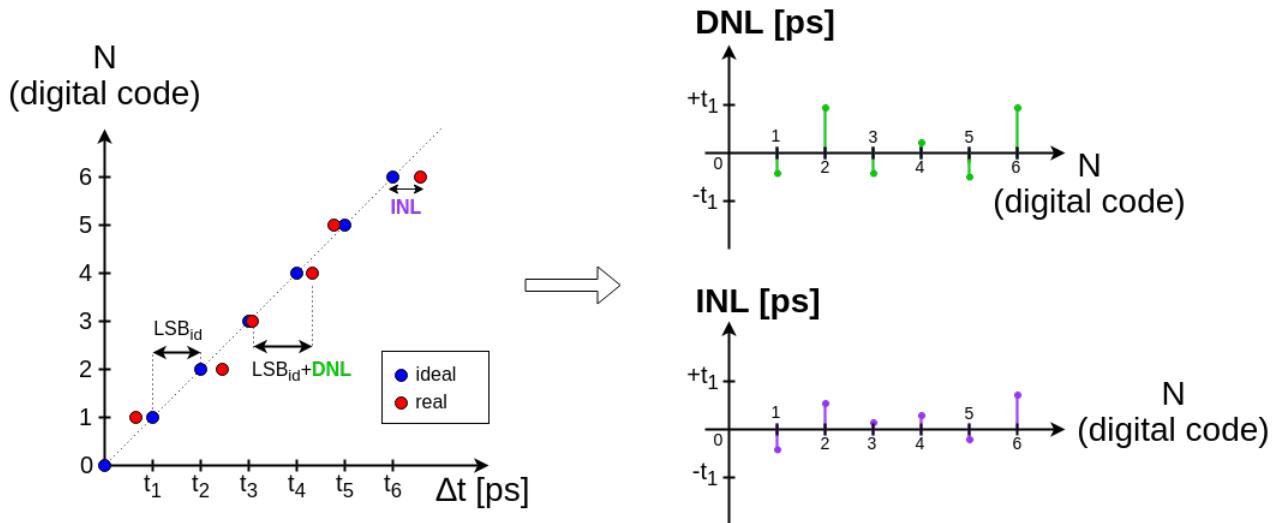


Figure 8: Graphical representation of DNL, which is the dispersion in time between consecutive digital codes (in red) with respect to LSB_{id} , and INL, which is the integral of the DNL and represents the difference in time between the ideal time interval (in blue) and the measured one (in red).

1.3.4. Linearity

In a TDC the non-linearity is represented by measuring its Differential Non-Linearity (DNL) and Integral Non-Linearity (INL). The DNL represents the dispersion in time between consecutive digital codes with respect to the nominal one (i.e., the LSB). Instead, the INL is the integral of the DNL and represents the difference in time between the ideal time interval and the measured one. They are commonly represented as a percentage of the LSB, and their calculation procedure will be shown in detail in Paragraph 5.6.1. Figure 8 graphically shows the DNL and the INL.

1.3.5. Conversion Time

The Conversion Time (T_{CONV}) is the time required by the TDC to produce a valid output measurement, given an input time interval. It is thus the propagation delay from the system's input to its output. In particular, if a pipelined TDC is used, T_{CONV} is equal to the time needed to cross all the Pipeline stages, and, if a storage mechanism (e.g., a First-In First-Out structure, FIFO) is also present, it is equal to the propagation delay needed to cross both the FIFO and the entire Pipeline. On the other hand, if a Pipeline structure is not implemented, T_{CONV} is equal to the propagation delay needed to cross the logic blocks used for generating the digital code. Figure 9 shows the concept of Conversion Time (in blue).

1.3.6. Maximum Rate

The Maximum Rate represents the maximum number of measures per second that one single channel (Maximum Channel Rate), or the global TDC system (Maximum Measurement Rate), can support. In a TDC requiring high Maximum Channel Rates, the channels are organized as pipelined structures in order to make the rate independent from T_{CONV} . In fact, while in a TDC without a Pipeline the Maximum Channel Rate corresponds to the inverse of its Conversion Time, in a pipeline-based TDC it corresponds to its throughput (i.e., $\frac{1}{T_{CLK}}$, where T_{CLK} is the period of the clock signal controlling the Pipeline). Figure 9 shows the concept of Maximum Channel Rate (in green). Concerning the Maximum Measurement Rate of multi-channel TDC systems, the bottleneck is usually given by the communication link between the TDC and the device, such as Personal Computers (PCs) or workstations, used to store and/or process the information.

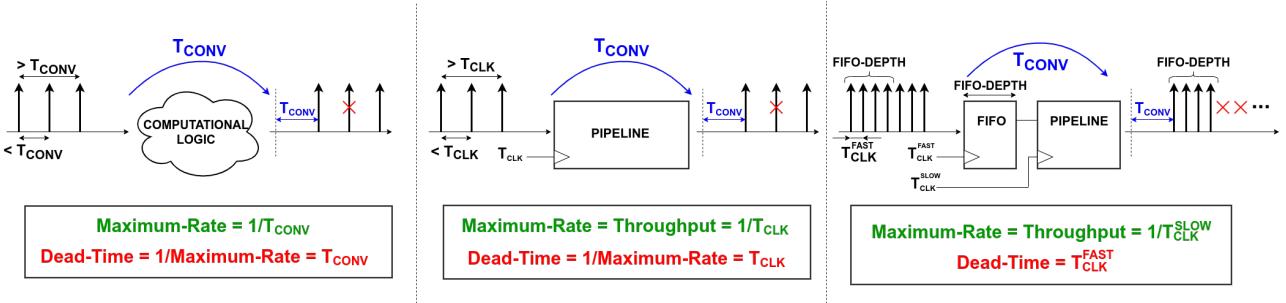


Figure 9: Conversion Time (T_{CONV} in blue), Maximum Rate (in green), and Dead-Time (in red) in the three different TDC architectures: without a Pipeline structure, a.k.a. only logic, (on the left); with a Pipeline structure clocked at the system clock T_{CLK} (at the center); with a fast dual-port storage mechanism (i.e., FIFO) that stores measurements at T_{CLK}^{FAST} alongside the Pipeline clocked with the system clock T_{CLK}^{SLOW} (on the right) where $T_{CLK}^{FAST} < T_{CLK}^{SLOW}$.

1.3.7. Dead-Time

As Dead-Time we intend the minimum time interval between two consecutive events on the same channel that can be measured. It is representative of the multi-hit capability of the TDC. If a pure computational logic (Figure 9 on the left) or a simple pipeline structure (Figure 9 on the center) is used, the Dead-Time corresponds to the inverse of the Maximum Channel Rate. A common strategy to achieve lower Dead-Time consists of using a FIFO with fast writing capability, controlled by a clock signal with period T_{CLK}^{FAST} , lower than T_{CLK}^{SLOW} used to clock the systems. In this way, only the first stages of the logic are sped up to the maximum allowed frequency (T_{CLK}^{FAST}), keeping the rest of the system at T_{CLK}^{SLOW} . In fact, until free space is present in the storage mechanism (e.g., FIFO not full), a lower Dead-Time, equal to T_{CLK}^{FAST} , is allowed. Figure 9 (on the right) represents the concept just explained, showing the improved multi-hit capability of the TDC with respect to the first two cases (on the left and at the center).

1.3.8. Voltage and Temperature Sensitivities

Voltage and Temperature Sensitivities describe how much the resolution, precision, and linearity of the TDC change, with respect to power supply and temperature fluctuations.

Usually, compensation techniques are required in order to reach high robustness against these variations [57].

1.4. FPGA vs. ASIC

TDCs, like every digital circuit, can be implemented both in an Application Specific Integrated Circuit (ASIC) [14] or using a Programmable Logic Device such as Field Programmable Gate Array (FPGA) [11]. An ASIC is a chip designed to perform one specific functionality, suited for high-volume production. All the resources are placed on a single silicon chip, and the functionality is fixed by the manufacturing process.

The main advantages of ASICs with respect to FPGAs are the very high-performance (e.g., fast operating frequency), the lower power consumption, and lower area occupancy, since the entire design is tailored to perform that specific functionality. Moreover, it can host analog and mixed-signal components other than digital ones.

The main drawback is represented by the impossibility of changing the ASIC's functionality; it thus lacks flexibility, due to the fixed nature of the design. Furthermore, it is characterized by a high time-to-market, in the order of years, due to the very long manufacturing and design processes, and by high Non-Recurring Engineering (NRE) costs for the fabrication, representing a higher entry barrier with respect to FPGAs. This makes ASICs more suitable for massive production governed

Feature	FPGA	ASIC
Performance		✓
Power Efficiency		✓
Flexibility	✓	
Time-to-market	✓	
NRE costs	✓	
Analog elements		✓

Table 1: Comparison between FPGA and ASIC.

by the economy of scale and not compatible with the request of fast-prototyping. For these reasons, another approach, less performant in terms of FoMs but leaner, is preferred in scientific research (in those cases where few units are requested) and for fast-prototyping. This method consists of using an FPGA, which is a fully-digital electronic device that can be configured and re-configured by the designer, after the manufacturing process, simply using the Hardware Description Language (HDL) code that describes the system. Moreover, if FPGAs are used, there is no need to produce custom lithographic photomasks, reducing the Non-Recurring Engineering (NRE) costs.

Table 1 summarizes the strengths belonging to each one of the two approaches.

In this work, the FPGA approach has been chosen to make the TDC more suitable for fast-prototyping, guaranteeing the FoMs required by the applications described in Section 1.1 at the State-of-the-Art.

1.5. Architectures

In research and industrial worlds different types of Time-Interval-Meters (TIMs) are employed. As generic electronic circuits, also TIMs can be fully analog, fully digital (a.k.a. TDCs), or mixed-signal. In the following paragraphs, the main types of TIMs will be described, mainly focusing on TDCs.

1.5.1. Time-to-Amplitude Converter

Considering the mixed-signal approach, the Time-to-Amplitude Converter (TAC) is historically one of the main device used in research (above all in TCSPC [31]), and it converts a time interval Δt into a voltage level [60]. More precisely, its working principle consists of building a voltage ramp that starts synchronously with the START event, and stops when the STOP event occurs. In this way, a voltage level proportional to the measured time interval is obtained. Then, an ADC converts the measured voltage level into a digital code. We can see the structure of a conventional TAC in Figure 10, where the capacitor C is charged by a constant current I as long as START = 1 and STOP = 0; then, as soon as STOP = 1, START transits to 0, and the capacitor gets isolated from the current generator, thus holding the charge.

The value of the voltage across the capacitor has the following expression:

$$V_C = \Delta t \cdot \frac{I}{C} \quad (3)$$

and it is converted, by the n -bit ADC, into a digital output code representing the time distance Δt . The resolution of the TAC is equal to:

$$\Delta t_{LSB} = \frac{V_{FSR}}{2^n} \cdot \frac{C}{I} \quad (4)$$

where V_{FSR} is the full voltage span of the ADC, given by its supply voltage. It can be seen from Equation 4 that this architecture suffers from a trade-off between resolution and Full-Scale Range (FSR). This trade-off, along with the non-idealities of the current generator and the mismatches of the other components due to process and temperature variations [31], represents a significant drawback and, for this reason, a fully-digital approach is usually preferred.

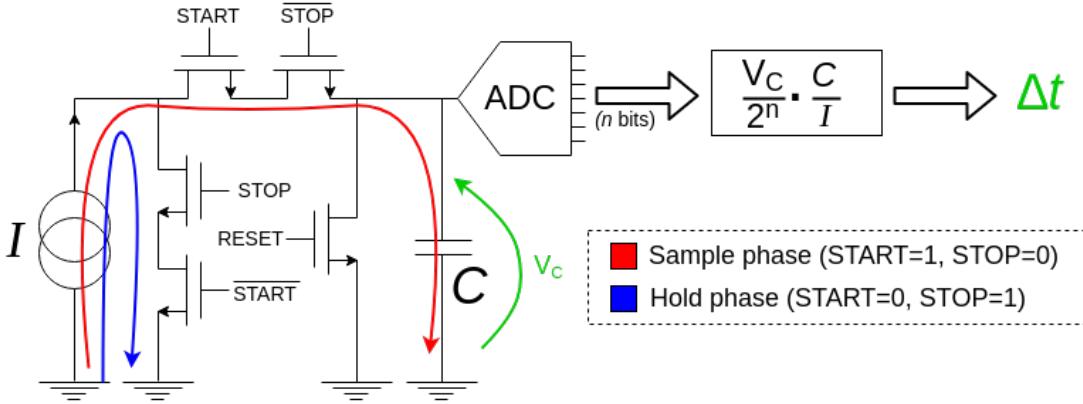


Figure 10: Structure and operation of a conventional Time-to-Amplitude Converter (TAC).

The great advantage of this architecture is the relatively small area occupied. On the other hand, a significant drawback is represented by the need to continuously charge and discharge the capacitor C , thus making this architecture quite slow, and not suitable for the multi-hit capability required by some applications. The power consumption depends on the wanted precision; in fact, to lower the noise and achieve a better precision, the current I has to be increased, thus increasing also the power consumption [43]. Being a mixed-signal approach, the TAC can be implemented in ASIC only.

1.5.2. Shift-Clock Fast-Counter TDC

The Shift-Clock Fast-Counter (SCFC) is a TIMESTAMP-type TDC architecture derived from a simple counter. In fact, it is possible to use a simple counter, driven by a clock signal with period T_{CLK} , as a TDC that produces a timestamp associated with the counter value at the asynchronous signal occurrence [18]. If the asynchronous signal happens after N_{cycles} clock cycles, the provided timestamp will be equal to:

$$Timestamp = N_{cycles} \cdot T_{CLK} \quad (5)$$

However, this single-counter implementation is useless for modern applications requiring high-resolution TDCs, since the achievable resolution Δt_{LSB} , in this case, is just equal to T_{CLK} . This result is not sufficient to cope with the applications since, in FPGAs, T_{CLK} is in the nanoseconds order. To increase the resolution of a factor N , N counters are used, each one driven by the same clock signal used before but shifted in phase by $\Delta\phi = \frac{2\pi}{N}$ thanks to a Phase-Locked Loop (PLL). This allows having more clock edges acting as intermediate levels in the counting process, allowing to achieve:

$$\Delta t_{LSB} = \frac{T_{CLK}}{N} \quad (6)$$

The FSR is instead equal to the case of the single counter:

$$\Delta t_{FSR} = (2^n - 1) \cdot T_{CLK} \quad (7)$$

where n is the number of bits composing the counter.

Figure 11 represents the architecture just described.

As shown by Equations 6 and 7, the main advantage of this architecture is the absence of the trade-off between resolution and FSR. Moreover, the extremely slender architecture minimizes the DNL and INL, which are limited only to clock fluctuations. The SCFC-TDC can be implemented both in ASIC and FPGA; in the FPGA case, the main drawback is a moderate resolution, limited to the maximum achievable clock frequency and the available clock lines on the device. Considering 28-nm Xilinx 7-Series FPGAs, just twelve clock lines are available [45], with a maximum clock frequency of hundreds of MHz, limiting the resolution to some hundreds of ps, while, in 20-nm Xilinx Ultrascale FPGAs, the available clock lines are 24 [52], with a maximum clock frequency of the same magnitude,

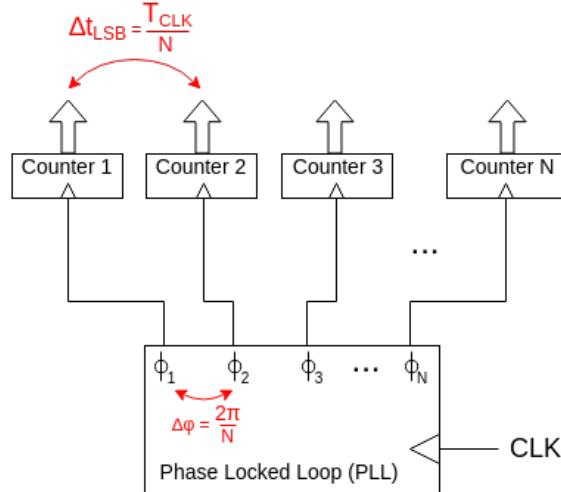


Figure 11: SCFC-TDC architecture employing N n-bit counters.

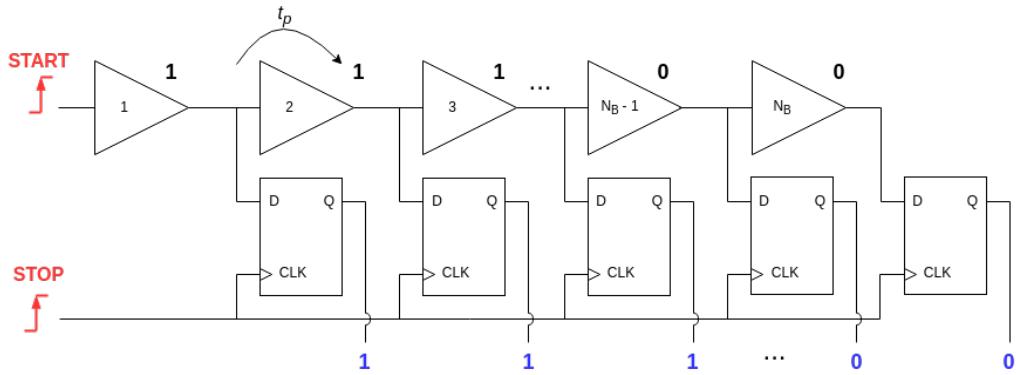


Figure 12: TDL structure and propagation mechanism.

thus giving a halved LSB. Obviously, an increase in terms of resolution is paid with a linear increase in area occupancy and greater power consumption [19].

1.5.3. Delay-Line TDC

To overcome the resolution limit given by the clock period, Delay-Line TDCs (DL-TDCs) are typically preferred. A Delay-Line (DL) is nothing but a chain of buffers, also called “taps” or “bins”, along which the input START and STOP events propagate. It is a START/STOP-type TDC, in which the time interval between the two signals is computed by taking a snapshot of the DL as soon as a sampling event occurs. Two main types of DL TDCs are available: the Tapped Delay-Line TDC (TDL-TDC) [11] and the Vernier Delay-Line TDC (VDL-TDC) [8].

Tapped Delay-Line TDC, TDL-TDC As reported in Figure 12, a Tapped Delay-Line (TDL) TDC consists of a sequence of buffers connected in series, where the output of each bin is sampled by a flip-flop (e.g., edge-triggered positive D flip-flop is the most common one). In this way, the time elapsed from the rising-edge of the START event to the STOP one can be obtained by simply counting the number of bins crossed by the former, when the latter occurs [29]. More precisely, the rising-edge of the START is propagated bin-by-bin (each one having its own propagation delay t_p) over the TDL, asserting in sequence the inputs of the flip-flops from ‘0’ to ‘1’; so, when the rising-edge of the STOP occurs (i.e., the following rising-edge of the clock signal used to sample the TDL), the status of the TDL is memorized by the flip-flops, thus generating a thermometric code that represents the number of bins crossed by the rising-edge of the START.

The resolution (LSB) of this architecture is equal to:

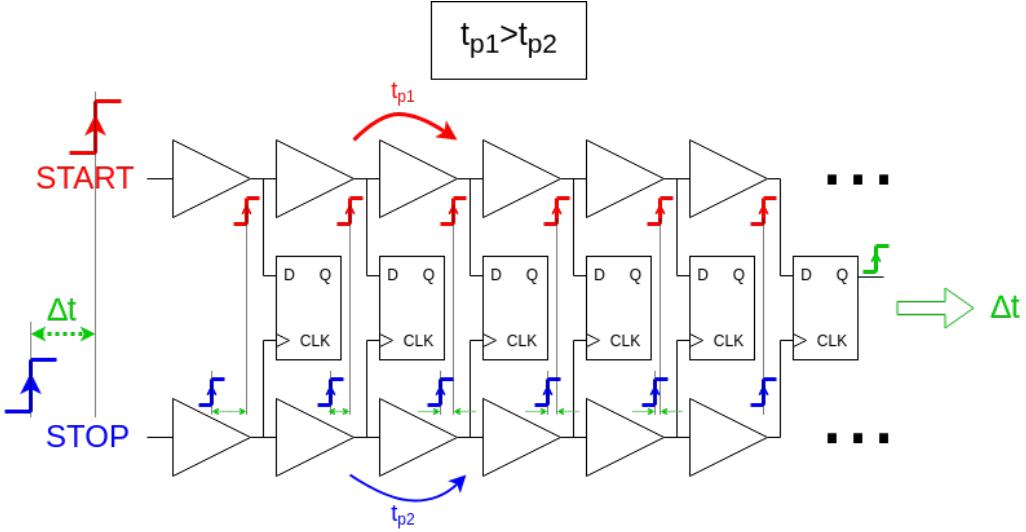


Figure 13: Vernier Delay-Line (VDL) structure and propagation mechanism.

$$LSB = t_p \quad (8)$$

The FSR is instead dependent on the number N_B of buffers composing the chain, as shown by the following Equation:

$$FSR = N_B \cdot t_p \quad (9)$$

Therefore, as we can see from Equations 8 and 9, at the same area occupancy (i.e., the same number of taps N_B), a trade-off between FSR and LSB arises. However, this trade-off can be overcome using interpolatory techniques like Nutt-Interpolation (as we will see in Section 2.1).

The main drawback of this architecture with respect to others is the high area consumption, due to different contributions. Firstly, to increase the FSR the number of buffers N_B must be increased; secondly, a further increase in the area is given by the need for a thermometric-to-binary converter, with the scope of extracting a binary value from the thermometric code.

Unlike ASICs, FPGAs do not allow self compensation techniques to overcome process, voltage, and temperature (PVT) variations that lead to dispersion in the propagation delays on the TDL. Therefore, a calibration process is mandatory to obtain a high-performance TDC in an FPGA, leading to further area consumption (see Section 2.3).

Vernier Delay-Line TDC, VDL-TDC The Vernier Delay-Line (VDL) TDC [8] relies on two chains with the same number N_B of buffers and, while the output of the first chain's taps are connected to the D-input of the flip-flops, the outputs of the second chain are connected to their clock inputs. The START signal propagates on the first chain with a propagation delay t_{p1} on each bin, and the STOP signal travels on the second chain with a propagation delay t_{p2} on each bin, which is lower than t_{p1} . Hence, as soon as the STOP signal, traveling on the faster chain, reaches the START one, the flip-flop corresponding to bin N_x changes its state, and provides the information of the time interval Δt between START and STOP. A representation of the VDL structure and working mode can be seen in Figure 13.

Precisely, the time instant at which both the START and STOP signals reach the same bin N_x is equal to:

$$T_x^{(start)} = N_x \cdot t_{p1} \quad (10)$$

$$T_x^{(stop)} = \Delta t + N_x \cdot t_{p2} \quad (11)$$

By equating relations 10 and 11, the time interval Δt is obtained:

$$\Delta t = N_x \cdot (t_{p1} - t_{p2}) , t_{p1} > t_{p2} \quad (12)$$

The minimum measurable time distance represents the resolution of the VDL, and it is equal to:

$$LSB = t_{p1} - t_{p2} \quad (13)$$

The FSR is instead equal to:

$$FSR = N \cdot (t_{p1} - t_{p2}) \quad (14)$$

Equations 13 and 14 show that the VDL-TDC suffers from the same trade-off between FSR and LSB, alongside the other drawbacks characterizing the TDL-TDC.

Furthermore, in comparison to TDL, the VDL requires a good matching between the START-chain and the STOP-chain. In fact, as shown in Equation 13, the resolution of the VDL-TDC depends on t_{p1} and t_{p2} . For this reason, due to the impossibility of achieving an accurate matching in FPGAs, this architecture is unreliable, and the TDL-TDC is the most suitable choice.

Ring Delay-Line TDC, RDL-TDC The Ring Delay-Line TDC (RDL-TDC) relies on a closed-loop DL structure, acting as a controlled clock generator (ring-oscillator) [27]. The clock generator uses N_B delay units, with a propagation delay equal to t_p , with the START and the STOP signals acting as control units; then, N_B latches are used to sample the state of the DL during the transitions of the generated clock signal. Let's see in detail the operation of the RDL-TDC. Firstly, when START and STOP both have a "low" logical level, the generated clock signal keeps a "high" value; then, as soon as START switches to '1', the clock generator acts as a ring oscillator, providing a clock signal with a certain period T_{CLK} . The battery of latches samples the DL during the transitions of the clock signal, thus providing the time information (i.e., $N_F \in [1; N_B]$). Basically, N_F is the number of "ones" or "zeros" (depending if the generated clock signal is in its "high" or "low" half-period, respectively) in the generated thermometric code, representing the number of transitions in half of a clock period. Once also the STOP signal transits to '1', the clock signal holds a "high" logical value, and the sampled data is provided at the output. The final time measurement will be equal to:

$$T_M = N_F \cdot LSB \quad (15)$$

where:

$$LSB = t_p \quad (16)$$

On the other hand, the FSR is equal to:

$$FSR = N_B \cdot t_p \quad (17)$$

thus showing the same trade-off of the other DL-TDCs.

Figure 14 shows the structure and the working principle of this architecture.

The RDL-TDC can be implemented both in ASIC and FPGA; however, since the latter does not allow compensation mechanisms to correct PVT variations, required to have matching in the duration of the generated clock periods, the ASIC approach is more suitable for this architecture.

1.6. State-of-the-Art

In this Section, we present some context about how the TDC can be implemented, showing the pros and cons of each implementation. Table 2 reports some of the existing academic TDC solutions found in the literature, with their relative technology nodes, performances (i.e., the achieved FoMs described in Section 1.3), architectures (i.e., the ones described in Section 1.5), and applications (i.e., the ones described in Section 1.1).

We can see from Table 2 that all the ASIC solutions are characterized by low power consumption (i.e., in the order of some μW with respect to FPGA that achieve some mW), low non-linearities (i.e.,

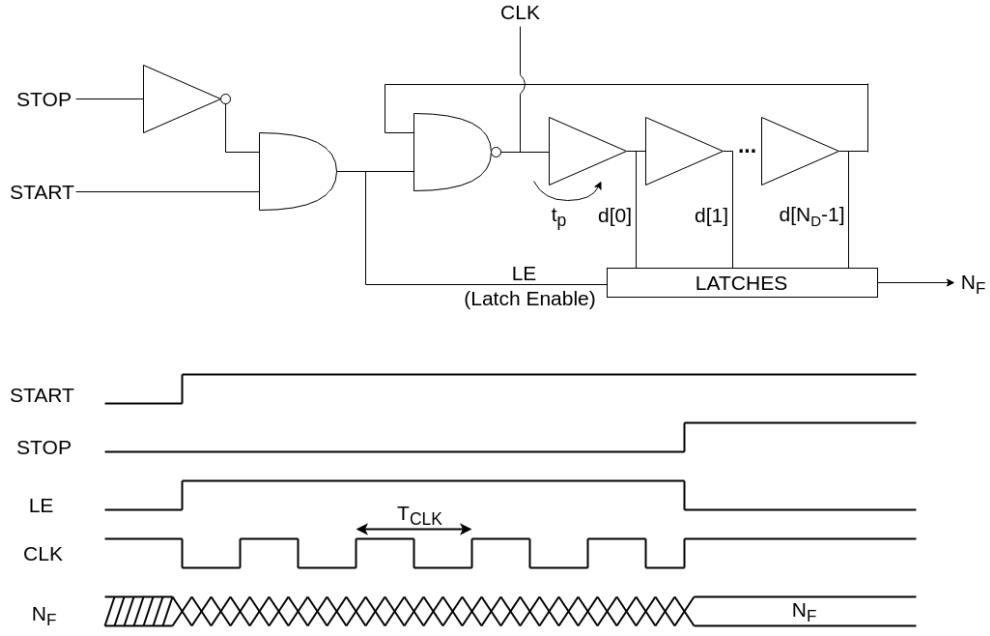


Figure 14: RDL-TDC structure and timing diagram.

ASIC TDCs	[27]	[9]	[4]	[40]	[26]
Technology	180 nm	180 nm	350 nm	130 nm	130 nm
Architecture	RDL	RDL	RDL	SCFC	SCFC
Application	TOF in space	Jitter meas.	Laser Range F.	Spectroscopy	Spectroscopy
LSB [ps]	201	2	40	781	5
Precision [ps r.m.s.]	N.A.	1.44	2.2	300	2.5
DNL [LSB]	[-0.18:0.05]	[-1:1]	N.A.	[-0.05:0.05]	[-0.8:0.6]
INL [LSB]	N.A.	[-1:1.3]	[-0.6:0.6]	[-0.05:0.05]	[-1.2:0.4]
FSR [ns]	6.8e+06	130	22	N.A.	N.A.
Dead-Time [ns]	N.A.	303	N.A.	N.A.	N.A.
Power [mW]	0.3155	18	0.0016	6.5	43
Num. of Chs	N.A.	N.A.	N.A.	48	8

FPGA TDCs	[41]	[39]	[44]	[59]	[36]	[35]
Technology	Kintex-7	Kintex-7	Actel 130-nm	Artix-7	Spartan-6	Virtex-6
Architecture	SCFC	TDL	VDL	TDL	TDL	TDL
Application	TOF-PET	TOF-PET	Nuclear science	TOF exp.	TOF exp.	TOF-PET
LSB [ps]	89.3	22.7	42	33	1.14	1.7
Precision [ps r.m.s.]	56.2	85.7	16.4	12.86	6	4.2
DNL [LSB]	[0.44:0.87]	<3	[-1:0.9]	[-0.6:0.6]	N.A.	[-1:4]
INL [LSB]	[0.44:0.82]	<4	[-1:3.5]	[-0.8:0.8]	19.36	[-9.8:6.2]
FSR [ns]	N.A.	5.24e+03	6.553e+05	N.A.	1e+10	6.25
Dead-Time [ns]	4.3	30	100	10	2000	N.A.
Power [mW]	N.A.	N.A.	N.A.	N.A.	750	N.A.
Num. of Chs	256	24	N.A.	32	N.A.	N.A.

Table 2: Comparison of different TDC solutions, both in ASIC (at the top) and FPGA (at the bottom).

DNL, $\text{INL} \leq 1 \text{ LSB}$), and good precision (i.e., $\sigma_{r.m.s.}$ in the order of some picoseconds). As explained in Section 1.4, ASICs make it possible to perfectly tailor the TDC and maximize the performance, at the expense of a higher time-to-market (i.e., years). It is also worth noting that the RDL-TDC has the highest precision and almost null power consumption; besides, it is present only in the ASIC technology due to the need of compensating PVT variations of the ring oscillator (see Paragraph 1.5.3), which is not possible in an FPGA.. Referring to the FPGA solutions, the [41] implementation reported in the Table relies on the SCFC architecture described in Paragraph 1.5.2, showing a limited resolution (i.e., in the order of one hundred picoseconds); this is due to the already mentioned limited number of available clock lines on the Kintex-7 FPGA fabric (i.e., 12). However, its compact structure allows it to have a high number of channels (i.e., 256). Regarding instead the implementations based on a VDL, [44] represents an academic solution; anyhow, as mentioned in Paragraph 1.5.3, VDL-TDCs are generally not easy to find for highly-scaled FPGAs (e.g., 28-nm) due to their mismatches between the START chain and the STOP one, which lead to low resolutions and unreliable measurements. The [44] represents an exception due to the poorly-scaled technological node (i.e., 130-nm), which leads to lower mismatches between the chains, and thus makes it possible to implement it in FPGA. The TDL thus represents the best architectural solution for a TDC in FPGA. We can see from the implementations reported in the Table (i.e., [39], [59], [35], and [36]) that the achieved LSBs and single-shot precisions are quite good (i.e., in the order of tens of picoseconds), alongside good linear behaviors (i.e., DNL, INL < 20 LSBs); FoMs that are comparable to ASIC solutions. This is achieved thanks to Sub-Interpolation and the bin-by-bin calibration procedure to overcome PVT variations, which will be described in detail in Sections 2.2 and 2.3. However, as said in Paragraph 1.5.3, the TDL-TDC suffers from a trade-off between FSR and LSB. For example, we can see in the [35] implementation that the achieved resolution is very good (i.e., 1.7 ps), but it is paid with a low FSR (i.e., 6.25 ns). The [39] implementation shows instead a better FSR (i.e., 5240 ns), which has been achieved using a technique that composes the time measurement of a Fine part (i.e., the one calculated by the TDL as shown in Paragraph 1.5.3) and of a Coarse part (i.e., the count of the elapsed clock cycles from the power-on of the TDC up to the generation of the time measurement). This technique, called Nutt-Interpolation, has been adopted in this thesis work and will be described in detail in Section 2.1.

1.7. Thesis Goal

To satisfy the applications described in Section 1.1, the implementation of a TIM has been undertaken in this thesis work. Then, after having analyzed the FoMs in Section 1.3, the ones required by State-of-the-Art TIMs have been identified:

- Resolution (LSB);
- Full-Scale Range (FSR);
- Single-Shot Precision;
- Linearity;
- Measure Rate;
- Dead-Time.

Therefore, the achievement of these FoMs has been set as the main target for this thesis.

After that, two different technological solutions (i.e., ASIC and FPGA) have been considered in Section 1.4, analyzing their pros and cons. In this way, to cope with those applications fitting with fast-prototyping and low NRE costs, the FPGA approach has been chosen; as a consequence of that, a fully-digital TIM, a.k.a. TDC, has been selected for the design process. In this way, an high tunability of the system is allowed.

Finally, by considering the different TDC architectures (Section 1.5) and their achieved FoMs in several State-of-the-Art implementations (Section 1.6), the TDL architecture has been chosen.

The system has been implemented at the Digital Electronics Laboratory (DigiLAB) at Politecnico di Milano as a tunable, plug-and-play, Intellectual Property Core (IP-Core) [11]. The IP-Core is a package containing the firmware and the parameters of all the blocks composing the TDC, and it is characterized by a high portability. The latter comes from the fact it can be used as a simple drag-and-drop unique block, thus allowing it to be promptly tested on different FPGA devices. Indeed,

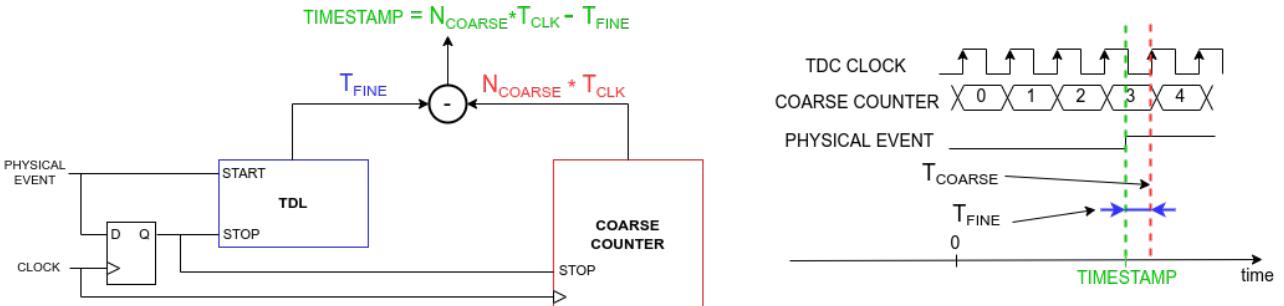


Figure 15: Nutt-Interpolation technique.

all the TDC's parameters such as FSR, resolution, and number of channels, can be set by the user in the instantiation stage by simply choosing their value on the IP-Core's Graphical User Interface (GUI). The implemented system will be described in detail in Chapters 2 and 3 and experimentally validated in Chapter 5.

2. TDL-TDC Design rules

As shown in Paragraph 1.5.3, the TDL-TDC suffers from three main problems:

1. a trade-off between LSB and FSR;
2. a minimum resolution (t_p) due to the propagation delay of the technological node;
3. moreover, PVTs have to be compensated in order not to let the dispersion of the t_p ruin the linearity. Thus, in the case of FPGA implementation, the calibration of the TDL is mandatory.

In this Chapter, we will describe the proposed solutions to solve each problem:

1. Nutt-Interpolation, to solve 1;
2. Sub-Interpolation, to solve 2;
3. Calibration, to solve 3.

2.1. Nutt-Interpolation

As mentioned before, the TDL-TDC architecture suffers from a trade-off between resolution (LSB) and Full-Scale Range (FSR). In order to solve this issue, a technique called Nutt-Interpolation is exploited [25]. This technique composes the time measurement of a Coarse part (T_{COARSE}) and a Fine one (T_{FINE}), the former being characterized by high FSR and low resolution, and the latter having low FSR and high resolution. If we would to implement a TIMESTAMP-type Nutt-Interpolated system, the time measurement (a.k.a., timestamp T_{MEAS}) will be the difference between these two parts:

$$T_{MEAS} = T_{COARSE} - T_{FINE} \quad (18)$$

As shown in Figure 15 (left side), the fine measurement (T_{FINE}) is performed by a TDL-TDC (START/STOP-type) and the coarse one (T_{COARSE}) by a simple counter (TIMESTAMP-type). In order to synchronize T_{FINE} to T_{COARSE} , the physical event is connected to the START of the TDL-TDC, while the STOPs of the TDL-TDC and the counter are connected to the sampled version of the physical event. The latter is obtained by sampling the external event using the same clock of the counter (with period T_{CLK}). In this way, T_{FINE} is equal to the time distance from the physical event to the following clock edge; instead, T_{COARSE} is equal to the number of clock cycles (N_{COARSE}) elapsed from the power-on of the system to the sampling of the physical event.

Finally, from Equation 18, we obtain:

$$T_{MEAS} = N_{COARSE} \cdot T_{CLK} - T_{FINE} \quad (19)$$

Figure 15 (right side) graphically explains Equation 19.

Thanks to the counting of the elapsed clock cycles, the FSR is greatly improved; in fact, considering a Coarse Counter with a BIT_COARSE bit-length, the FSR of the TDC is equal to:

$$FSR = 2^{BIT_COARSE} \cdot T_{CLK} \quad (20)$$

The LSB is instead fixed by the TDL and is equal to:

$$LSB = t_p \quad (21)$$

thus solving the FSR vs. LSB trade-off. Moreover, we have to consider that the FSR of the TDL (i.e., $N_B \cdot t_p$) has to cover the range of T_{CLK} ; as a consequence of that, the TDL must have a number of buffers $N_B \geq T_{CLK}/t_p$.

2.2. Sub-Interpolation

The Sub-Interpolation is a technique used to reduce the quantization error (i.e., improving the resolution) of the TDL-TDC. In this sense, it exploits redundancy in the measurement process to lower the LSB as much as the desired delay beyond the minimum one offered by the technological node [22]. The proposed Sub-Interpolated TDL-TDC exploits spatial redundancy, i.e., the Super Wave Union (SuperWU), characterized by the use of more physical TDLs placed in parallel. If we have M TDLs in parallel, each one with a real number of buffers (taps) N_B , we obtain the so-called Virtual Tapped Delay-Line (V-TDL), composed of $N_V = N_B \cdot M$ “virtual” taps with a M -time faster propagation delay.

In practice, let's consider $m \in [1; M]$ replicas of the measurement of the same time interval, performed using M TDLs in parallel with $n_B \in [1; N_B]$ real bins each, characterized by a real “bin-by-bin” propagation delay $t_{p,B}[n_B]$ and FSR $T_{FSR} = \sum t_{p,B}$. In this situation, the sub-interpolation process generates a V-TDL composed of $n_V \in [1; N_V]$ “virtual bins” with a virtual “bin-by-bin” propagation delay $t_{p,V}[n_V]$, where:

$$\begin{cases} t_{p,V}[n_V] = \frac{1}{M} \sum_{m=1}^{m=M} t_{p,B}[n_B[m]] \\ n_V = \sum_{m=1}^{m=M} n_B[m] \end{cases} \quad (22)$$

Referring to Equation 22 and to [22], we can demonstrate that $N_V = N_B \cdot M$ and that $T_{FSR} = \sum t_{p,V}$; therefore the propagation delays of the real-bins belonging to the different TDLs get averaged each other, reducing the propagation delays of the virtual-bins of a factor M .

To obtain an effective reduction of the propagation delays on the virtual-bins, the replicas of the TDLs should be as uncorrelated as possible. For this reason, the START signal is increasingly delayed on each one of the M TDLs to have a minimum probability that it falls on similar real-bins. The parallel architecture resulting from the SuperWU algorithm is very suitable for FPGA's spatial-computing structures; moreover, the delay introduced on the START signal comes for free, since the routing process creates different paths the signal has to cross moving towards each TDL (the so-called “skew” time). We can see a Super Wave Union implementation of order $M = 3$ in Figure 16.

2.3. Calibration

Since in FPGAs there is no possibility to tailor a N -taps-long buffer-chain (where N refers to N_B if Sub-Interpolation is not performed or to N_V if Sub-Interpolation is performed) with constant propagation delays $t_p(n)$ ($n \in [1; N]$) (where t_p represents $t_{p,B}$ if Sub-Interpolation is not performed or $t_{p,V}$ if Sub-Interpolation is performed), but its internal structural blocks get adapted as a buffer chain, the bins will all have dispersed $t_p(n)$ values. In analogy with the ASICs, where a deviation of the bins' delays $t_p(n)$ from their ideal constant value is present, we will refer to the aforementioned dispersion as process (P) variation. Due to the P variations affecting each buffer, a periodic bin-by-bin calibration process is required, otherwise, the system would suffer from very high non-linearities [57]. This process consists of the creation of a Calibration Table (CT) listing all the propagation delays $t_p(n)$ of

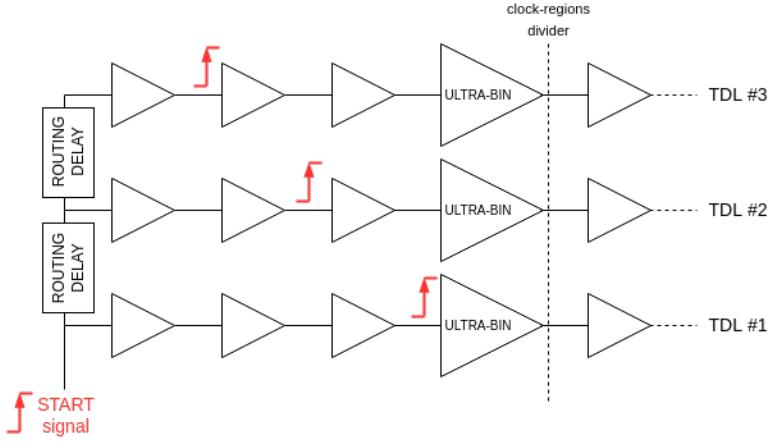


Figure 16: Super Wave Union implementation.

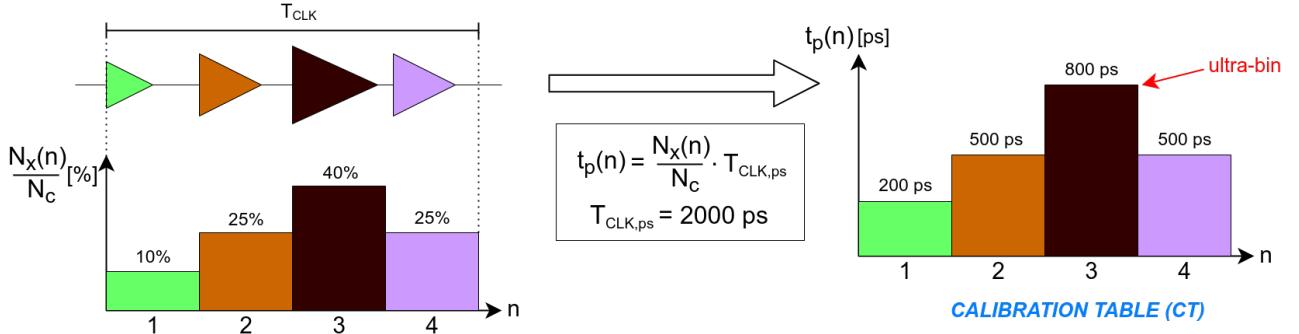


Figure 17: CDT and resulting CT, in the example case with $N = 4$ and $T_{CLK} = 2000 \text{ ps}$

each bin n (or “virtual” bin, if sub-interpolation is performed) composing the TDL (or V-TDL). Without calibration, the propagation delays would have unacceptably dispersed values, and a wrong time measurement would be performed by considering t_p constant and equal to the LSB. Moreover, due to the impossibility of implementing inside the FPGA analog hardware architectures for the voltage and temperature (VT) fluctuations, it is also necessary that the calibration process has a periodic refresh during the entire use of the TDC to update the correct propagation delay of each bin and compensate the VTs.

The CT is built by performing a Code Density Test (CDT) on the TDL (or V-TDL), which is T_{CLK} -wide if the Nutt-Interpolation is implemented and composed of a total of N real (or virtual) bins. The CDT consists of sending a set of random signals (N_C in total), characterized by a uniform distribution, to the TDL (or V-TDL) and counting how many samples $N_X(n)$ fall into each real (or virtual) bin. Finally, the $t_p(n)$ of each bin is obtained thanks to a normalization to T_{CLK} :

$$t_p(n) = \frac{N_X(n)}{N_C} \cdot T_{CLK}, \quad n \in [1; N] \quad (23)$$

Figure 17 graphically explains the procedure just described.

Due to the crossing of different clock regions, some bins can be particularly slow. These bins, characterized by the maximum value of the CT, will be referred to as “ultra-bins” and are responsible for worsening the single-shot precision of the TDC.

At Digital Electronics Laboratory (DigiLAB) at Politecnico di Milano, it has been experimentally found on a Xilinx 7-Series FPGA, for a non-sub-interpolated TDL, a minimum bin equal to 1 ps, a maximum bin (i.e., the “ultra-bin”) equal to 70 ps, and a mean bin equal to 12 ps [22][10].

On the other hand, in a Xilinx Ultrascale FPGA, the achieved minimum bin is equal to 1 ps, the maximum bin (i.e., the “ultra-bin”) is equal to 19 ps, and the mean bin is equal to 4.8 ps [16][17].

Once the CT is built, the Characteristic Curve (CC) of the device can be derived. The CC is nothing but the integration of the CT, thus representing, for each bin, the corresponding time measurement in

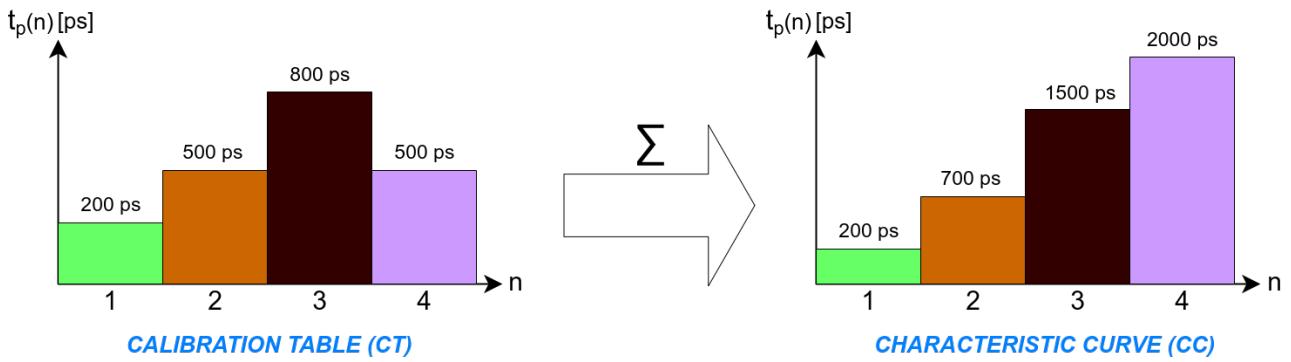


Figure 18: Graphical representation of the Calibration Table (CT) and the derived Characteristic Curve (CC).

picoseconds. In this way, by addressing the CC, the value in ps can be retrieved without performing the CT integration for each measurement.

Figure 18 reports a CT and the derived CC.

3. Main Work

The main goal of this thesis work has been to push even more the TDC performances, obtaining a drag-and-drop, tunable IP-Core, compatible with 20-nm Xilinx Ultrascale (XUS) and 16-nm Xilinx Ultrascale+ (XUS+) FPGAs. This TDC has been designed starting from an already existing implementation in the 28-nm Xilinx 7-Series (X7S) technological node [11]. The migration to XUS/XUS+, which is characterized by an improved scaling than X7S, has allowed increasing the frequency of the clock involved in the time measurements by an amount equal to one hundred MHz, resulting in a TDC with lower Dead-Time (see Paragraph 1.3.7). The scaling is also beneficial for the resolution since the propagation delays of the bins are lower than their X7S counterparts (see Equation 8); moreover, for the same reason, the “ultra-bins” phenomenon, described in Section 2.3, is slightly improved, resulting in an overall improvement of the single-shot precision of the TDC. Firstly, let’s briefly describe the blocks composing the main core of the TDC; then, each one of them will be described in detail, highlighting the improvements done in this thesis work.

3.1. Main blocks of the TDC

The [11] version of the TDC, implemented at the Digital Electronics Laboratory (DigiLAB) at Politecnico di Milano, has been implemented on the 28-nm X7S technological node, mainly in Artix-7 FPGAs, and it is a TDL-based TDC (TDL-TDC) [6][21]. Table 3 summarizes the achieved performances of the [11] implementation. It is a TIMESTAMP-type TDC, thus being composed of as many channels as the number of physical events to detect; besides, to break the trade-off between FSR and LSB, it exploits Nutt-Interpolation (see Section 2.1), thus composing each timestamp of a Fine part (i.e., the one calculated by the TDL) and of a Coarse one (i.e., the one calculated by a Coarse Counter). The system also exploits the SuperWU Sub-Interpolation to lower the LSB beyond the minimum propagation delay, given by the X7S technological node (see Section 2.2), and the bin-by-bin Calibration algorithm to compensate for PVT variations (see Section 2.3). The processing chain inside each channel starts with the V-TDL, which provides a thermometric code in output, as shown in Paragraph 1.5.3 . The thermometric code is then converted into a binary one by a Decoder, which is a very cumbersome module having a N -wide input, where N is the number of bins (or “virtual” bins) of the TDL (or V-TDL), and a $[\log_2(N) + 1]$ -wide output (see Section 3.3); for this reason, this module limits the maximum frequency of the system.

Feature	Artix-7 implementation
Resolution	36.6 fs
Precision	8.0 ps r.m.s.
Full-Scale Range	10.3 s
DNL	0.25 ps
INL	2.5 ps
Number of Chs	16
Channel Rate	150 MHz
Dead-Time	5 ns

Table 3: Performances of the Xilinx 7-Series TDC, implemented in an Artix-7 FPGA.

Once the binary code of the Fine measurement is obtained, Nutt-Interpolation is implemented by a module called Coarse Extension Core (CEC). This block attaches the Fine part to a Coarse part, the latter provided by a Coarse Counter sub-module that can be internal to the CEC or external. Furthermore, the CEC performs a Clock-Domain Crossing (CDC), meaning that the system goes from a faster clock to a slower one, allowing the data processing on all the following blocks to respect the timing constraints of the system. The following block in the processing chain is the Calibrator, which exploits storage mechanisms to store the CT and the CC. For this reason, this module can not work at the same frequency as the V-TDL, showing that the CDC performed by the CEC is mandatory. The CDC is done using a FIFO, which is written with the same clock controlling the V-TDL (i.e., TDC-clock) and read with a slower clock (i.e., SYS-clock). Until the FIFO is not full, it allows lowering the Dead-Time of the TDC, as explained in Paragraph 1.3.7. Finally, a module called Overflow Counter further extends the FSR of the TDC by counting the overflows of the Coarse Counter present in the CEC.

Figure 19 shows the block diagram of a generic channel of the TDC, composed of all the aforementioned modules. Since this thesis work consisted in modifying the internal blocks, the macroscopic structure of the channel has been maintained. We can see from Figure 19 that the implemented TDC relies on a “FIFO+Pipeline” structure since all the modules work in Pipeline and the CEC contains a FIFO that allows buffering at TDC-clock.

The communication between all the blocks is done using the AXI4-Stream protocol [1]. This protocol consists of a flux of data from the output of a module, called “master”, to the input of the following one, called “slave”. The main signals of the AXI4-Stream protocol are:

- TDATA, which is effectively the data to be sent;
- TVALID, which is a flag produced by the master, telling the slave that the incoming data is valid;
- TREADY, which is a flag produced by the slave, which tells the master that the module is ready to accept the data.

The AXI4-Stream protocol relies on the concept of Handshake between TVALID and TREADY. A Handshake transaction is reported in Figure 20, and takes place only when both TVALID and TREADY are equal to ‘1’.

All the blocks in Figure 19 are IP-Cores (see Section 1.7) connected to each other in the AXI4-Stream protocol, which are organized inside a Hierarchical IP-Core. The latter is nothing but a set of IP-Cores [50], and, with a simple drag-and-drop operation, an entire channel of the TDC can be promptly tested on different FPGAs. Since we are usually interested in the relative time distance between physical events, the TDC will be characterized by more than one channel, each one providing the respective timestamp with respect to a common absolute time reference (i.e., the power-on instant of the device).

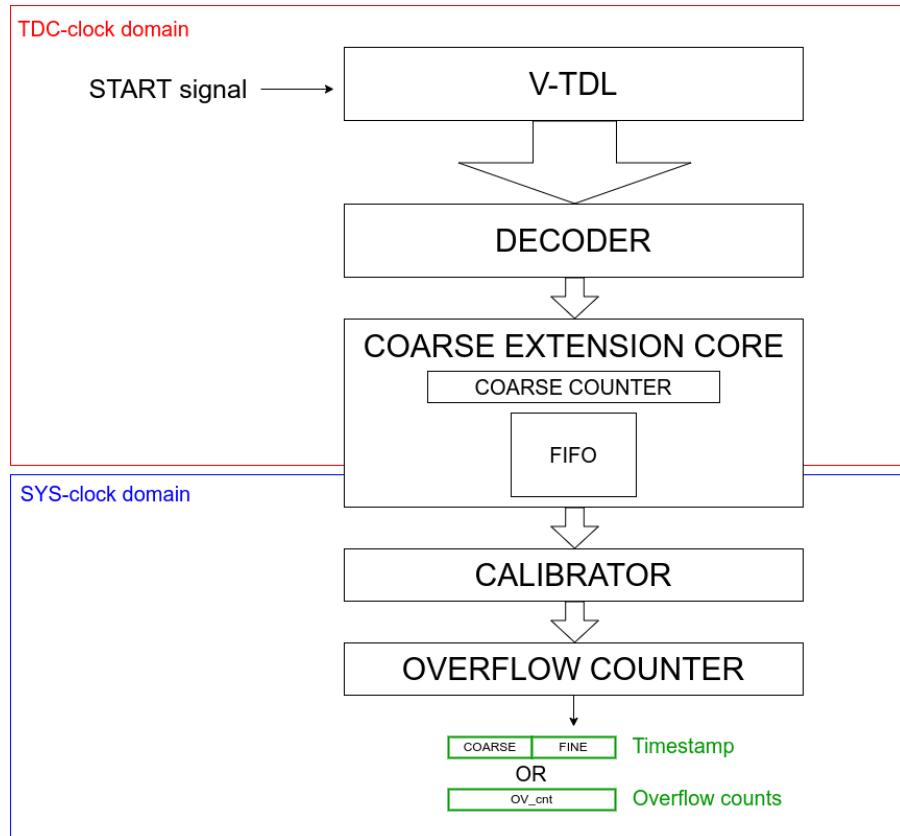


Figure 19: Schematic of the structure of a generic channel of the TDC.

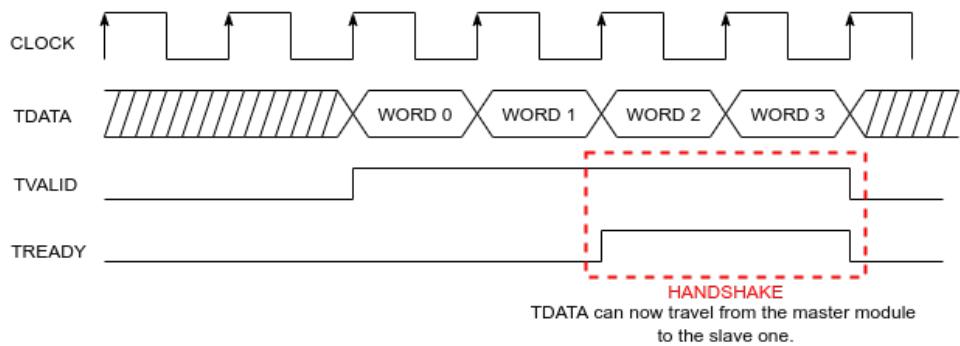


Figure 20: AXI4-Stream protocol.

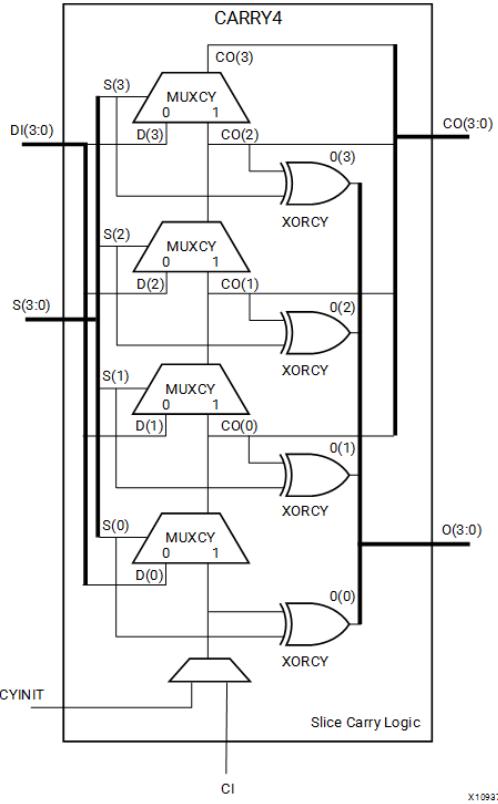


Figure 21: CARRY4 primitive structure.

Port	Direction	Width	Function
O	Output	4	Carry chain XOR general data out
CO	Output	4	Carry-out of each stage of the carry chain
DI	Input	4	Carry-MUX data input
S	Input	4	Carry-MUX select line
CYINIT	Input	1	Carry-in initialization input
CI	Input	1	Carry cascade input

Table 4: CARRY4 primitive Port descriptions.

3.2. Virtual Tapped Delay-Line (V-TDL)

The core of this thesis work relies on the migration of the V-TDL from the X7S to the XUS/XUS+ technology. The TDL has been obtained by cascading the carry chains of the adders [42], which are the more abundant and fast resources on the FPGA for this kind of implementation.

3.2.1. X7S Version

In the X7S version of the V-TDL, the elementary block used to create the chain structure is the CARRY4 primitive [46]. Figure 21 shows its internal structure, while Table 4 reports its Port descriptions.

As we can see, the CARRY4 primitive has two 4-bit outputs, the first being the CO port (i.e., the set of the output carry-signals), the second being the O port (i.e., the set of the XOR-outputs, representing the result of the sum). The four stages composing the CARRY4 primitive are nothing but four buffers (or taps) composing the TDL, cascaded thanks to the carry-out (CO) connections. Therefore,

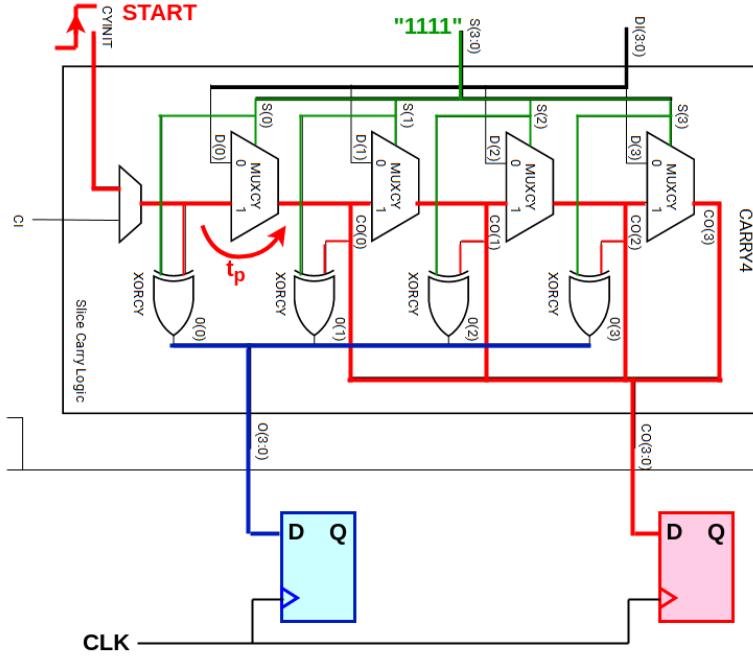


Figure 22: Resources employed in the CARRY4 primitive to correctly propagate the START signal.

according to the number of taps required to compose the entire TDL, a certain number of CARRY4 primitives will be cascaded by connecting the carry-output of the fourth stage (i.e., CO(3)) of the previous primitive to the carry-input port (CI) of the following primitive. The asynchronous input event to be measured (i.e., the START signal) is connected to the carry-input port (CYINIT) of the first primitive. The multiplexer's selection signal (S) of each stage is set to propagate the signal on CI along all the stages. In this way, as soon as the asynchronous input switches from '0' to '1', propagation of "ones" takes place along the carry-chain, thus causing a ' $0 \rightarrow 1$ ' transition on each CO output (or a ' $1 \rightarrow 0$ ' transition on each O output, which is nothing but the inverted and delayed CO output) with slightly different propagation delays. Each output (CO or O) of the chain is then connected to the D-input of a flip-flop (FF), used to sample the logic value of that specific tap, when the STOP signal occurs. Figure 22 shows the used resources of the CARRY4 primitive. It is worth saying that the CO and the O outputs can not be sampled at the same time. Let's now give an order of magnitude for the TDL length, based on the description given in Section 2.1. In an implementation exploiting the Nutt-Interpolation, the number N of bins (or "virtual" bins) is such that $N \geq \frac{T_{CLK}}{t_p}$. In the X7S technology, the available clock can run at most at 628 MHz in the Artix-7 FPGAs [55] and 741 MHz in Kintex-7 [51] and Virtex-7 [54]. However, due to the big dimension of the Decoder, the minimum clock period is limited to be $> 2.4\text{ ns}$ in order to satisfy the timing constraints of the system. The typical minimum value for t_p is instead equal to 16 ps in X7S FPGAs [11], leading to a minimum value for N equal to 150. Obviously, a faster clock would lead to a shorter TDL, allowing achieving a more compact architecture with a smaller Decoder and a smaller Calibrator.

The STOP signal of the TDL is a user-chosen rising-edge of the clock signal with period $> 2.4\text{ ns}$; when the STOP signal occurs, a snapshot of the TDL state (i.e., the number of buffers hit by the propagation of "ones" up to that moment) is taken and, therefore, a thermometric code is generated at the output of the chain of FFs. Figure 23 graphically shows the propagation mechanism of the TDL.

In the following description, the words written in italic capital letters refer to the Hardware Description Language (HDL) generics used in the code, which are settable by the IP-Core's Graphical User Interface (GUI). This allows having a highly-tunable TDL. As explained in Section 2.2, the SuperWU Sub-Interpolation is performed, obtaining a V-TDL composed of *NUMBER_OF_TDL* TDLs in parallel, where each of them is *NUM_TAP_TDL*-taps long. This results in a V-TDL composed of *NUMBER_OF_TDL*·*NUM_TAP_TDL* "virtual" taps. *BIT_SMP_TDL* FFs will be placed at the output

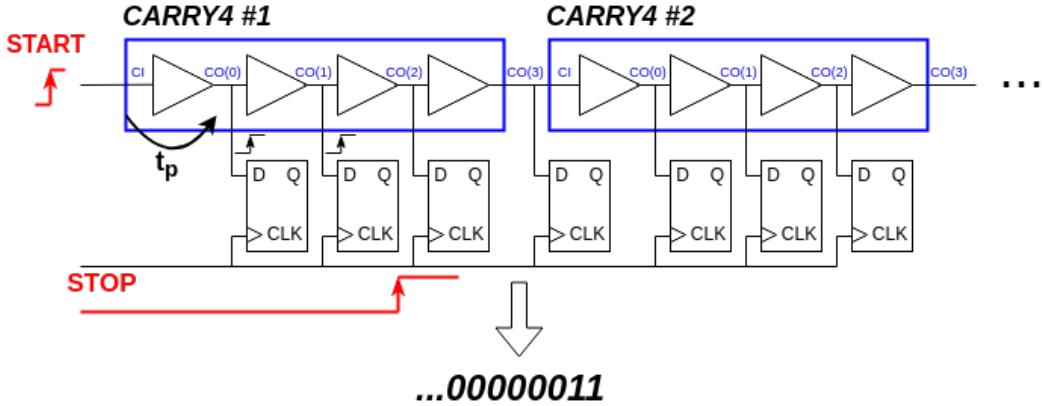


Figure 23: CARRY4-based TDL propagation mechanism.

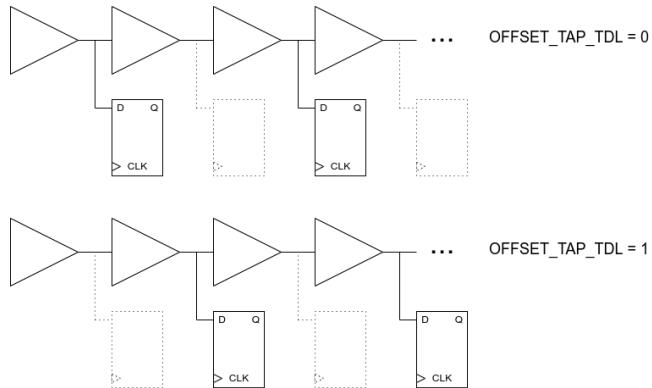


Figure 24: Concept of decimated sampling and offset.

of each TDL. Since $BIT_SMP_TDL \leq NUM_TAP_TDL$ holds, it is possible either to sample the output of each buffer, by placing a number of FFs equal to the number of taps, or to sample just a decimated number of taps, by placing fewer FFs than buffers. In the latter case, hardware resources are saved, but a thermometric code composed of a lower number of bits is generated; hence, the resolution (LSB) is worsened. This decimation process has been provided to allow the implementation of a TDL-TDC characterized by lower area occupancy at the expense of a lower resolution. Obviously, this feature has to be used without Sub-Interpolation. The sampling of the V-TDL is also managed by the generics $TYPE_TDL_i$ and $OFFSET_TAP_TDL_i$, where i is a value between 0 and $NUMBER_OF_TDL-1$ indicating which physical TDL we are referring to. With $TYPE_TDL_i$ it is possible to choose which taps, CO or O, of the CARRY4 primitive we want to sample, for the i -th TDL. Then, if we are in the decimated case (i.e., $BIT_SMP_TDL < NUM_TAP_TDL$), with the generic $OFFSET_TAP_TDL_i$ it is possible to set an initial offset in the sampling of the chains, which means that the first FF of the i -th TDL is placed after $OFFSET_TAP_TDL_i$ taps rather than on the first one. The concept of decimation and offset is exemplified in Figure 24, where we have a certain NUM_TAP_TDL with $BIT_SMP_TDL = NUM_TAP_TDL/2$, and the two different cases with $OFFSET_TAP_TDL = 0$ (at the top) and $OFFSET_TAP_TDL = 1$ (at the bottom).

In Paragraph 1.5.3, we have said that the STOP signal, which saves the output of the TDL in the FFs, is the rising clock-edge following the arrival of the START event. That was a general case, whereas the proposed architecture, based on the AXI4-Stream protocol, relies on a “VALID” mechanism that, if it is ‘1’, signals to the next module that the thermometric code present on TDATA is the Fine measurement. The TVALID is generated by the $0 \rightarrow 1$ transition of one tap of the TDL, which is selectable by the user. Thanks to this mechanism, it is possible to compensate skews introduced by the automatic place&route process. As we saw in Section 2.1, in an implementation exploiting Nutt-Interpolation, the measured Fine interval can have T_{CLK} as the maximum value; therefore, the Fine measurement will be entirely contained in the TDL, avoiding the saturation of the latter (i.e., all the bins are hit by

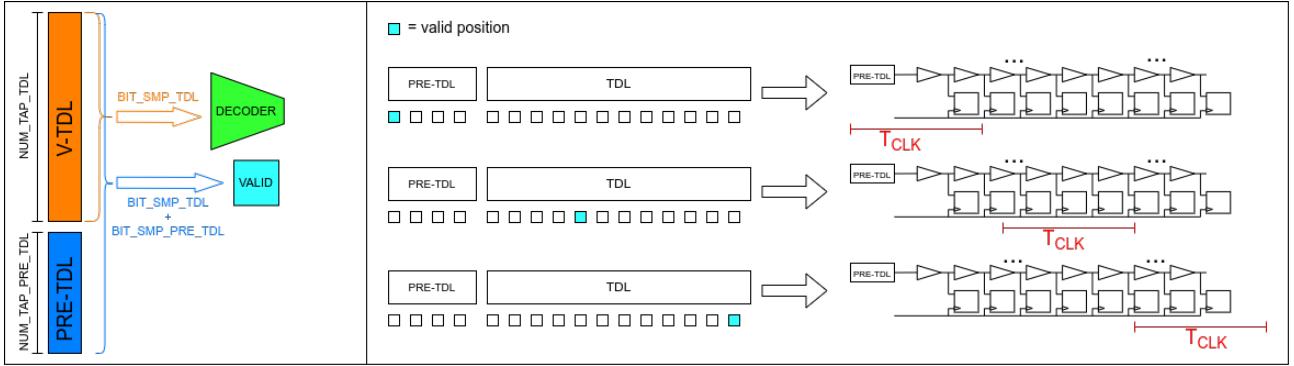


Figure 25: Valid mechanism of the TDL and role of the PRE-TDL.

the START signal). As a consequence, since the TDL is implemented to be longer than T_{CLK} , choosing the “VALID” position on different taps is equivalent to sampling different T_{CLK} -wide parts of the TDL. If the user decides to select the TVALID assertion on one of the first taps, a T_{CLK} -wide portion at the beginning of the TDL will be sampled to retrieve the Fine measurement; therefore, the output TDATA (i.e., the thermometric code) will be composed of just a few “ones” since the asynchronous input signal still has not propagated that much along the chain. On the other hand, if the user selects the TVALID assertion on the last few taps, a T_{CLK} -wide portion at the very end of the buffer-chain will be sampled, and the thermometric code will be composed mostly of “ones” since the input signal has traveled as far as these last taps. The same reasoning applies to all the other intermediate positions. To further improve the TVALID selection, a PRE-TDL has been introduced before each TDL. The PRE-TDL is composed of `NUM_TAP_PRE_TDL` taps, of which just `BIT_SMP_PRE_TDL` taps are sampled. The PRE-TDL is not used for measuring the incoming signal, but just to anticipate the acquisition of the TVALID before the acquisition of the asynchronous input signal. Indeed, if we choose the TVALID assertion on PRE-TDL’s tap, the thermometric code, given by the actual TDL only, would have very few “ones” since the input signal would have traveled just on the PRE-TDL and on the very first TDL’s taps. Figure 25 shows the concepts just explained, highlighting the different sampled portions of TDL chosen by the “VALID” mechanism. This mechanism allows the TVALID signal to have a high logical value for just one clock period, whose rising-edge is nothing but the STOP signal. The great usefulness given by the “VALID” mechanism is being able to sample a convenient portion of the TDL, which is free from “ultra-bins” and therefore capable of providing highly-precise measurements.

The TVALID selection is performed in two different modes, based on the generic `DEBUG_MODE`. If `DEBUG_MODE = FALSE`, the “VALID” position is statically chosen by the generic `VALID_POSITION_TAP_INIT`, which is a value between 0 and `BIT_SMP_PRE_TDL+BIT_SMP_TDL-1`. Conversely, if `DEBUG_MODE = TRUE`, the “VALID” position is chosen at run-time, by Port. In this case, we can select just a few of the `BIT_SMP_PRE_TDL+BIT_SMP_TDL` sampled taps from which we could extract the TVALID, and it is done thanks to the generics `MIN_VALID_TAP_POS`, `MAX_VALID_TAP_POS`, and `STEP_VALID_TAP_POS`. With these generics, we choose just some FFs along the chain from which we can select the TVALID, thus performing a decimation and avoiding using large multiplexers that may reduce the timing performance. Among these few FFs, the final position is selected with the Port “ValidPositionTap”. Figure 26 shows, with an example, the explained working principle in `DEBUG_MODE = TRUE` and with `MIN_VALID_TAP_POS = 0`, `MAX_VALID_TAP_POS = 4`, `STEP_VALID_TAP_POS = 2`. Since, if we use the SuperWU Sub-Interpolation algorithm, there are more TDLs in parallel, we must select one single TDL among them in which the aforementioned “VALID” selection process is performed. This is done statically by the generic `VALID_NUMBER_OF_TDL_INIT` if `DEBUG_MODE = FALSE`, or by the Port “ValidNumberOfTDL” if `DEBUG_MODE = TRUE`.

Finally, there is the possibility of just simulating the TDL rather than implementing it, thanks to the generic `SIM_VS_IMP`. If `SIM_VS_IMP = "SIM"`, the TDL is not physically implemented with the CARRY4 primitive, but a fictitious buffer-chain is created just for a test-bench scope. The simulated

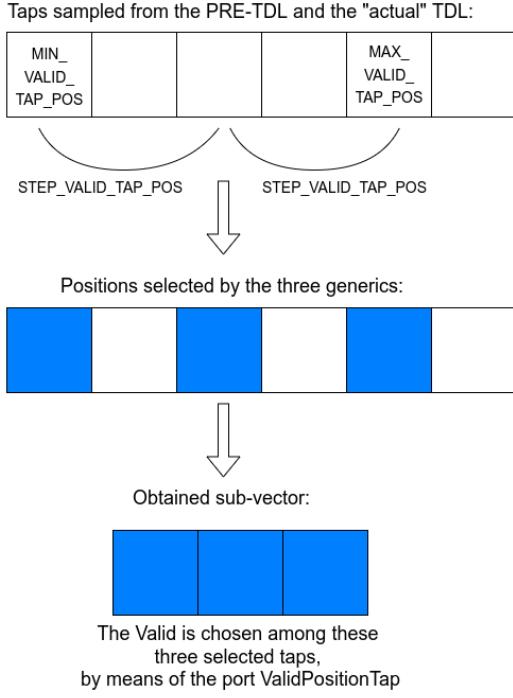


Figure 26: Valid selection in *DEBUG_MODE* = TRUE.

delays of the fictitious buffers are imported from two .txt files, *FILE_PATH_NAME_O_DELAY* and *FILE_PATH_NAME_CO_DELAY*, containing their estimated delays.

3.2.2. XUS/XUS+ Version

In this thesis work, the portability of the V-TDL has been improved, making it compatible also with 20-nm Xilinx Ultrascale (XUS) and 16-nm Ultrascale+ (XUS+) technological nodes. Let's describe this new version of the V-TDL, which has been entirely implemented in this work.

Major differences between X7S and XUS/XUS+ Other than the lower power consumption, which is a direct benefit of the improved scaling, the other main differences are structure-related, thus involving Configurable Logic Blocks (CLBs) and slices [46] [47]. The CLBs are resources capable of implementing general-purpose combinatorial and sequential circuits in FPGA, while the slices are the internal parts of the CLBs. Slices and CLBs are arranged in columns throughout the device, and they can easily connect to each other, thus creating large functions. Both X7S and XUS/XUS+ CLBs provide:

- Real 6-input look-up table (LUT) technology;
- Dual 5-input LUT (LUT5) option;
- Distributed Memory and Shift Register Logic capability;
- Wide multiplexers (MUXes) for efficient utilization;
- Dedicated high-speed carry logic for arithmetic functions.

The XUS/XUS+ CLB also has dedicated storage elements that can be configured as FFs or latches with flexible control signals.

The main structural difference between X7S and XUS/XUS+ is internal to the CLB, thus residing on the Slices. Table 5 shows the main differences inside a single CLB.

As we can see, in the XUS/XUS+ technology two independent slices are now combined in a single, bigger slice, in order to have better efficiency in the routing process on the FPGA's fabric. In particular, the carry logic is extended from 4 bits to 8 bits, achieving better routing and faster arithmetic functions. One single 8-bit carry-chain is present in a CLB. This is why the XUS/XUS+ technology is compatible with the CARRY8 primitive to perform arithmetic functions rather than the CARRY4

Table 5: Differences in a single CLB between Xilinx 7-Series and Xilinx Ultrascale technology nodes.

Features	X7S	XUS
Number of slices	2	1
Number of 6-input LUTs in a single slice	4	8
Number of FFs in a single slice	8	16

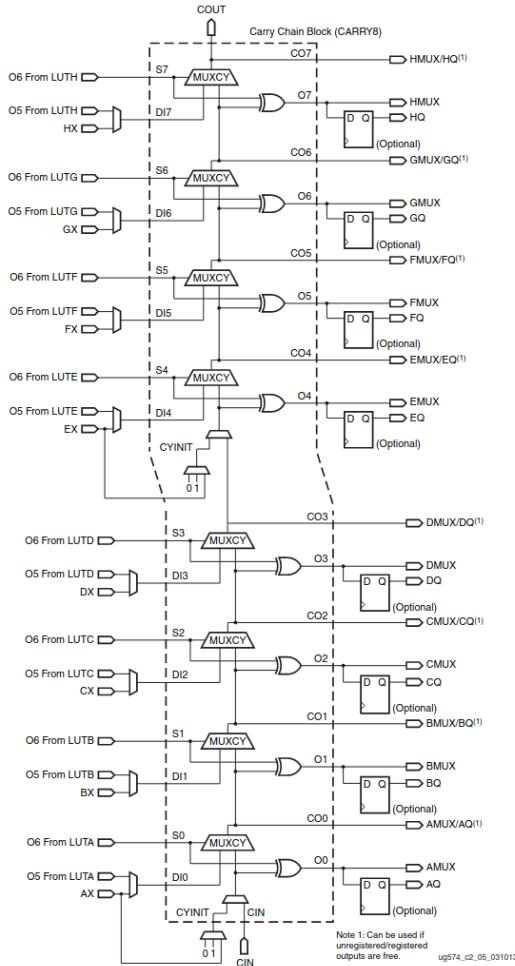


Figure 2-4: Fast Carry Logic Path and Associated Elements

Figure 27: CARRY8 primitive structure.

of the previous generation. Figure 27 shows the internal structure of the CARRY8 primitive, while Table 6 reports its Port descriptions.

As we can see from Figure 27, the CARRY8 primitive can be configured either as a single 8-bit carry-chain or as two independent 4-bit carry-chains; the first option has been chosen in this thesis work. In the XUS/XUS+ technology, the CO and the O outputs can be sampled at the same time.

XUS-TDL structure In analogy with the X7S version of the TDL, a single CARRY8 block implements eight buffers (or taps) of the TDL and, as before, more primitives are cascaded by connecting the last carry-out (CO(7)) of the previous primitive with the carry-in (CI) of the following one. As before, by choosing the value of the generic *TYPE_TDL* from the IP-Core’s GUI, either the CO outputs or the O outputs of the TDL can be sampled by the FFs. Figure 28 shows the propagation mechanism of the TDL.

The sampling process is managed in the same way as before, with the possibility of decimating the FFs along the chain and putting an initial offset in their positioning. The same “VALID” mechanism for choosing the clock period in which the snapshot of the TDL will be done has also been maintained.

Port	Direction	Width	Function
CI	Input	1	Carry input for 8-bit carry or lower portion of 4-bit carry.
CI_TOP	Input	1	Upper carry input when CARRY_TYPE=DUAL_CY4.
CO<7:0>	Output	8	Carry-out of each stage of the carry chain.
DI<7:0>	Input	8	Carry-MUX data input.
O<7:0>	Output	8	Carry chain XOR general data out.
S<7:0>	Input	8	Carry-MUX select line.

Table 6: CARRY8 primitive Port descriptions.

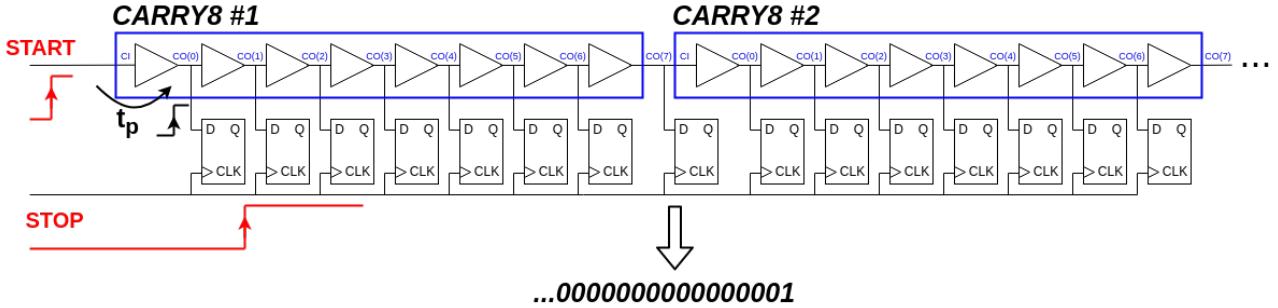


Figure 28: CARRY8-based TDL propagation mechanism.

Thanks to the benefits of the scaling, the Decoder has been pushed in its operating frequency, thus allowing the TDC-clock to run up to 500 MHz.

3.2.3. IP-Core Integration of both V-TDL versions

The X7S V-TDL described in Paragraph 3.2.1 has been integrated into the same IP-Core of the XUS/XUS V-TDL described in Paragraph 3.2.2, increasing the portability of the system on different FPGA technologies. In particular, a new generic, *XUS_VS_X7S*, has been added to the IP-Core GUI. Thanks to this generic, the user can choose, in the instantiation stage, the technological node of the V-TDL. Figure 29 shows the V-TDL IP-Core.

3.3. Decoder

Since this module has not been modified in this thesis work, its only version will be described in the following.

The Decoder has the task of converting the *BIT_UNDECO*-wide thermometric code (coming from the V-TDL; i.e., *NUMBER_OF_TDL*·*BIT_SMP_TDL*) into a *BIT_SUBINT*-wide binary code [33].

The conversion is done in a Pipeline structure using a Thermo-to-Binary (T2B) engine. Let's briefly describe the working principle of the T2B engine. Firstly, it detects the position of the rising-edge or falling-edge of the input signal by summing the '0's or the '1's of the thermometric data, respectively. Then, thanks to a pipelined tree structure, it transforms the thermometric code into a binary one, by summing all the '1's in the case of rising-edge sensitivity to the input signal, or all the '0's in the case of falling-edge sensitivity. The following relation holds:

$$BIT_SUBINT = \lfloor \log_2(BIT_UNDECO) \rfloor + 1 \quad (24)$$

Table 7 shows the mechanism just explained, in the simple case of a 4-bit-wide thermometric code coming from the TDL, thus resulting in a 3-bit-wide output binary code.

The Decoder also has the task of performing Sub-Interpolation by summing all the binary codes provided by each one of the "real" TDLs, using a pipelined Tree Adder (TA). It is possible to dynamically choose the number of TDLs involved in the sub-interpolation employing the so-called

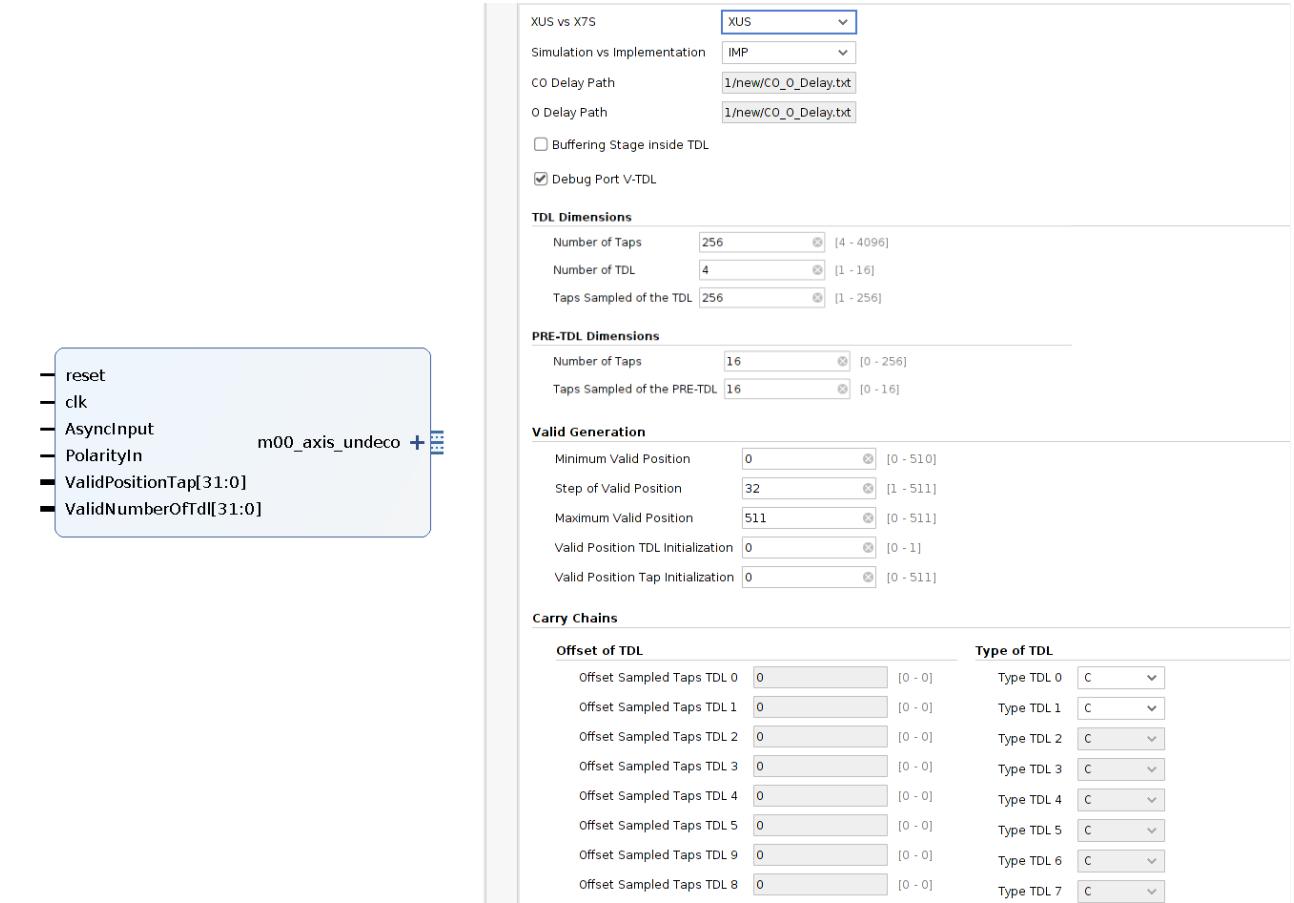


Figure 29: V-TDL IP-Core, highlighting the new generic XUS_VS_X7S.

Thermometric code	Binary code	Thermometric code	Binary code
0000	000	1111	000
0001	001	1110	001
0011	010	1100	010
0111	011	1000	011
1111	100	0000	100

Table 7: Thermometric to Binary conversion in the case of rising-edge sensitivity to the input signal (on the left) and with falling-edge sensitivity (on the right).

SubInterpolationMatrix	V-TDL
000	-
001	TDL #1
010	TDL #2
100	TDL #3
011	TDL #1 + TDL #2
101	TDL #1 + TDL #3
110	TDL #2 + TDL #3
111	TDL #1 + TDL #2 + TDL #3

Table 8: Selection of the TDLs to involve in the Sub-Interpolation using the Sub-Interpolation Matrix.

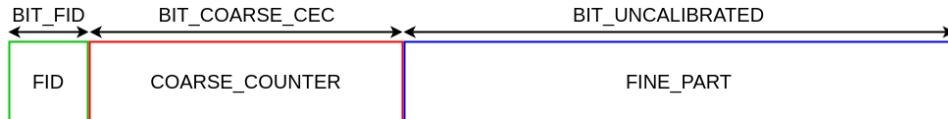


Figure 30: Coarse Extension Core data structure.

Sub-Interpolation Matrix. Table 8 shows a case where three TDLs are placed in parallel, thus giving eight possible combinations. Obviously, by sub-interpolating all three TDLs, the best resolution is achieved, according to the SuperWU algorithm.

3.4. Coarse Extension Core

In the Coarse Extension Core (CEC) module, Nutt-Interpolation is carried out to break the trade-off between LSB and FSR [25]. A *BIT_COARSE_CEC*-wide Coarse-Counter is in charge of counting the number of clock periods elapsed from the power-on instant of the system up to the generation of the Fine measure, accomplished by the V-TDL. Overflow-detection is also implemented and will be used by one of the following modules (i.e., the Overflow Counter) to extend even more the FSR (see Section 3.6). The aforementioned overflow detection is performed by a sub-module called TreeComparator, working in a Pipeline structure in order to minimize timing issues, which compares the value of the Coarse-Counter with a reference value equal to:

$$OVERFLOW_REF = 2^{BIT_COARSE_CEC} - 1 - NUM_STAGES \quad (25)$$

In this way, as soon as the Coarse-Counter reaches its maximum value (i.e., $2^{BIT_COARSE_CEC} - 1$), an overflow condition is reported on the next clock cycle. The subtraction of *NUM_STAGES* is required to take into account the delay introduced by the pipeline stages of the TreeComparator.

Figure 30 shows the structure of the data provided by the CEC, which is composed of three fields. The *FINE_PART* field contains the binary code provided by the Decoder; the *COARSE_COUNTER* field contains the number of the elapsed clock cycles; the *FID* field contains the information about the occurrence of an overflow in the Coarse-Counter. *FID* = '1' indicates that no overflows happened, and the *COARSE_COUNTER* and the *FINE_PART* fields contain the valid measurement; on the other hand, *FID* = '0' signals an overflow occurrence, and the other two fields will contain their older values, signaling that the measurement is not valid.

The CEC also performs Clock Domain Crossing (CDC), meaning that the system goes from a faster clock to a slower one. Indeed, it contains a FIFO that prevents losing consecutive measurements, and correctly manages the case in which a valid measure and an overflow occur concurrently. As said in Section 3.1, the FIFO is written and read under two different clock domains; the writing process is done under the TDC-clock domain (i.e., the same clock feeding the V-TDL and the Decoder), while the reading process is done under the SYS-clock domain, which has a lower frequency that is constrained by the Calibrator.

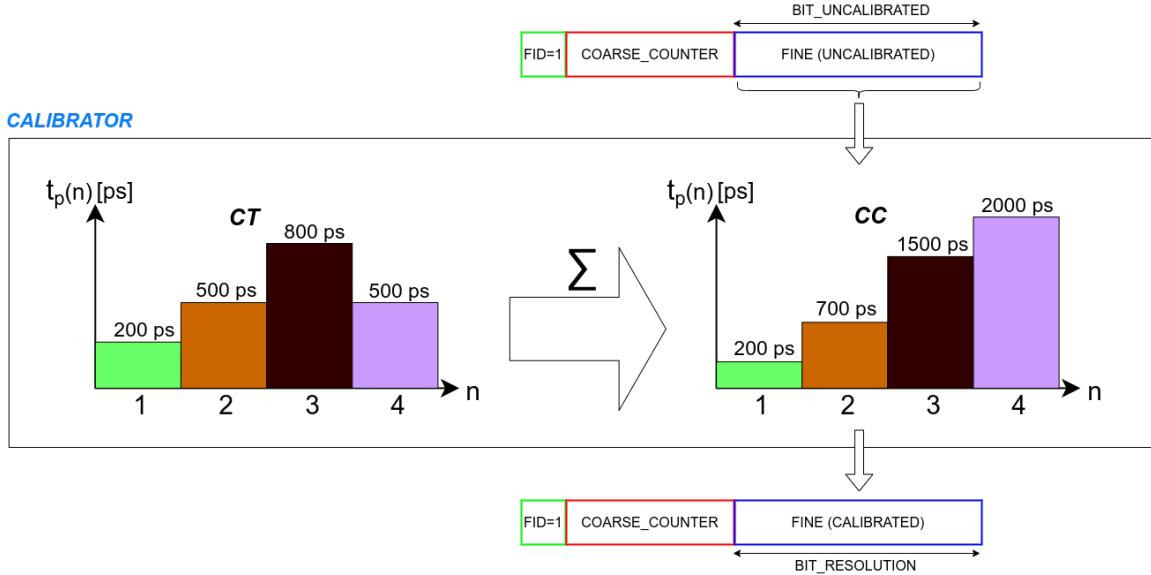


Figure 31: Calibrator working principle.

This CEC implementation guarantees a maximum channel-rate equal to:

$$MaximumChRate = f_{SYSclock} - \frac{f_{TDCclock}}{2^{BIT_COARSE_CEC}} \quad (26)$$

where $\frac{f_{TDCclock}}{2^{BIT_COARSE_CEC}}$ is the rate of the managed overflows.

3.5. Calibrator

This module has the same structure as in the old TDC and is in charge of keeping track of the PVT variations affecting each bin of the V-TDL [57]. As explained in Section 2.3, it performs calibration by creating a Calibration Table (CT) and a Characteristic Curve (CC), where the CC represents the “bin-to-picoseconds” conversion. The “Fine” field (uncalibrated) of the CEC is used to address the CC in the Calibrator, thus obtaining the value in picoseconds. Therefore, a “Fine” calibrated field is provided at the output. Let’s briefly describe the generics of this module. *BIT_CALIBRATION* sets the number of samples used to build the CT ($N_C = 2^{BIT_CALIBRATION}$). *BIT_RESOLUTION* is instead the length of the calibrated Fine measurement, in picoseconds, that is exposed at the output of the module. The calibration on the Fine data is performed only if *FID*=’1’, which means that a valid measure is coming from the CEC; otherwise, if *FID*=’0’, it means that an overflow is coming from the CEC, and the data is passed unaltered through the Calibrator. Figure 31 shows the overall data-flow and working principle of the Calibrator.

3.6. Overflow Counter

3.6.1. X7S Version

The Overflow Counter (OC) is strongly related to the CEC. Indeed, while the CEC just signals the presence of an overflow in the Coarse-Counter (by putting *FID*=’0’), the OC counts the number of overflows coming from the CEC, allowing to extend the FSR of the TDC up to some days. The working principle of the OC relies on the value of the *FID* field of the input data (i.e., *s00_timestamp_tdata*). *FID*=’1’ signals a valid timestamp, which passes unaltered through the module; therefore, the output data (i.e., *m00_beltbus_tdata*) is equal to the *s00_timestamp_tdata*. On the other hand, *FID*=’0’ signals that an overflow is coming from the CEC, and a signal *Overflow_cnt* is incremented by one and sent in output, along with *FID*=’0’. Obviously, *Overflow_cnt* has a bit-length equal to *BIT_COARSE_CEC*.

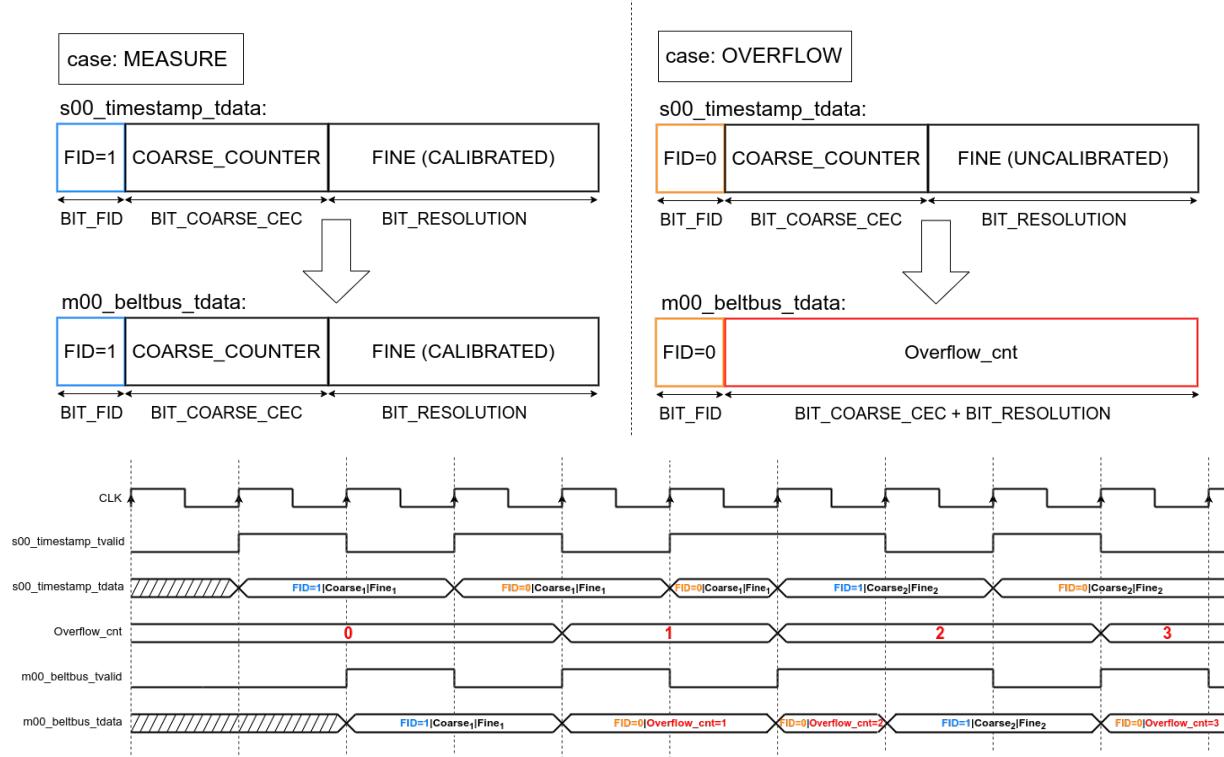


Figure 32: Data structure (at the top) and Timing Diagram (at the bottom) of the Overflow Counter.

+ $BIT_RESOLUTION$. Figure 32 graphically shows with a timing diagram the data structure and the working principle of this module.

3.6.2. XUS/XUS+ Version

A new solution for the OC has been implemented in this thesis work to improve the timing performance (i.e., the Worst Negative Slack, WNS) of the TDC. Referring to Figure 30 (Section 3.4), by shortening the data passing through the FIFO, a TDC with better timing performances can be achieved, thus leading to a smaller Decoder, a higher measure-rate, and a lower Dead-Time. Since $BIT_UNCALIBRATED$, also equal to BIT_SUBINT , is fixed by the Sub-Interpolation and the desired LSB, it is convenient to lower BIT_COARSE_CEC . In fact, BIT_COARSE_CEC is now used with a lower value than the one (i.e., $BIT_COARSE_CEC_{old}$) giving the desired FSR (i.e., $2^{BIT_COARSE_CEC_{old}} \cdot T_{CLK}$). However, a problem arises since, by having BIT_COARSE_CEC with a lower value, the overflow occurrences happen no more every FSR but are way more frequent. This lowers the Maximum Channel Rate, as shown by Equation 26 (Section 3.4). The new version of the OC, developed in this thesis work, aims to solve this issue, by introducing an output Coarse field with a different bit-length to the input Coarse field. The generic for selecting the coarse bit-length of the input data is the same as the previous modules (i.e., BIT_COARSE_CEC); the generic for the bit-length of the output data is instead BIT_COARSE . The following relation holds:

$$BIT_COARSE_CEC \leq BIT_COARSE \quad (27)$$

Since the $BIT_COARSE_CEC = BIT_COARSE$ case behaves exactly as in the older version of the OC, described in Paragraph 3.6.1, and since we just said that it would cause problems to the channel's management of the data, we will now describe just the $BIT_COARSE_CEC < BIT_COARSE$ case. Figure 33 shows the IP-Core package and the generics of the OC.

To correctly manage the counting of the overflows, an *AuxiliaryCounter* signal, which is ($BIT_COARSE_DIFFERENCE = BIT_COARSE - BIT_COARSE_CEC$)-wide, is in charge of counting the overflows in input (i.e., the data with $FID='0'$ coming from the CEC) and it is appended to the



Figure 33: Overflow Counter IP-Core.

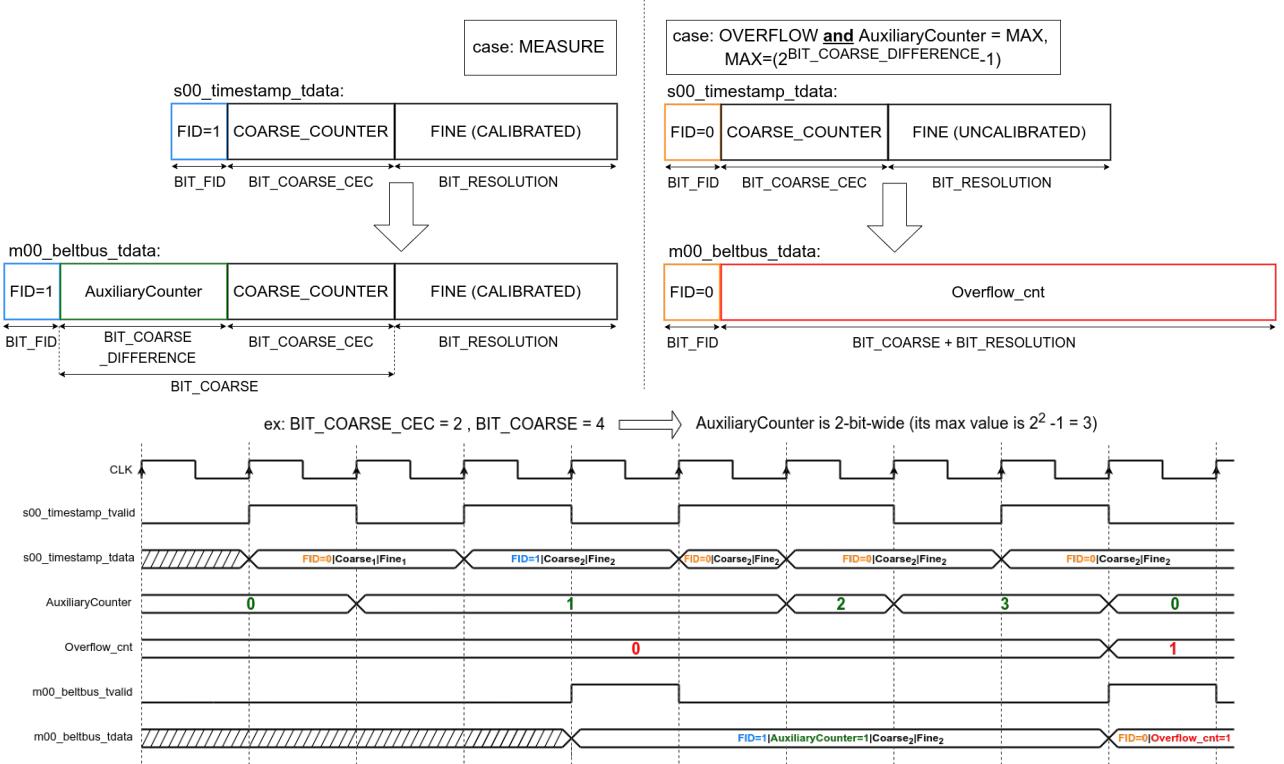


Figure 34: Data structure (at the top) and Timing Diagram (at the bottom) of the improved Overflow Counter.

Coarse part of the input data. In this way, the output data will be [(FID = '1') | AuxiliaryCounter | CoarseIn | Fine] if we have a valid measure as an input (i.e. FID = '1'); otherwise, if AuxiliaryCounter has reached its maximum value (i.e. $2^{**\text{BIT_COARSE_DIFFERENCE}-1}$) and we have a new overflow in input (i.e., FID = '0'), AuxiliaryCounter resets to 0, and the signal Overflow_cnt is increased by one. In this case, the output data will be [(FID = '0') | Overflow_cnt]. Obviously, Overflow_cnt has a bit-length equal to BIT_COARSE + BIT_RESOLUTION. Figure 34 shows the data structure and the working principle of the improved OC, in a simplified example using BIT_COARSE_CEC = 2 and BIT_COARSE = 4.

It is clear that, since the update of Overflow_cnt only happens when AuxiliaryCounter overflows, there are fewer FID='0' output transactions than the input ones. The lower channel-rate problem is thus solved, together with the initial timing errors the CEC suffered from.

4. Exploration of a new TDL solution

A new solution for implementing the V-TDL has been explored in this thesis work. This solution exploits Digital Signal Processor (DSP) blocks to create the TDL structure rather than the usual carry primitives [15].

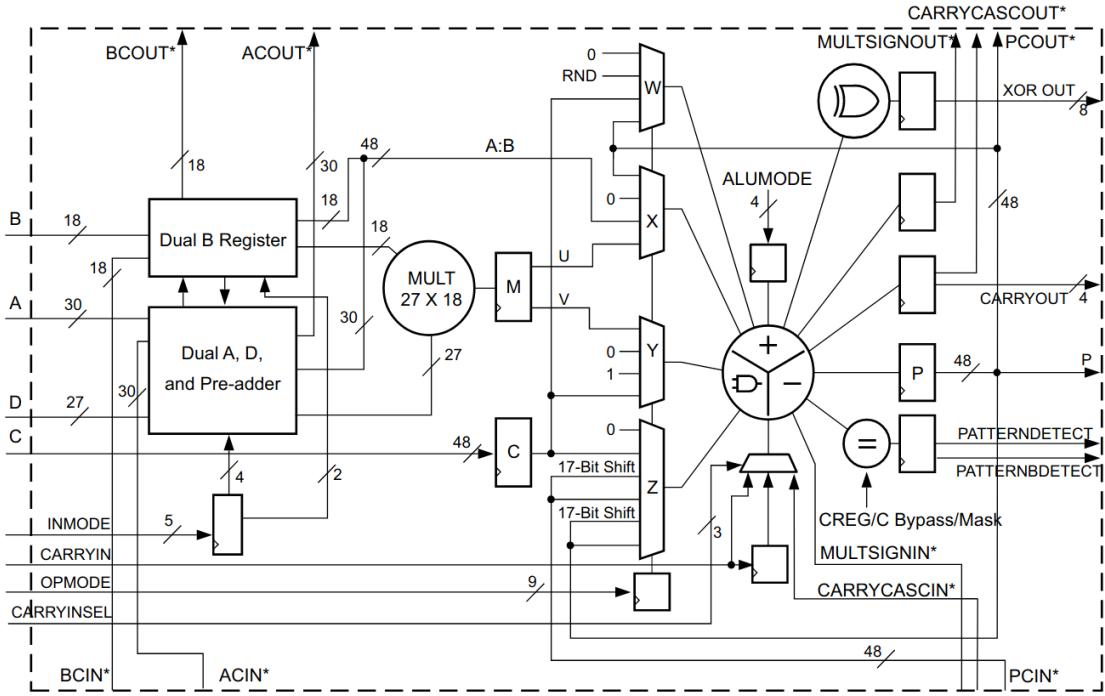


Figure 35: DSP48E2 structure.

4.1. Xilinx Digital Signal Processor (DSP)

FPGAs are suitable for Digital Signal Processing applications since they can implement custom, fully-parallel algorithms. The basic operation performed by DSPs is the MAC (Multiply and Accumulate), which consists of the product between two numbers, and its addition in the accumulator. In hardware, it is implemented as a multiplier followed by an adder and a register; however, in FPGA, such implementation employs a lot of resources and has a low processing speed. For this reason, Xilinx decided to place dedicated DSP blocks on the FPGA fabric.

The X7S technology uses the DSP48E1 slice as DSP primitive [48], while the XUS/XUS+ uses the DSP48E2 slice [53]. Since there are minor differences between the two aforementioned blocks, not relevant for the TDL implementation, we will refer just to the DSP48E2 block to describe both. Figure 35 shows the internal structure of the DSP48E2 slice, consisting of a 27-bit pre-adder, a 27 x 18 multiplier, and a flexible 48-bit Arithmetic Logic Unit (ALU) that serves as a post-adder/subtractor, accumulator, or logic unit.

We can see from Figure 35 that there are four direct inputs (i.e., A, B, C, D) and a 48-bit-wide direct output (i.e., P), which is the result of the operation performed by the ALU. An input carry (i.e., CARRYIN) and an output one (i.e., CARRYOUT) are also present, along with an output port for pattern detection (i.e., PATTERNDETECT/PATTERNBDETECT) and the output of a XOR gate (i.e., XOR OUT). Furthermore, five input (i.e., ACIN, BCIN, PCIN, CARRYCASCIN, MULTSIGNIN) and output ports (i.e., ACOUT, BCOUT, PCOUT, CARRYCASCOUT, MULTSIGNOUT) are present, with the function of cascading more DSP slices. All the remaining inputs are control signals:

- ALUMODE: it controls the operation performed by the ALU, which can be either a sum, a subtraction, or a logic operation;
- CARRYINSEL: it decides whether the direct carry-in or the cascaded one is fed to the ALU;
- OPMODE: it contains fields for W, X, Y, and Z multiplexer selects. Therefore, this signal decides which input signals are used as the factors for the ALU operation;
- INMODE: it controls the functionality of the pre-adder.

These control signals require a specific bit-mask to perform all the different functionalities, as can be seen in the DSP48E2 User Guide [53].

Finally, it is worth saying that the DSP48E2 has three different working modes. Indeed, thanks to the

Single-Instruction-Multiple-Data (SIMD) mode, the 48-bit ALU (i.e. the first, default, working mode) also supports dual 24-bit (i.e., the 48-bit ALU gets split into two 24-bit ALUs working in parallel) or quad 12-bit (i.e., the 48-bit ALU gets split into four 12-bit ALUs working in parallel) arithmetic operations.

4.2. DSP-based TDL

The propagation mechanism of the START along the DSP's output bits (i.e., the taps of the TDL) relies on the subtraction operation performed by the ALU. Let's see more in detail the ports and the control signals involved. The dual A and B registers, along with the pre-adder and the multiplier, are not involved in this implementation; therefore, they are bypassed by all the signals. Besides, the D input signal and the INMODE control are not used. The ALUMODE signal is set to make the ALU perform the following operation:

$$P = Z - (W + X + Y + CIN) \quad (28)$$

where CIN is set by CARRYINSEL to be the direct CARRYIN signal, which is put equal to zero. Z, W, X, and Y are the input signals, multiplexed by OPMODE. The selection is such that:

- Z transmits the C input signal to the ALU;
- W transmits a constant vector of zeroes to the ALU;
- X transmits the concatenated A and B input signals (i.e., A:B) to the ALU;
- Y transmits a constant vector of zeroes to the ALU.

The A:B vector is the one containing the asynchronous input signal (START) the TDC has to measure, while a constant vector of zeroes is assigned to the C port.

In this thesis work, the dual 24-bit SIMD mode has been used; therefore, a single DSP-chain is equivalent to two TDLS in parallel receiving the same START signal. This mode has been chosen to perform Sub-Interpolation inside a single DSP-chain; in fact, two propagation mechanisms and two thermometric codes are generated by a single chain, leading to a beneficial bin-splitting, as will be shown in Section 5.3. If the default 48-bit mode would have been chosen, another entire DSP-chain would have been placed in parallel to implement Sub-Interpolation, leading to a higher area occupancy. Since dual 24-bit SIMD mode is used, it is like having a single DSP slice containing two independent 24-bit ALUs working in parallel, where each one of them has 24-bit wide P, C, A:B signals, and 24-bit wide vectors of zeroes.

To create a long DSP-based TDL, more DSP slices have to be cascaded, and this is done by connecting the ACOUT and BCOUT ports of the previous slice to the ACIN and BCIN of the following one. Therefore, the START signal is connected to the direct A and B input ports of the first DSP block. Then, on the following ones, it propagates along the cascade passing through the ACIN and ACOUT inputs; here, the direct A and B inputs are not used anymore. Figure 36 shows the data assignment on the A:B vector. It is shown that, since A is 30-bit-wide and B is 18-bit-wide, the START signal must be assigned both to the 0-th bit of B and to the 6-th bit of A, thus having our signal of interest on the least-significant bit of each 24-bit half of the A:B vector (i.e.; $30 - 6 = 24$ for A and $18 + 6 = 24$ for B). In this way, in both the two 24-bit ALUs, the decimal value of the factor coming from the X multiplexer is equal to 0 when START = '0', and it increases to 1 as soon as START makes a '0' → '1' transition.

Finally, at the P output, the internal register of the DSP48E2 block is used to sample the TDL; therefore, external FFs are not required, as it was in the CARRY-based TDL. We can see in Figure 37 all the internal resources and the signals involved in the DSP setup.

By taking Equation 28 and substituting the values assigned to all the signals in the instantiation stage of the DSP48E2, we obtain the following equation:

$$P = 0 - (0 + START + 0 + 0) \quad (29)$$

which leads to the following subtraction:

$$P = 0 - START \quad (30)$$

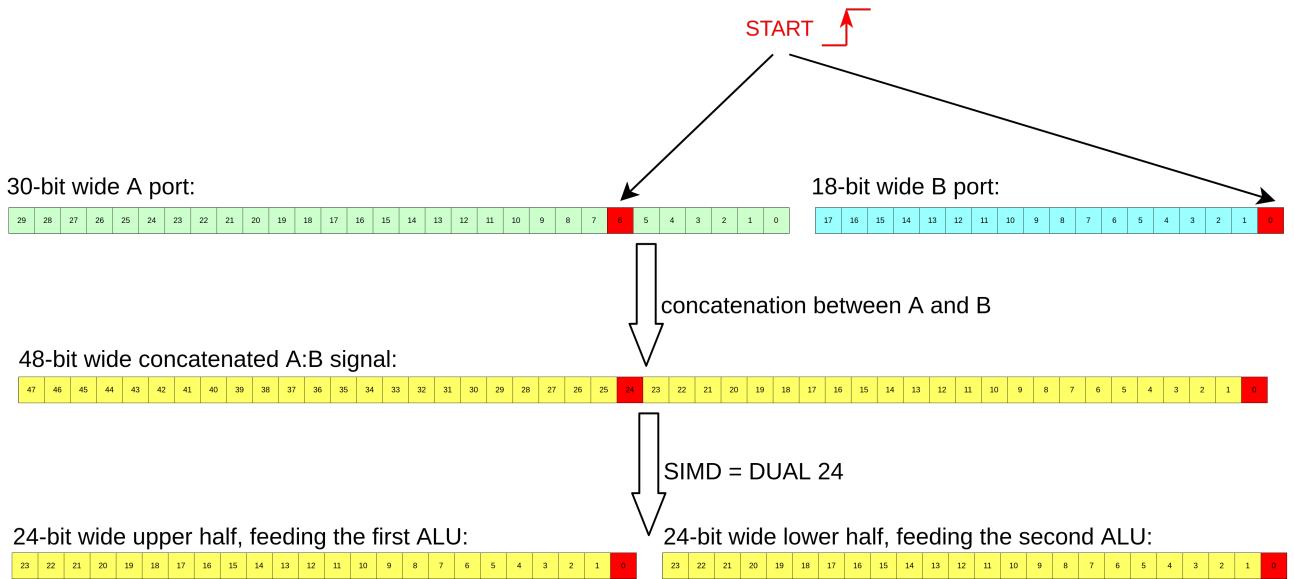


Figure 36: START signal connection to A and B ports of the DSP slice.

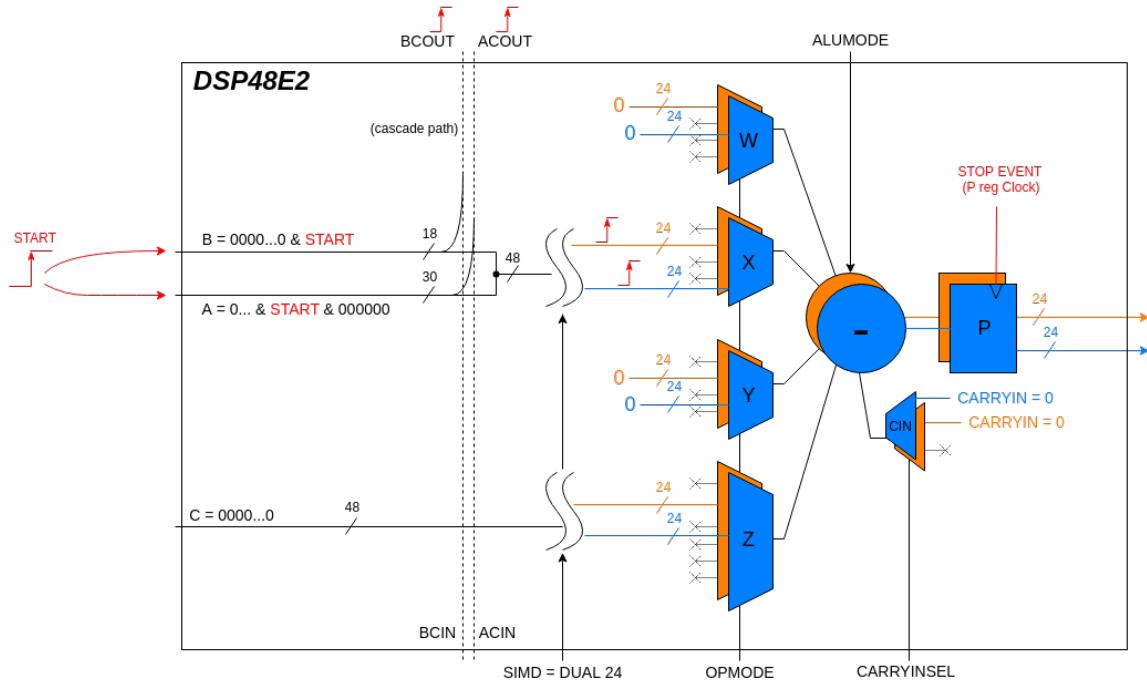


Figure 37: Internal operation of the DSP48E2 block.

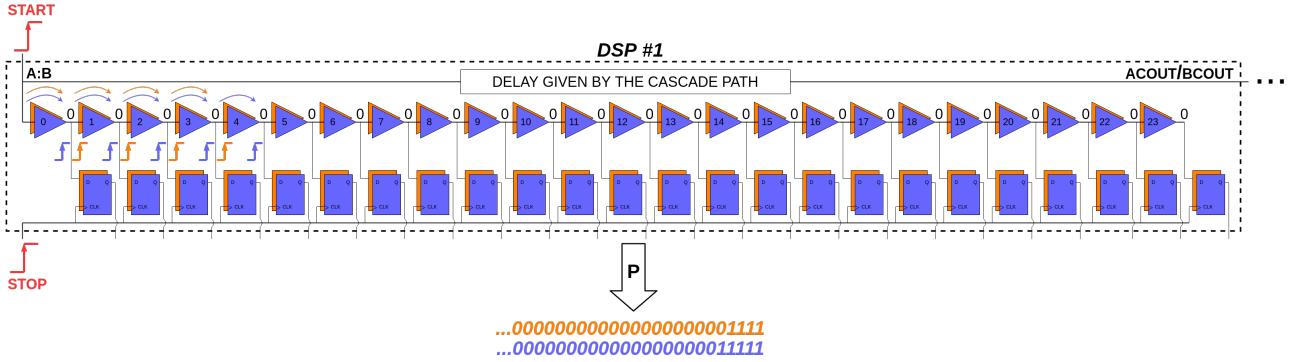


Figure 38: Propagation mechanism along the DSP-based TDL.

Therefore, until START stays at '0', P is a 24-bit wide vector of zeroes; then, as soon as START transits from '0' to '1', an underflow takes place, causing propagation of "ones" from the least-significant bit to the most-significant bit of P. This behavior is the same as in the CARRY-based TDL, where the bits of the P output are the taps of the TDL and the underflow-carry propagates with a certain propagation delay t_p on each one of them.

The DSP-based TDL relies on the same "VALID" mechanism of the CARRY-based counterpart, and the selected clock period for the TVALID assertion is the STOP signal taking a snapshot of the TDL's state. Since a single DSP48E2 is used in dual 24-bit mode, two thermometric codes are generated in parallel, instead of one. Figure 38 shows the propagation mechanism of the DSP-chain.

4.3. IP-Core Integration of both DSP-based and CARRY-based TDLs

The DSP-based solution for the V-TDL has been integrated into the IP-Core described in Paragraph 3.2.3.

In fact, as we will see in Chapter 5, when the experimental results will be shown, the combination of the CARRY-based TDL and the DSP-based TDL creates a "hybrid" V-TDL which, thanks to the SuperWU Sub-Interpolation algorithm, exploits the best features of both the two architectures. This "hybrid" implementation is managed by two new generics, *NUMBER_OF_CARRY_CHAINS* and *NUMBER_OF_DSP_CHAINS*, introduced in substitution of *NUMBER_OF_TDL*. The resulting V-TDL is composed of *NUMBER_OF_CARRY_CHAINS* + *NUMBER_OF_DSP_CHAINS* TDLs in parallel, where each one is *NUM_TAP_PRE_TDL* + *NUM_TAP_TDL* tap long. This adds even more versatility to the IP-Core since it allows having a V-TDL either made of CARRY-based TDLs only (if *NUMBER_OF_DSP_CHAINS*=0), DSP-based TDLs only (if *NUMBER_OF_CARRY_CHAINS*=0), or a combination of them (if *NUMBER_OF_CARRY_CHAINS*≠0, *NUMBER_OF_DSP_CHAINS*≠0).

It is worth saying that the generic *NUMBER_OF_DSP_CHAINS* decides how many "active" ALUs perform the subtraction operation in parallel. For example, in both cases where

NUMBER_OF_DSP_CHAINS = 1 or *NUMBER_OF_DSP_CHAINS* = 2, the hardware TDL is composed of a single DSP-chain; however, since every DSP48E2 is used in dual 24-bit SIMD mode, the difference is the following one: in the second case, both the 24-bit ALUs perform the operation, and two thermometric codes are generated in output (we will refer to it as "FULL-DSP-chain"); instead, in the first case, both ALUs still perform the operation, but one single thermometric code is provided, while the other is discarded (we will refer to it as "HALF-DSP-chain"). Since, in both cases, the same amount of hardware resources is exploited and since, thanks to the Sub-Interpolation, having two thermometric codes is more beneficial than having just a single code, choosing an even value for *NUMBER_OF_DSP_CHAINS* is always more convenient.

Moreover, the generic *XUS_VS_X7S* allows choosing the technological node of the DSP block, as for the CARRY primitive. If *XUS_VS_X7S* = "X7S", only DSP48E1 primitives are instantiated, and the TDC is made compatible with X7S FPGAs; on the other hand, if *XUS_VS_X7S* = "XUS", only DSP48E2 primitives are implemented, and the TDC is made compatible with XUS/XUS+ FPGAs. Figure 39 shows the V-TDL IP-Core.

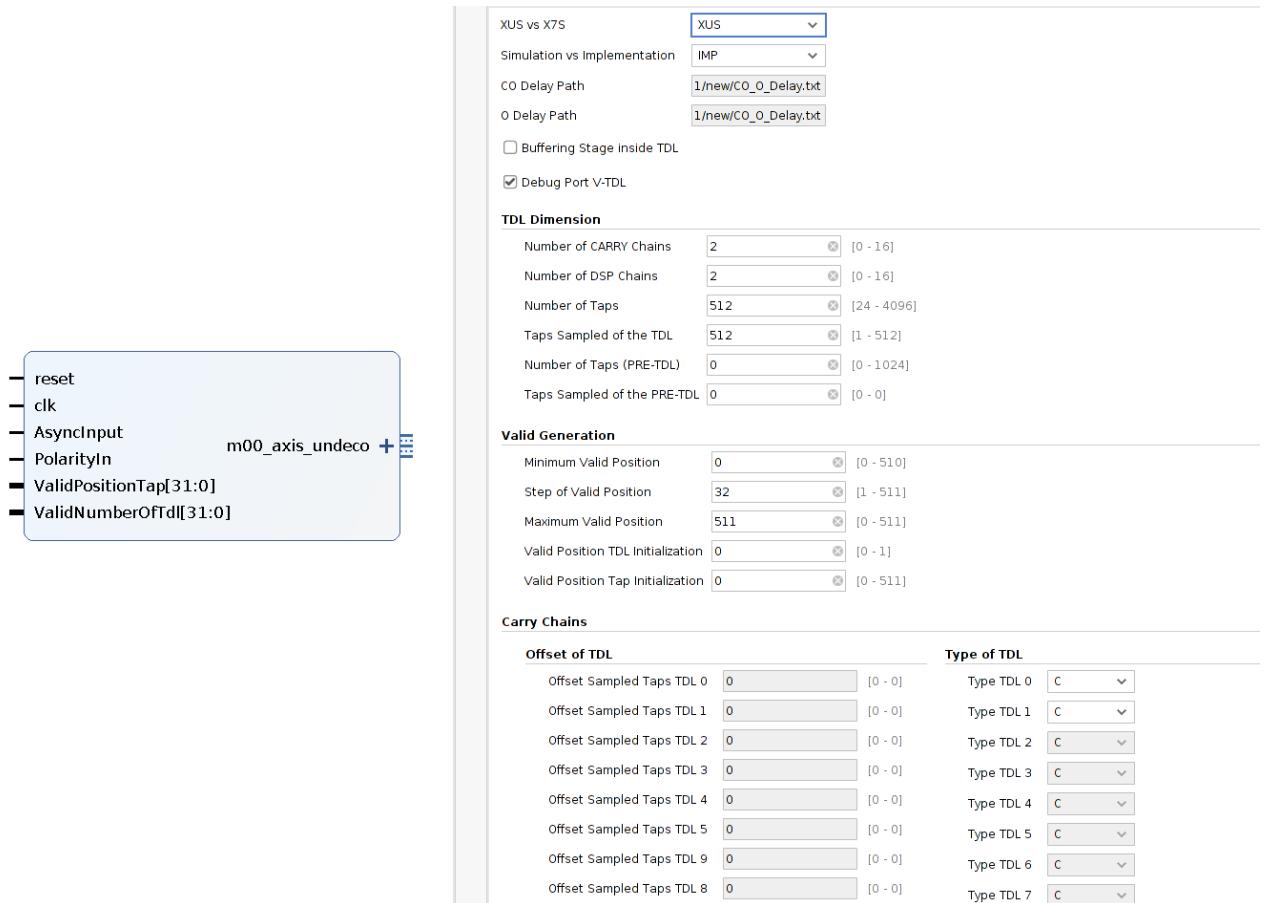


Figure 39: V-TDL IP-Core, exploiting both CARRY-chains and DSP-chains thanks to the new generics `NUMBER_OF_CARRY_CHAINS` and `NUMBER_OF_DSP_CHAINS` in substitution of `NUMBER_OF_TDL`. The generic `XUS_VS_X7S` is also still present.

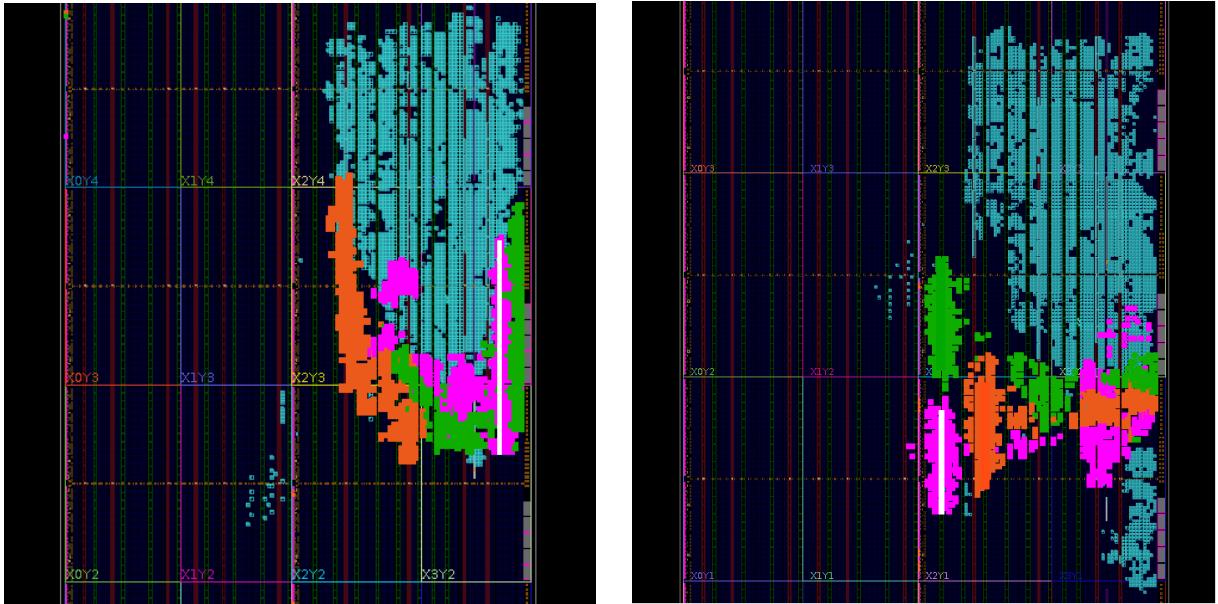


Figure 40: Length comparison between a CARRY-chain (on the left, in white) and a DSP-chain (on the right, in white) in a Kintex-Ultrascale FPGA.

4.4. Advantages and drawbacks of the DSP-based TDL

The main advantage of this architecture is that a more compact structure is obtained, since long CARRY chains, hindering the routing across different regions of the FPGA, are avoided. Figure 40 shows the concept just explained, showing the higher compactness of a DSP-chain (on the right) than a CARRY-chain (on the left) on a Kintex-Ultrascale FPGA fabric. It is worth noting that the DSP-chain routing will be facilitated in large designs since it has a lot of room before crossing all the FPGA fabric.

Furthermore, the DSP-based architecture allows a relaxation in the hardware utilization, since the sampling of the TDL is made with the P-register integrated with the DSP itself, thus not requiring the use of external FFs as in the CARRY-based approach. For example, if a 512-taps long CARRY-based TDL is implemented, it will need 512 as many FFs; on the other hand, the DSP-based counterpart will save those 512 FFs since it uses the ones integrated into the DSP blocks. However, the DSP-chains suffer from a relevant problem, since the cascade path from one DSP slice to another introduces a huge propagation delay and therefore the presence of a very high “ultra-bin” in the chain. The recurring presence of “ultra-bins” along the DSP-chain is the main drawback of this architecture since, as explained in Section 2.3, it worsens the single-shot precision of the TDC.

5. Measurements

The experimental results achieved by the XUS/XUS+ TDC implementation will now be shown. All the performances of the TDC have been tested on the KCU105 Evaluation Board by Xilinx [49], which is a development kit hosting a 20-nm Kintex-UltraScale™ FPGA (xcku040-ffva1156-2-e). Figure 41 shows the aforementioned evaluation board.

5.1. Measurement Setup

Figure 42 shows the block diagram of the complete system. As we can see, it contains way more blocks than the main core of the TDC (i.e., “TDC_Calib” in Figure). The most relevant are the Memory Management Engine (MME) [32][5], which manages the communication of the time measurements to the PC, and the Histogrammer [7] (i.e., “BeltBus_TDCHistogrammer” in Figure), which is in charge

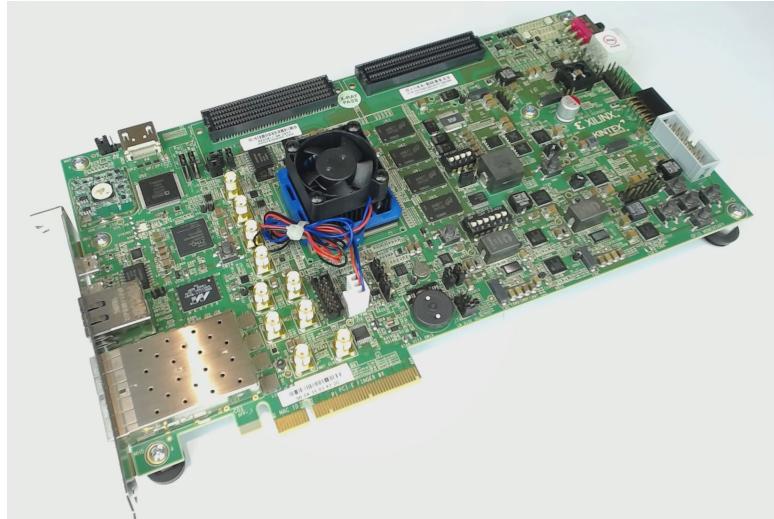


Figure 41: KCU105 Evaluation Board.

of calculating the time difference between the events on two different channels. Since these modules have not been developed in this thesis work, they will not be described in this presentation, but considered as black-boxes.

The employed measurement setup exploits the “Start-Stop Generator” module, in charge of producing the asynchronous input signals. It consists of an internal Ring Oscillator that produces a periodic square wave. The produced square wave is directly sent at the START output, while at the STOP output its delayed version is sent, with a delay equal to t_{Delay} . This module is the one in charge of sending the random input signals, at the power-on of the system, needed to build the CT of the TDC; however, once the Calibration process is completed, these signals can also be used as the “real” asynchronous input signals to be measured. In this work, a 3-channel TDC has been implemented, and the Start-Stop Generator is connected to the channels as shown in Figure 43. As we can see, the STOP signal feeds both Channel 1 and Channel 2, and, thanks to the routing on the FPGA, a delay t_{Skew} is present between them. The time distance between two rising-edges of the asynchronous input signals on Channel 1 and Channel 2 (i.e., t_{Skew}) is the measured quantity. Obviously, the same measurement process could be done either between Channel 0 and Channel 1 or between Channel 0 and Channel 2.

The experiments that have been carried out are the following ones:

- Verification of the benefits given by SuperWU Sub-Interpolation, in a V-TDL is composed of CARRY-chains only: the mean bins’ propagation delay ($\bar{t}_{p,V}$), the single-shot precision ($\sigma_{r.m.s.}$), and the hardware utilization have been measured, in the two cases where the V-TDL is composed of either a single TDL or 4 TDLs in parallel.
- Study of the sub-interpolating capability of the DSP-chains.
- Inspection of hardware utilization vs. single-shot precision, in the two cases where the 4 TDLs (i.e., the maximum level of beneficial Sub-Interpolation) are composed of either CARRY-chains only or a mix of DSP-chains and CARRY-chains.
- Linearity test of the TDC.

Let’s describe each experiment and show the achieved experimental results.

5.2. Verification of the Super Wave Union Sub-Interpolation on a CARRY-based V-TDL

The first experiment consisted of comparing two different implementations of the V-TDL, the first one without Sub-Interpolation (i.e., one single physical TDL is used to perform the Fine measurement) and the second one with Sub-Interpolation, using 4 TDLs in parallel. Both these two implementations have been made by using CARRY-chains only, and no DSP-chains. Firstly, a comparison between the Calibration Tables (CTs) in the two cases has been made. Figure 44 shows the obtained results.

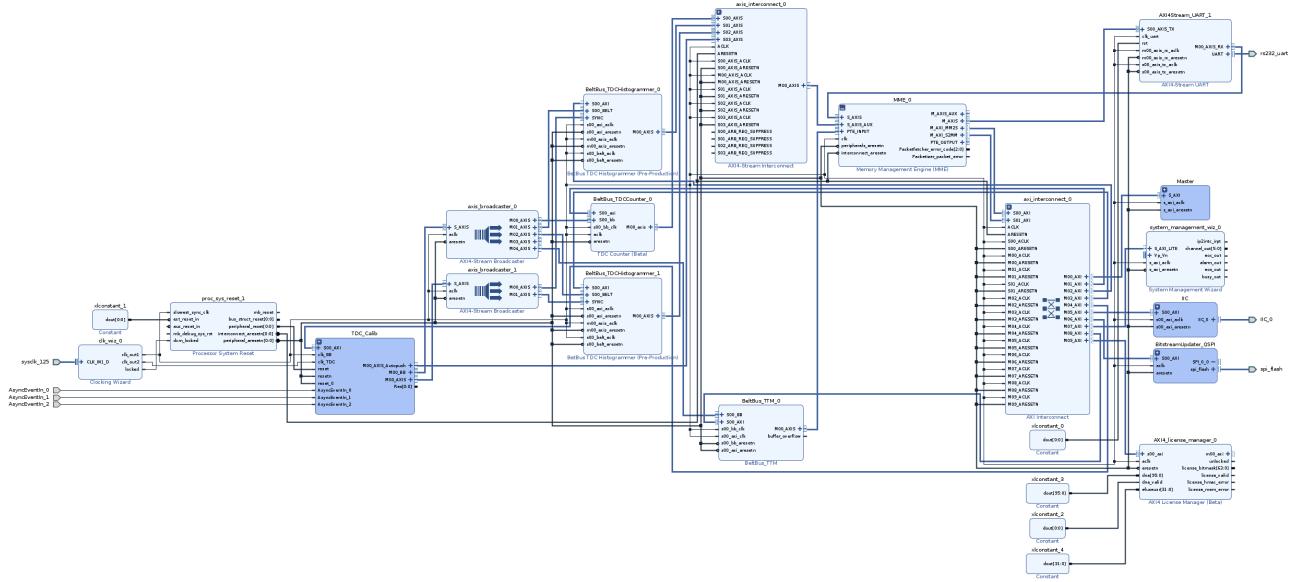


Figure 42: Block design of the entire system.

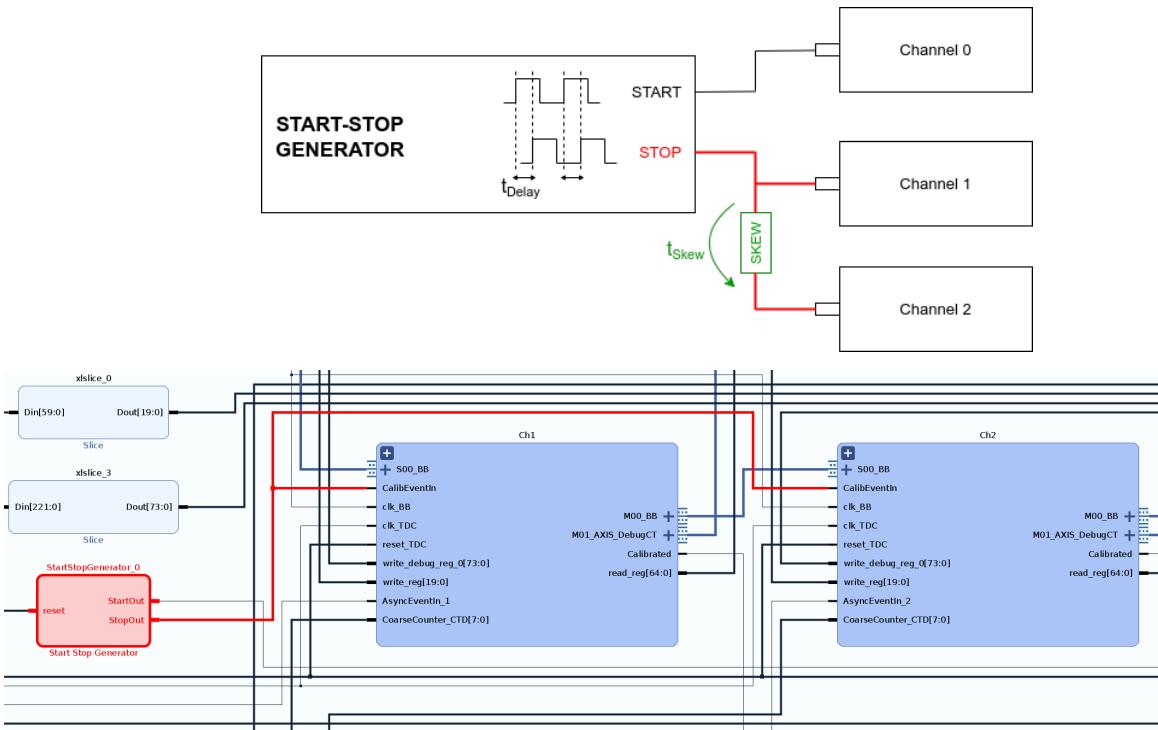


Figure 43: Start-Stop Generator connection to the TDC's channels.

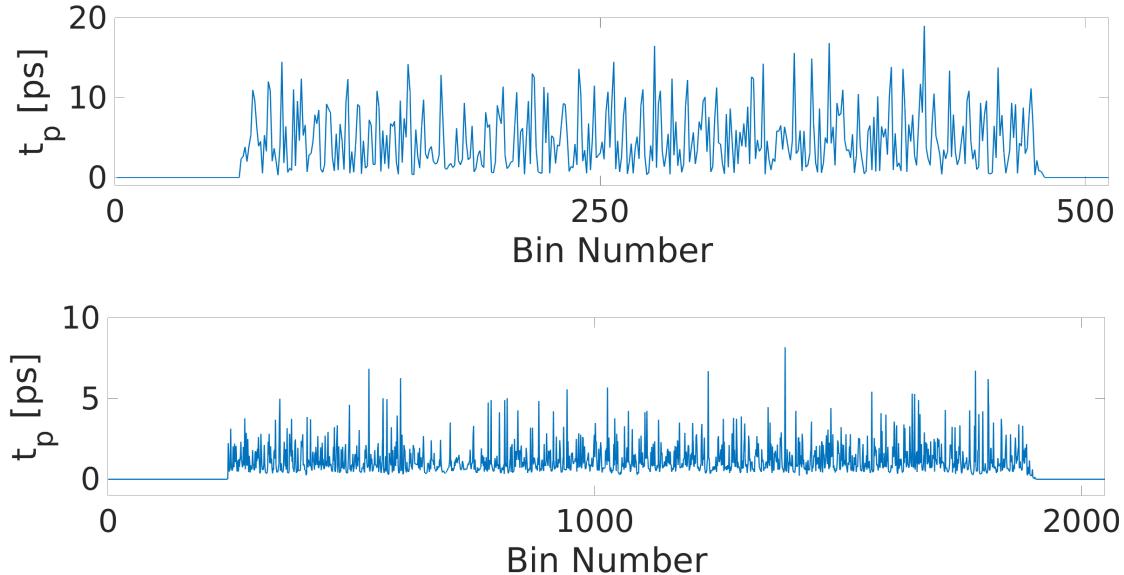


Figure 44: Comparison between the Calibration Tables in the case without Sub-Interpolation (at the top) and with Sub-Interpolation, using 4 TDLs in parallel (at the bottom).

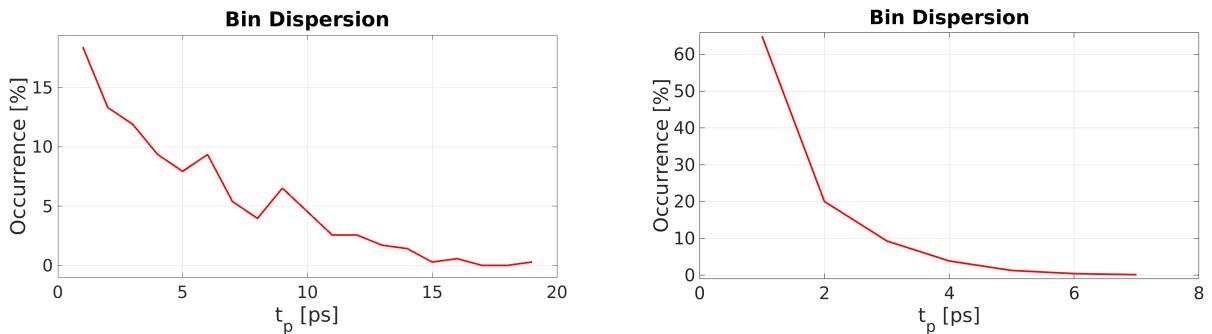


Figure 45: Comparison between the dispersion of the bins' propagation delays in the case without Sub-Interpolation (on the left) and with Sub-Interpolation (on the right).

We can also see in Figure 45 the comparison between the dispersion of the bin's propagation delays in the case without Sub-Interpolation (on the left) and with Sub-Interpolation (on the right), showing a better result in the latter case.

The implementation using Sub-Interpolation produces a four-times wider CT than the one without Sub-Interpolation. This is because the SuperWU algorithm produces N_V virtual taps equal to $M \cdot N_B$, where N_B is the number of taps of the single, “real”, TDL (i.e., 512 in our case), and M is the number of TDLs placed in parallel (i.e., 4 in our case). Therefore, 2048 “virtual” taps have been obtained in the sub-interpolated case. The most relevant information provided by the CT is the one describing the propagation delays of the TDL’s taps. In fact, the “virtual” taps of the sub-interpolated TDL are overall faster than their “real” counterpart and, mostly, the 19-ps ultra-bin is lowered a lot in the process. This leads to a better single-shot precision of the time measurement between Channel 1 and Channel 2. The information about the time measurement is contained in a histogram produced by the BeltBus_TDCHistogrammer module mentioned before, which is in charge of performing a set of repeated measurements of the time intervals elapsing between the events on the two channels; the x-axis of the histogram represents the time difference, in ps, while the y-axis represents the number of occurrences of that specific time measurement. Therefore, as explained in Paragraph 1.3.3, if a Gaussian curve is fitted on this histogram, we will consider as a highly-precise time measurement the one providing an as narrow as possible Gaussian curve, meaning that the uncertainty of the measurement is low. The standard deviation $\sigma_{r.m.s.}$ of the fitted Gaussian curve will be used as the FoM of the single-shot precision of the TDC. Since the time measurement is carried out between two

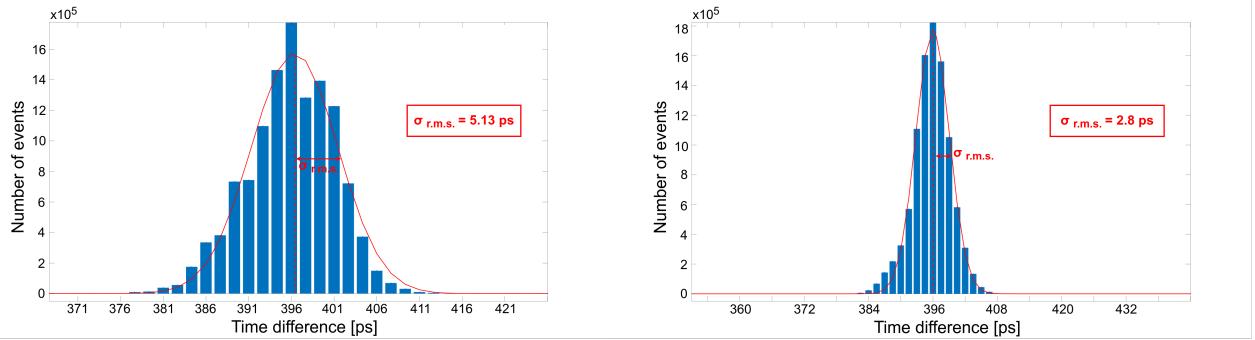


Figure 46: Comparison between the single-shot precision extracted from the Histograms representing the time measurement, in the case without Sub-Interpolation (on the left, i.e., $\sigma_{r.m.s.} = 5.13 \text{ ps r.m.s.}$, $\sigma_{CH} = 3.6 \text{ ps r.m.s.}$) and with Sub-Interpolation (on the right, i.e., $\sigma_{r.m.s.} = 2.8 \text{ ps r.m.s.}$, $\sigma_{CH} = 1.98 \text{ ps r.m.s.}$).

channels, the following relation holds:

$$\sigma_{r.m.s.} = \sqrt{\sigma_{CH1}^2 + \sigma_{CH2}^2} \quad (31)$$

where σ_{CH} is the standard deviation belonging to a single channel (i.e., CH1 or CH2).

Since the two channels are very similar, we can assume that $\sigma_{CH1} \simeq \sigma_{CH2}$; therefore, the single-shot precision σ_{CH} of a single TDC channel is equal to:

$$\sigma_{CH} \simeq \frac{\sigma_{r.m.s.}}{\sqrt{2}} \quad (32)$$

Figure 46 shows the histograms representing the time measurement between Channel 1 and 2, both in the case without Sub-Interpolation (on the left) and with 4-th order Sub-Interpolation (on the right). As expected, thanks to the bin-splitting phenomenon given by the SuperWU algorithm, the latter case results in a lower standard deviation and, therefore, a better single-shot precision.

It is possible to retrieve from the CT in Figure 46 (at the top) the mean propagation delay of the real bins ($\overline{t_{p,B}}$) of the TDL, resulting in 4.8 ps. The mean propagation delay of the virtual bins ($\overline{t_{p,V}}$) of the V-TDL is instead improved by a factor of $M = 4$ thanks to the Sub-Interpolation, and it is equal to:

$$\overline{t_{p,V}} = \frac{\overline{t_{p,B}}}{M} = \frac{4.8 \text{ ps}}{4} = 1.2 \text{ ps} \quad (33)$$

The “ultra-bin” also improves more or less by a factor of $M = 4$ (i.e., $t_{p,B}^{MAX}/M = 19 \text{ ps}/4 = 4.75 \text{ ps}$ very similar to $t_{p,V}^{MAX} = 7 \text{ ps}$).

In the X7S TDC version, each TDL was composed of a number N_B of taps equal to 256 rather than 512, due to the slower bins belonging to the less scaled technology; furthermore, a Sub-Interpolation of order $M = 4$ was implemented. Therefore, the XUS TDC implemented in this work is characterized by a doubled hardware occupancy than the X7S version. Tables 9 and 10 summarize all the achieved performances and the used hardware resources in the XUS CARRY8-based TDC, tested in this experiment, and the X7S CARRY4-based TDC, respectively. $M = 4$ has been used as the maximum Sub-Interpolation order in both the XUS TDC and the X7S TDC; in fact, with higher values of M , the area/power consumption was growing excessively, and the SuperWU was becoming ineffective [22].

5.3. Preliminary studies about the introduction of DSP-chains in the Super Wave Union Sub-Interpolation

In order to validate the advantages and drawbacks described in Section 4.4, the DSP-based TDL architecture has been studied, using both a HALF-DSP-chain and a FULL-DSP-chain (see Section 4.3

Feature	without Sub-Interpolation	with Sub-Interpolation
Clock TDC Freq.	500 MHz	
$N_B/M/N_V$	512/1/512	512/4/2048
TDL mean propagation delay (t_p)	4.8 ps	1.2 ps
TDL ultra-bin	19 ps	7 ps
Full-Scale Range	some days	
Single-Shot Precision ($\sigma_{r.m.s.}$)	5.13 ps r.m.s.	2.8 ps r.m.s.
Max. Channel Rate	200 MHz	
Dead-Time	2 ns	
LUT/FF/BRAM (for one ch.)	2209/3050/1.5	5431/7716/3.5
Number of Chs (*)	up to 150	up to 62
Power Consumption (for one ch.)	0.343 W	0.596 W

Table 9: Achievable performances of the XUS CARRY8-based TDC, with and without Sub-Interpolation. (*) in the xcku040-ffva1156-2-e.

Feature	without Sub-Interpolation	with Sub-Interpolation
Clock TDC Freq.	416 MHz	
$N_B/M/N_V$	256/1/256	256/4/1024
TDL mean propagation delay (t_p)	12 ps	3 ps
TDL ultra-bin	70 ps	20 ps
Full-Scale Range	some days	
Single-Shot Precision ($\sigma_{r.m.s.}$)	15 ps r.m.s.	12 ps r.m.s.
Max. Channel Rate	100 MHz	
Dead-Time	5 ns	
LUT/FF/BRAM (for one ch.)	1185/1840/1.5	3399/4615/2
Power Consumption (for one ch.)	0.27 W	0.359 W

Table 10: Reference performances of the X7S CARRY4-based TDC, with and without Sub-Interpolation.

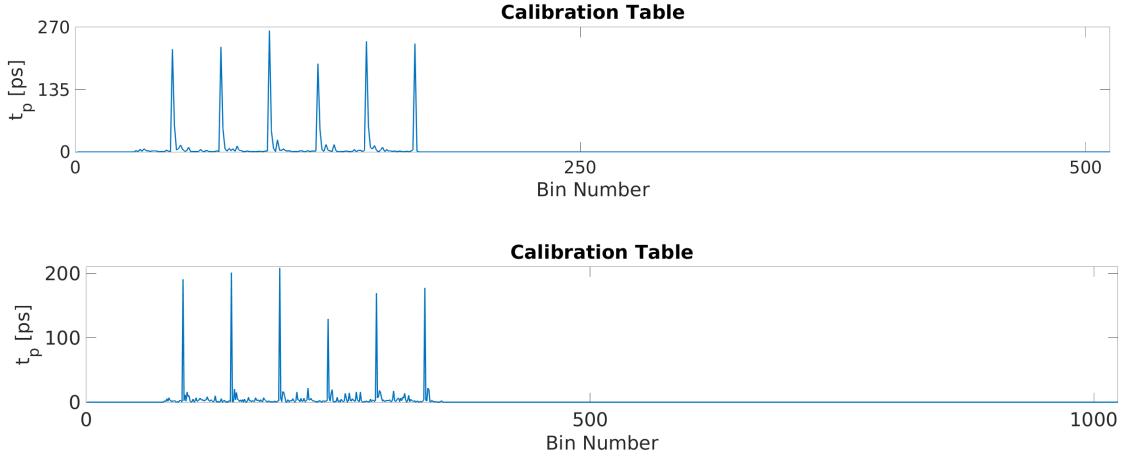


Figure 47: Calibration Table of a HALF-DSP-chain (at the top) and of a FULL-DSP-chain (at the bottom).

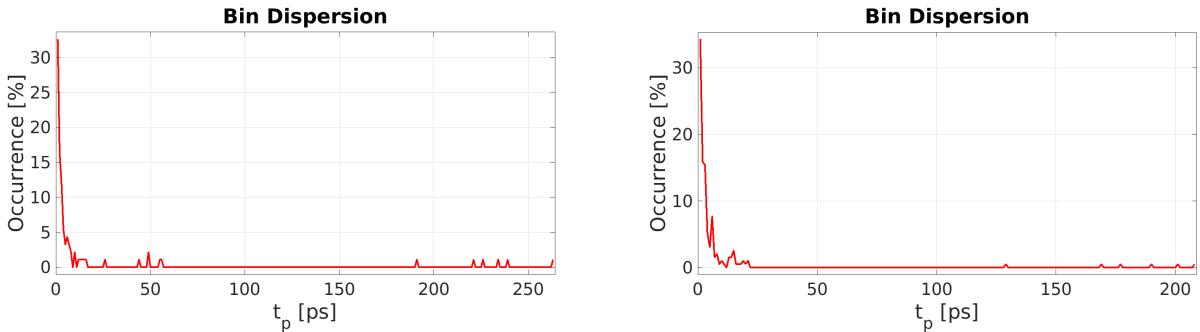


Figure 48: Comparison between the dispersion of the bins' propagation delays in the HALF-DSP-chain case (on the left) and in the FULL-DSP-chain case (on the right).

for the adopted nomenclature). Figure 47 shows that, for a HALF-DSP-chain, the “ultra-bins” have a propagation delay around 260 ps, while in the FULL-DSP-chain they are around 200 ps, thanks to the Sub-Interpolation between the two halves.

We can also see in Figure 48 the comparison between the dispersion of the bins’ propagation delays in the HALF-DSP-chain case (on the left) and in the FULL-DSP-chain case (on the right), showing a slightly better result in the latter case.

As we can see from the histograms in Figure 49, both the two cases depicted in Figure 47 lead to a very bad single-shot precision of the time measurement. This problem can be solved only by placing CARRY-chains in parallel to the DSP-chains (see Paragraph 5.4.2).

The mean propagation delay of the virtual bins ($\bar{t}_{p,V}$) of the V-TDL is improved by a factor of $M = 2$ thanks to the Sub-Interpolation, in the FULL-DSP-chain (i.e., $\bar{t}_{p,B}/M = 14\text{ ps}/2 = 7\text{ ps}$ like the $\bar{t}_{p,V} = 7\text{ ps}$); however, this is not true for the “ultra-bins” (i.e., $t_{p,B}^{MAX}/M = 263\text{ ps}/2 = 131.5\text{ ps}$ that is different with respect to $t_{p,V}^{MAX} = 203\text{ ps}$) as the Sub-Interpolation is not very effective on them (see Paragraph 5.4.3 for the detailed explanation). Since both these cases exploit the same hardware resources (i.e., the same DSP48E2 blocks), and, since there is a benefit on the “ultra-bin” in the second case, there are no reasons not to always use the FULL-DSP-chain in practical contexts. Table 11 summarizes the achieved performances and the used hardware resources in the HALF-DSP-based TDC and in the FULL-DSP-based TDC.

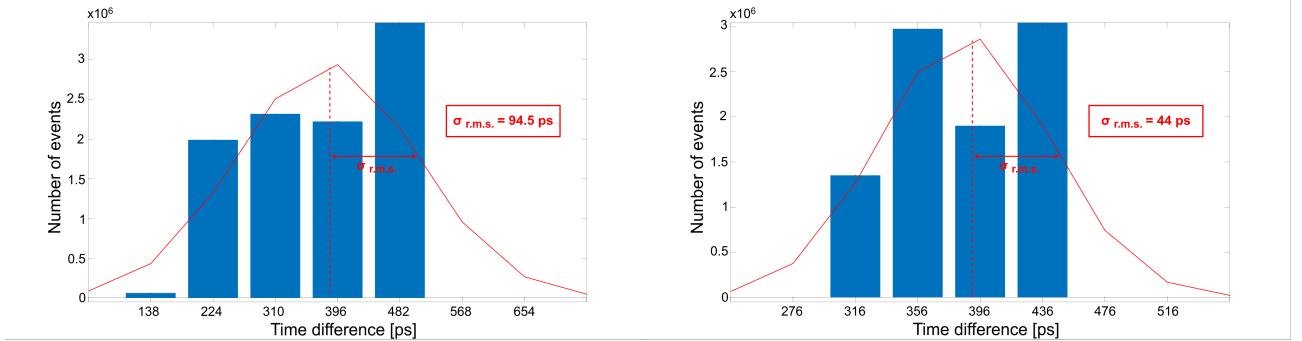


Figure 49: Comparison between the single-shot precision extracted from the Histograms representing the time measurement, in the case of a HALF-DSP-chain (on the left, i.e., $\sigma_{r.m.s.} = 94.5 \text{ ps}$, $\sigma_{CH} = 66.8 \text{ ps}$) and of a FULL-DSP-chain (on the right, i.e., $\sigma_{r.m.s.} = 44 \text{ ps}$, $\sigma_{CH} = 31.1 \text{ ps}$).

Feature	HALF-DSP-based TDC	FULL-DSP-based TDC
Clock TDC Freq.		500 MHz
$N_B/M/N_V$	512/1/512	512/2/1024
TDL mean propagation delay (t_p)	14 ps	7 ps
TDL ultra-bin	263 ps	203 ps
Full-Scale Range		some days
Single-Shot Precision ($\sigma_{r.m.s.}$)	94.5 ps r.m.s.	44 ps r.m.s.
Max. Channel Rate		200 MHz
Dead-Time		2 ns
LUT/FF/BRAM (for one ch.)		2216/2565/1.5
Power Consumption (for one ch.)		0.343 W

Table 11: Achievable performances of the HALF-DSP-based TDC and of the FULL-DSP-based TDC.

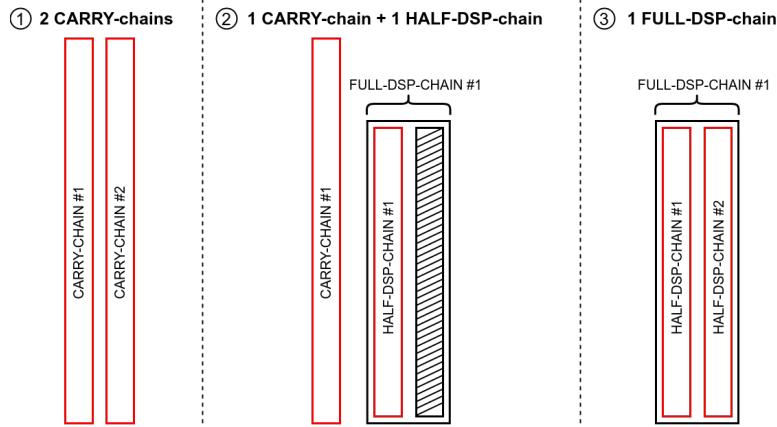


Figure 50: SuperWU combinations tested in the experiment.

5.4. Preliminary studies about the introduction of the DSP-chains in parallel to the CARRY-chains

This experiment consisted of acquiring the CT of the V-TDL and studying how much the Sub-Interpolation algorithm can be beneficial with the introduction of DSP-chains alongside the CARRY-chains. In this preliminary study about the SuperWU effectiveness, a Sub-Interpolation of order $N = 2$ has been implemented. Figure 50 shows the three different combinations of TDLs tested in this experiment: 2 CARRY-chains in parallel; 1 CARRY-chain + 1 HALF-DSP-chain in parallel; 1 FULL-DSP-chain (i.e., 2 HALF-DSP-chains in parallel).

Based on the SuperWU operation [22], to have an effective Sub-Interpolation (i.e., it is capable of splitting the "ultra-bins" and providing "virtual bins" with as homogeneous as possible propagation delays) we would like the two involved TDLs to be as different as possible, concerning their taps' propagation delays. The degree of dissimilarity between the two TDLs has been calculated by performing the cross-correlation function between the CTs of each real TDL. The cross-correlation function is a measure of similarity between two signals. By calling CT_1 the CT (composed of N_{CT1} bins) acquired from the first TDL, and CT_2 (composed of N_{CT2} bins) the one acquired from the second TDL, we call CT_1^* and CT_2^* their mean-free version, equal to:

$$CT_1^*[n] = CT_1[n] - \text{mean}(CT_1), n = 1, 2, \dots, N_{CT1} \quad (34)$$

$$CT_2^*[n] = CT_2[n] - \text{mean}(CT_2), n = 1, 2, \dots, N_{CT2} \quad (35)$$

The mean-free results of Equations 34 and 35 are used in the cross-correlation function to avoid the ramp-like behavior of the resulting plot. Then, the cross-correlation function has been calculated as follows:

$$R_{CT_1^*CT_2^*}[k] = \sum_{m=-\infty}^{\infty} CT_1^*[m] \cdot CT_2^*[m+k] \quad (36)$$

where k is the lag between the two signals, representing the shift of the second CT with respect to the first one, in number of bins. Finally, the cross-correlation has been normalized in the [-1;1] range, using this relation:

$$\rho_{CT_1^*CT_2^*}[k] = \frac{R_{CT_1^*CT_2^*}}{\sqrt{\sum_{m=-\infty}^{\infty} (CT_1^*[m])^2 \cdot \sum_{m=-\infty}^{\infty} (CT_2^*[m])^2}} \quad (37)$$

Let's now show, for each case reported in Figure 50, the normalized cross-correlation function and the resulting CTs after Sub-Interpolation.

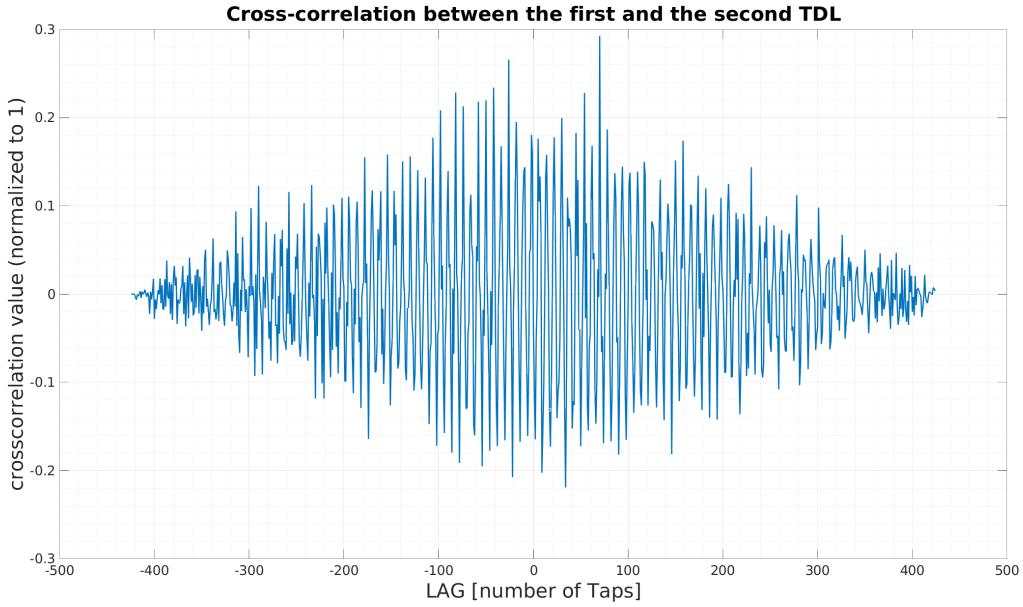


Figure 51: Normalized cross-correlation function for the two CARRY-chains.

Implementation	$\overline{t_{p,V}}$	ultra-bin	$\sigma_{r.m.s.}$
4 CARRY-chains	1.2 ps	7 ps	2.8 ps r.ms.
2 CARRY-chains	2.4 ps	11.6 ps	
1 CARRY-chain	4.8 ps	19 ps	5.13 ps r.ms.

Table 12: Achievable mean bins' propagation delay and "ultra-bin" value of a 4 CARRY-chains-based V-TDL (see Paragraph 5.2), a 2 CARRY-chains-based V-TDL and a single CARRY-chain-based TDL.

5.4.1. 2 CARRY-chains

The normalized cross-correlation function for two CARRY-chains placed in parallel is shown in Figure 51.

The maximum value, equal to 0.3, proves that the CARRY-chains are quite dissimilar, thus giving an effective Sub-Interpolation. In Figure 52 (at the top) we can see that the CT of the resulting V-TDL is quite flat, meaning that the propagation delays along the bins are homogeneous and the SuperWU algorithm was effective. Figure 52 also shows the dispersion of the bins' propagation delays (at the bottom).

Confronting Figure 52 (at the top) with Figure 44 (at the top), and Figure 52 (at the bottom) with Figure 45 (on the left), we can see the better performances of the TDC based on a V-TDL with 2 CARRY-chains with respect to the one based on a single CARRY-chain. Table 12 summarizes the mean bins' propagation delays and the "ultra-bin" values of the aforementioned implementations.

5.4.2. 1 CARRY-chain + 1 HALF-DSP-chain

The resulting normalized cross-correlation function for a CARRY-chain placed in parallel to a HALF-DSP-chain is reported in Figure 53.

The maximum value is equal to 0.2, which is comparable to the previous case. Therefore, the Sub-Interpolation is still very effective; however, since the "ultra-bins" of the HALF-DSP-chain are huge, in the order of 260 ps, the resulting CT will not be as flat as in the previous case, as we can see in Figure 54 (at the top). Despite this, the Sub-Interpolation's effectiveness can be seen by the fact that the "ultra-bin" is now just 17 ps, rather than the original 260 ps. Figure 54 also shows the dispersion of the bins' propagation delays (at the bottom).

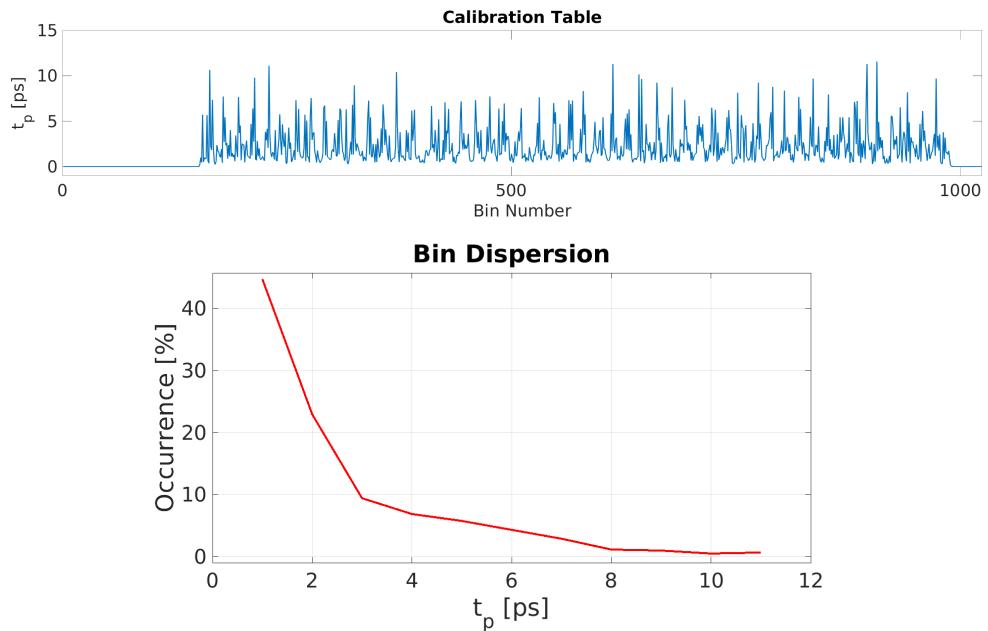


Figure 52: Calibration Table of a V-TDL composed of two CARRY-chains (at the top) and its dispersion of the bins' propagation delays (at the bottom).

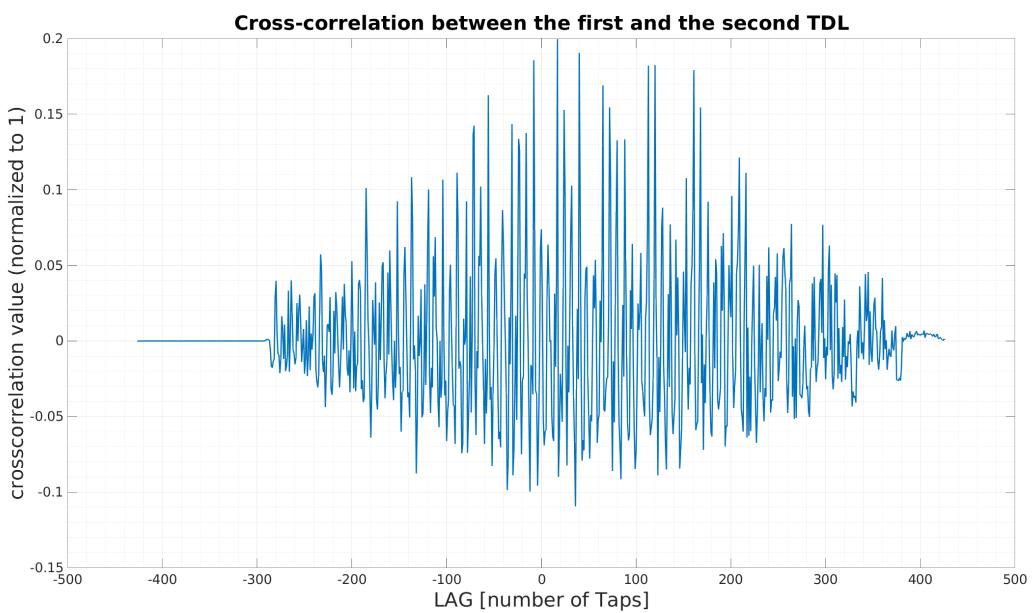


Figure 53: Normalized cross-correlation function for the CARRY-chain in parallel to the HALF-DSP-chain.

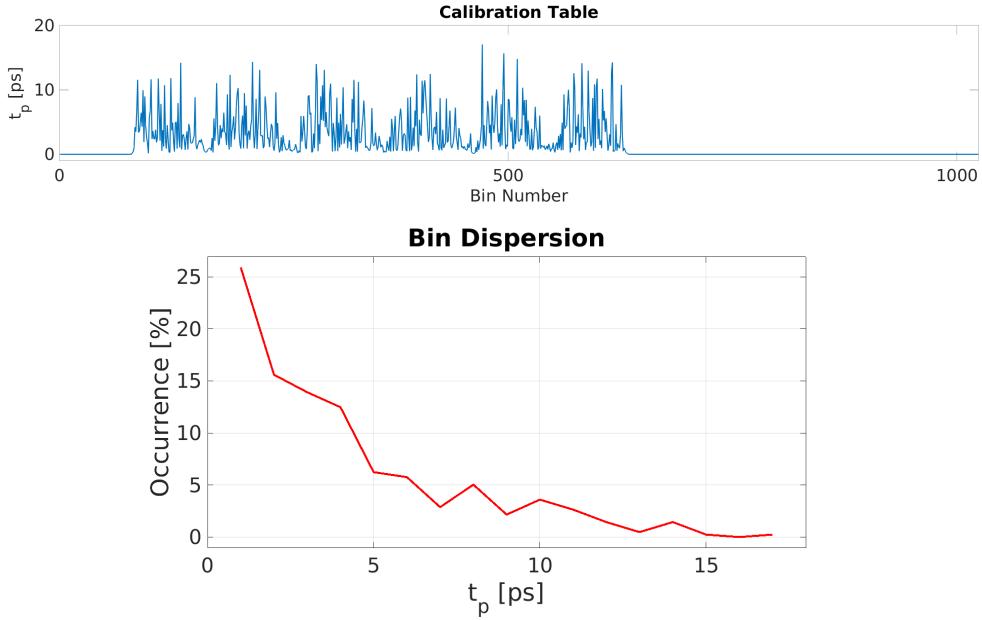


Figure 54: Calibration Table of a V-TDL composed of a CARRY-chain in parallel to a HALF-DSP-chain (at the top) and its dispersion of the bins' propagation delays (at the bottom).

Implementation	$\bar{t}_{p,V}$	ultra-bin	$\sigma_{r.m.s.}$
1 CARRY-chain + 1 HALF-DSP-chain	3.6 ps	17 ps	
1 CARRY-chain	4.8 ps	19 ps	5.13 ps r.m.s.
1 HALF-DSP-chain	14 ps	263 ps	94.5 ps r.m.s.

Table 13: Achievable mean bins' propagation delay and “ultra-bin” value of a 1 CARRY-chain + 1 HALF-DSP-chain based V-TDL, a 1 CARRY-chain based TDL (see Paragraph 5.2), and a 1 HALF-DSP-chain based TDL (see Paragraph 5.3).

We can thus compare Figure 54 (at the top, i.e., 1 CARRY-chain + 1 HALF-DSP-chain based V-TDL) with Figures 44 (at the top, i.e., 1 CARRY-chain based TDL) and 47 (at the top, i.e., 1 HALF-DSP-chain based TDL), and Figure 54 (at the bottom) with Figures 45 (on the left) and 48 (on the left). Table 13 summarizes the mean bins' propagation delays and the “ultra-bin” values of the aforementioned implementations.

5.4.3. 1 FULL-DSP-chain

Figure 55 shows the normalized cross-correlation for two HALF-DSP-chains in parallel.

We can see that the maximum value is equal to 1, and other high peaks are also present. This means that two HALF-DSP-chains are usually very similar, and the SuperWU algorithm is not effective in this case; therefore, the “ultra-bin” does not improve by a factor of $M = 2$, but just by a lower quantity (i.e., 60 ps). Figure 56 shows the CT of the resulting V-TDL (at the top) and the dispersion of the bins' propagation delays (at the bottom).

We can thus compare Figure 56 (at the top, i.e., 1 FULL-DSP-chain based V-TDL) with Figure 47 (at the top, i.e., 1 HALF-DSP-chain based TDL), and Figure 56 (at the bottom) with Figure 48 (on the left). Table 14 summarizes the mean bins' propagation delays and the “ultra-bin” values of the aforementioned implementations.

Despite the Sub-Interpolation not being very effective, achieving a 60 ps-faster “ultra-bin”, using the exact same hardware resources of a HALF-DSP-chain, leads to the obvious choice of always using the FULL-DSP-chain.

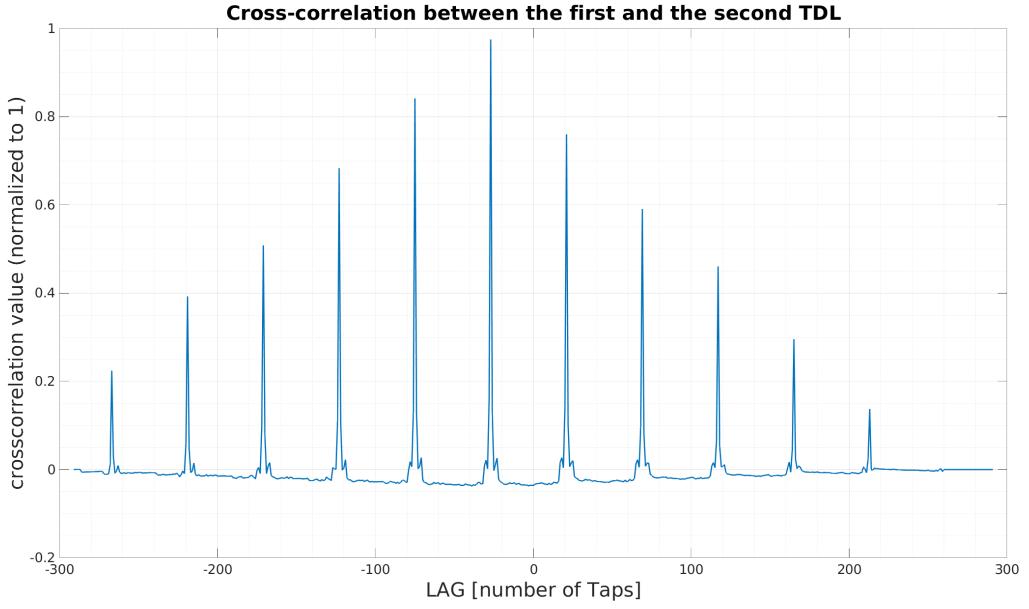


Figure 55: Normalized cross-correlation function for the two half-chains composing the FULL-DSP-chain.

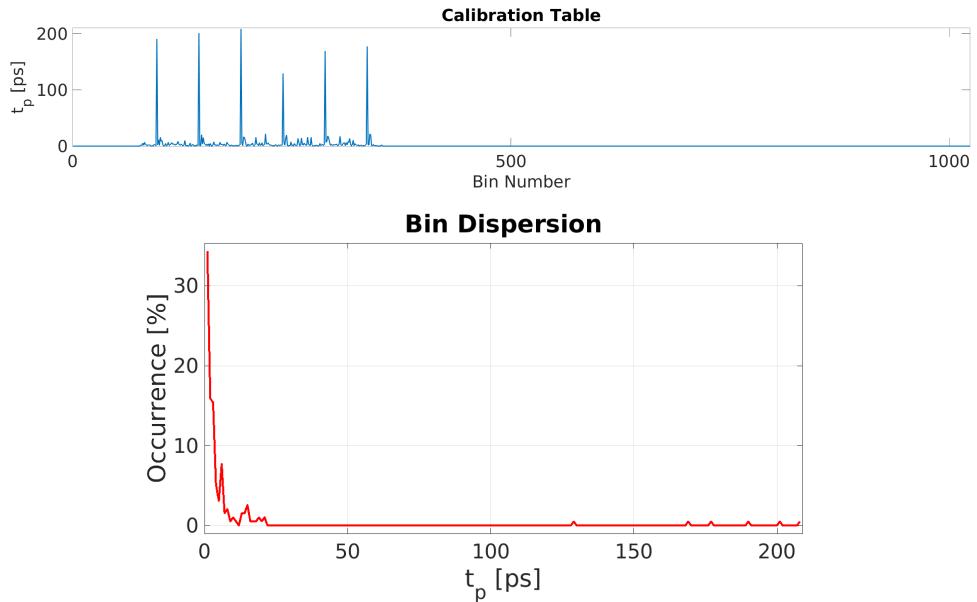


Figure 56: Calibration Table of a V-TDL composed of two HALF-DSP-chain (at the top) and its dispersion of the bins' propagation delays (at the bottom).

Implementation	$\bar{t}_{p,V}$	ultra-bin	$\sigma_{r.m.s.}$
1 FULL-DSP-chain	7 ps	203 ps	44 ps r.m.s
1 HALF-DSP-chain	14 ps	263 ps	94.5 ps r.m.s

Table 14: Achievable mean bins' propagation delay and “ultra-bin” value of a 1 FULL-DSP-chain based V-TDL and a 1 HALF-DSP-chain based TDL (see Paragraph 5.3).

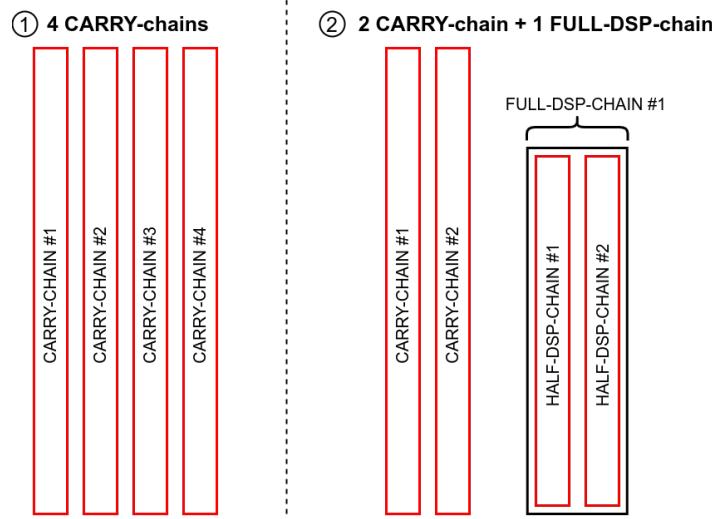


Figure 57: Implementations studied in this experiment: 4 CARRY-chains in parallel (first implementation) and 2 CARRY-chains + 1 FULL-DSP-chain in parallel (second implementation).

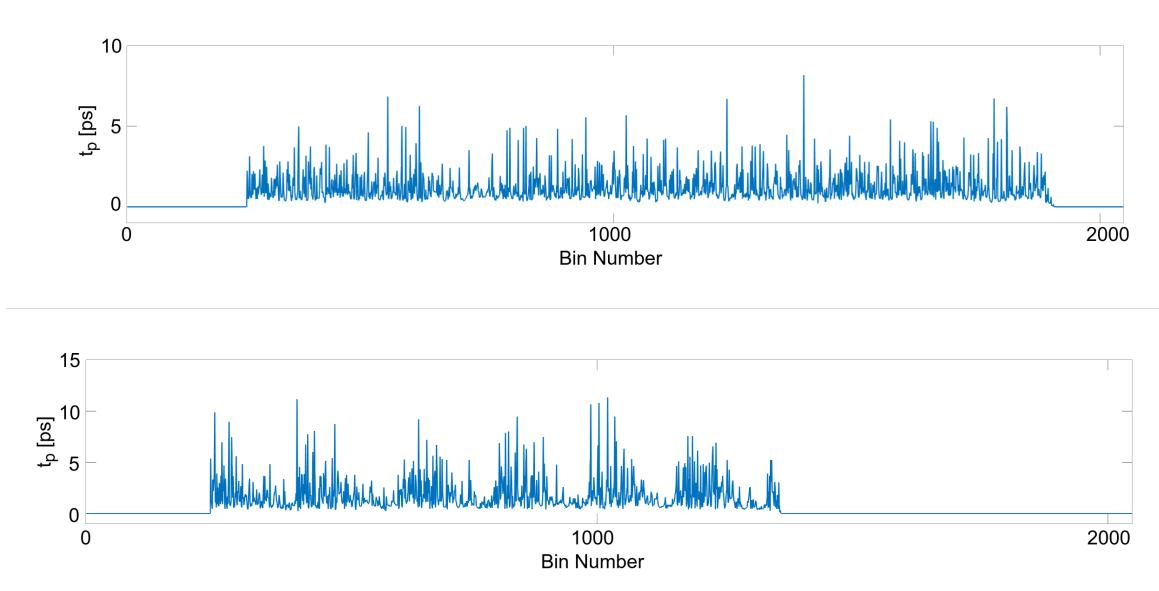


Figure 58: Calibration Table of a V-TDL composed of 4 CARRY-chains (at the top) and of 2 CARRY-chains + 1 FULL-DSP-chain (at the bottom).

5.5. Study of the trade-off between single-shot precision and hardware utilization in a fully-CARRY-based V-TDL vs. a mixed-CARRY-DSP V-TDL

As a fourth experiment, the two architectures reported in Figure 57, characterized by the maximum Sub-Interpolation order (i.e., $M = 4$ in this thesis work), have been compared, highlighting the trade-off existing between single-shot precision and hardware utilization.

Figure 58 shows the comparison of the resulting CTs for the two implementations.

We can also see in Figure 59 the comparison between the dispersion of the bin's propagation delays in the first implementation (i.e., 4 CARRY-chains, on the left) and in the second implementation (i.e., 2 CARRY-chains + 1 FULL-DSP-chain, on the right), showing a better result in the first one.

We can thus summarize in Table 15 the mean bins' propagation delays and the "ultra-bin" values of all the implementations described in this Chapter.

Figure 60 shows the comparison of the acquired histograms for the two implementations tested in this experiment (i.e., 4 CARRY-chains and 2 CARRY-chains + 1 FULL-DSP-chain).

We can see from Figures 58 and 59 that the first implementation has a slightly better $\overline{t_{p,V}}$ and "ultra-

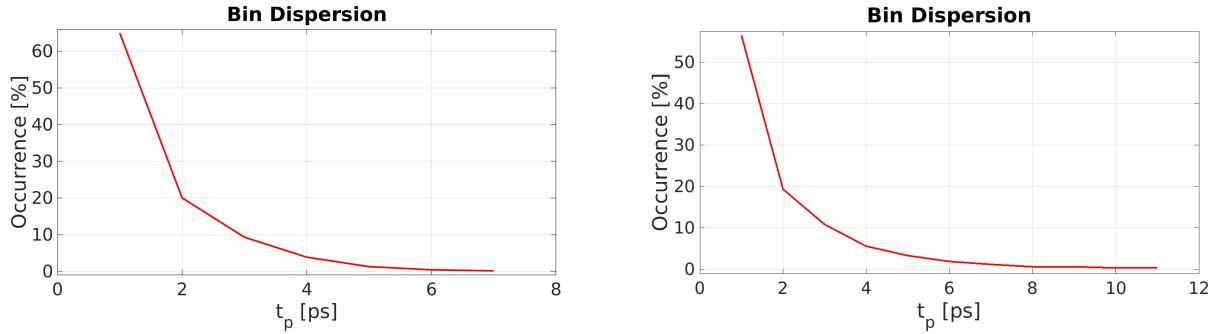


Figure 59: Comparison between the dispersion of the bin's propagation delays in the first implementation (i.e., 4 CARRY-chains, on the left) and in the second implementation (i.e., 2 CARRY-chains + 1 FULL-DSP-chain, on the right).

Implementation	$\bar{t}_{p,V}$	ultra-bin	$\sigma_{r.m.s.}$
4 CARRY-chains	1.2 ps	7 ps	2.8 ps r.m.s.
2 CARRY-chains + 1 FULL-DSP-chain	1.7 ps	11.3 ps	
2 CARRY-chains	2.4 ps	11.6 ps	
1 FULL-DSP-chain	7 ps	203 ps	44 ps r.m.s.
1 CARRY-chain	4.8 ps	19 ps	5.13 ps r.m.s.
1 HALF-DSP-chain	14 ps	263 ps	44 ps r.m.s.

Table 15: Achievable mean bins' propagation delay and “ultra-bin” value of a 4 CARRY-chain based V-TDL (see Paragraph 5.2), a 2 CARRY-chains + 1 FULL-DSP-chain based V-TDL, a 2 CARRY-chains based V-TDL (see Paragraph 5.4.1), a 1 FULL-DSP-chain based V-TDL (see Paragraph 5.3), a 1 CARRY-chain based TDL (see Paragraph 5.2), and a 1 HALF-DSP-chain based TDL (see Paragraph 5.3).

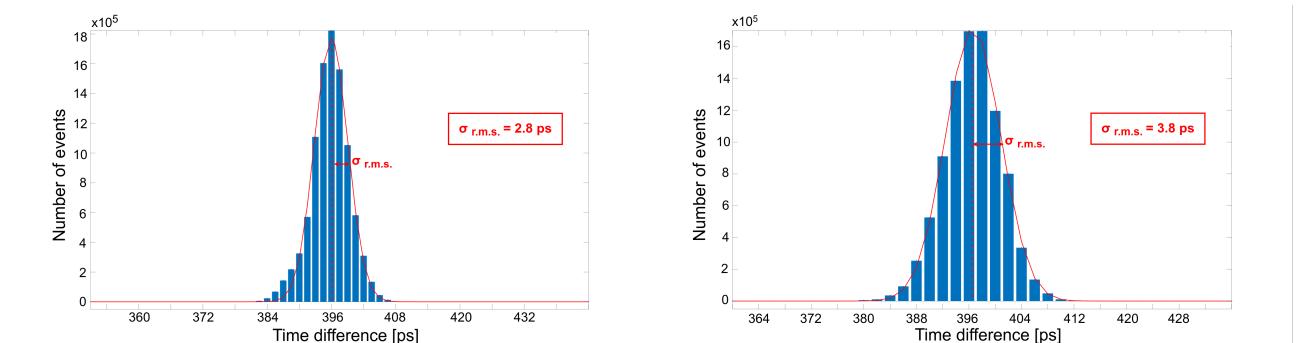


Figure 60: Histograms representing the time measurement in the case of a V-TDL composed of 4 CARRY-chains (on the left, i.e., $\sigma_{r.m.s.} = 2.8$ ps, $\sigma_{CH} = 1.98$ ps) and of 2 CARRY-chains + 1 FULL-DSP-chain (on the right, i.e., $\sigma_{r.m.s.} = 3.8$ ps, $\sigma_{CH} = 2.7$ ps).

Feature	4 CARRY-chains	2 CARRY-chains + 1 FULL-DSP-chain
Clock TDC Freq.		500 MHz
$N_B / M / N_V$		512/4/2048
Full-Scale Range		some days
Max. Channel Rate		200 MHz
Dead-Time		2 ns
Single-Shot Precision ($\sigma_{r.m.s}$)	2.8 ps	3.8 ps
TDL mean propagation delay ($t_{p,V}$)	1.2 ps	1.7 ps
TDC ultra-bin	7 ps	11.3 ps
LUT/FF/BRAM (for one ch.)	5431/7716/3.5	5429/6691/3.5
Number of Chs (*)	up to 62	up to 74
Power Consumption (for one ch.)	0.596 W	0.49 W

Table 16: Performances and hardware usage of a V-TDL composed of 4 CARRY-chains in parallel (first implementation) and of 2 CARRY-chains + 1 FULL-DSP-chain in parallel (second implementation). (*) in the xcku040-ffva1156-2-e.

bin”. As shown in Figure 60, this allows the first implementation achieving a better single-shot precision rather than the second implementation. Anyhow, the second implementation has as strength points the lower power consumption (i.e., 0.1 W less) and the more relaxed hardware utilization (i.e., 1000 FFs less), which makes it possible to have a greater number of TDC-channels (i.e., 12 channels more, in the xcku040-ffva1156-2-e) on the FPGA fabric. Table 16 summarizes the performances and the count of hardware resources in both the implementations under study.

In conclusion, each one of the two implementations studied in this experiment showed its strengths: the one with a fully-CARRY-based V-TDL has a single-shot precision in the order of 3 ps, thus being the most suitable choice for those applications requiring very high precision in the time measurement; conversely, the “hybrid” one (i.e., 2 CARRY-chains + 1 FULL-DSP-chain) saves a lot of FFs resources and power, and it is more suitable for those applications requiring a higher number of channels to detect a higher number of physical events.

5.6. Linearity of the Time-to-Digital Converter

This last experiment aimed at measuring the non-linear behavior of the TDC. To quantify the linearity of the system, a Code-Density Test (CDT) has been performed with the BeltBus_TDCHistogrammer module, by measuring equally distributed time intervals. In particular, by defining as LSB_{histo} the width of each histogram’s bin, increasingly greater time intervals (of a quantity equal to LSB_{histo}) have been measured, covering the entire measurement range FSR_{histo} . In a perfectly ideal, free from non-linearities, TDC, the resulting histogram should be completely flat, meaning that every equally distributed time interval would have the same probability of being measured over the entire range. However, this is not the case, since there are relevant sources of non-linearity in the system, the first one being the TDL itself, which is not optimized on the FPGA, and the second being Cross-Talk (XT) [20]. XT is a phenomenon of electromagnetic interference that can corrupt the physical signal to be measured, thus giving a wrong time measurement. The higher the operating frequency and the density of the components in the system, the more critical this phenomenon will be.

By measuring the “deviation” of the histogram from his ideal, flat, behavior, the information about the system’s non-linearity can be retrieved. This “deviation” is nothing but the DNL, already presented in Paragraph 1.3.4, while the integration of the latter is the INL.

5.6.1. DNL and INL calculation

Let's see in detail how to calculate the DNL and the INL. By calling N_{bins} the total number of bins of the histogram, we have that:

$$N_{bins} = \frac{FSR_{histo}}{LSB_{histo}} \quad (38)$$

By denoting as $h_{CD}[n]$ (with $n \in [0; N_{bins} - 1]$) the height of each bin, representing the number of events falling on it, we have the following relation:

$$N_{SAMPLE} = \sum_{n=0}^{N_{bins}-1} h_{CD}[n] \quad (39)$$

where N_{SAMPLE} is the total number of performed measurements. The average number of events on each bin is equal to:

$$N_{CD} = \frac{1}{N_{bins}} \cdot \sum_{n=0}^{N_{bins}-1} h_{CD}[n] = \frac{N_{SAMPLE}}{N_{bins}} \quad (40)$$

Finally, the deviation of each bin's value from its average can be calculated, thus deriving the so-called relative DNL:

$$dnl_{Rel}[n] = \frac{h_{CD}[n] - N_{CD}}{N_{CD}} \quad (41)$$

By multiplying the relative DNL by the LSB of the histogram, the absolute DNL is obtained:

$$dnl_{Abs}[n] = dns_{Rel}[n] \cdot LSB_{histo} \quad (42)$$

Once the results in Equations 41 and 42 have been calculated, the INL can be retrieved by simply integrating the DNL curves:

$$inl_{Rel}[n] = \sum_{i=0}^{i=n} dns_{Rel}[i] \quad (43)$$

$$inl_{Abs}[n] = \sum_{i=0}^{i=n} dns_{Abs}[i] \quad (44)$$

5.6.2. Experimental Setup

To generate a set of time intervals uniformly distributed along the entire FSR_{histo} , two uncorrelated signal generators have been used, working at different frequencies [20]. Once the Start-Stop Generator module has provided all the samples needed to perform the Calibration of the TDL, the two external signal generators have been used to provide the physical events to Channel 1 and Channel 2 of the TDC. Figure 61 shows the used experimental setup, where we can see that an Agilent 33120A Arbitrary Waveform Generator [38] has been connected to Channel 1, and a Juntek JDS-2900 Signal Generator [12] has been connected to Channel 2.

The signal on channel 1 is a square wave with a frequency f_1 equal to 1000016 Hz, while the signal on channel 2 is a square wave with a frequency f_2 equal to 1000014 Hz. This results in a period $T_1 = 999.984 \text{ ns}$ for the signal on channel 1, and a period $T_2 = 999.986 \text{ ns}$ for the signal on channel 2. Therefore, a uniform distribution of time intervals have been generated, since:

$$T_1 = FSR \quad (45)$$

$$T_2 = T_1 + LSB \quad (46)$$

where $LSB = 2 \text{ ps}$.



Figure 61: Experimental Setup used for the linearity test of the TDC.

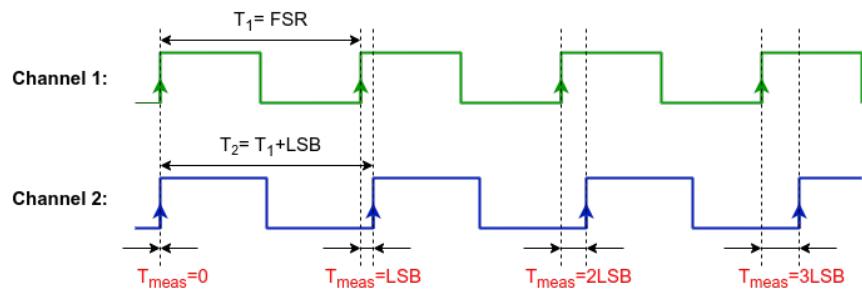


Figure 62: Equally distributed time intervals obtained from the two signals at f_1 and f_2 frequencies.

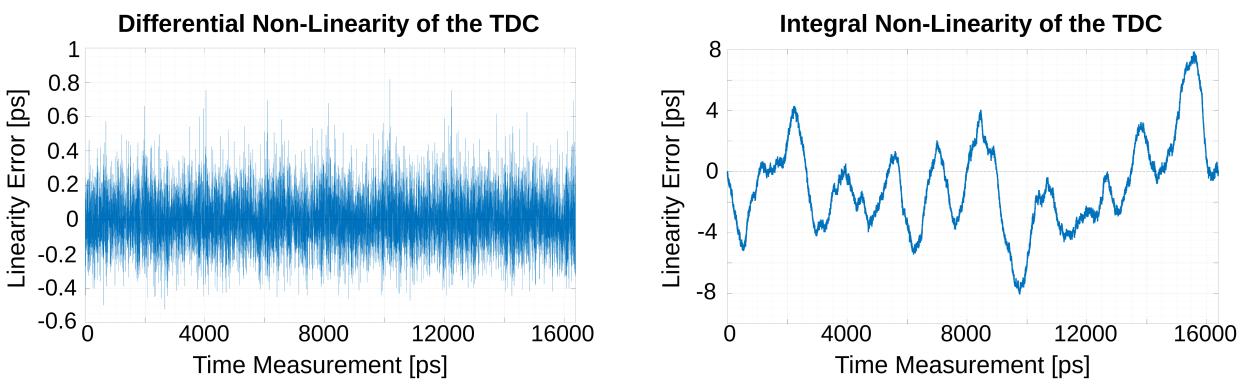


Figure 63: Absolute DNL (on the left) and INL (on the right) for a TDC with a TDL composed of 1 CARRY-chain.

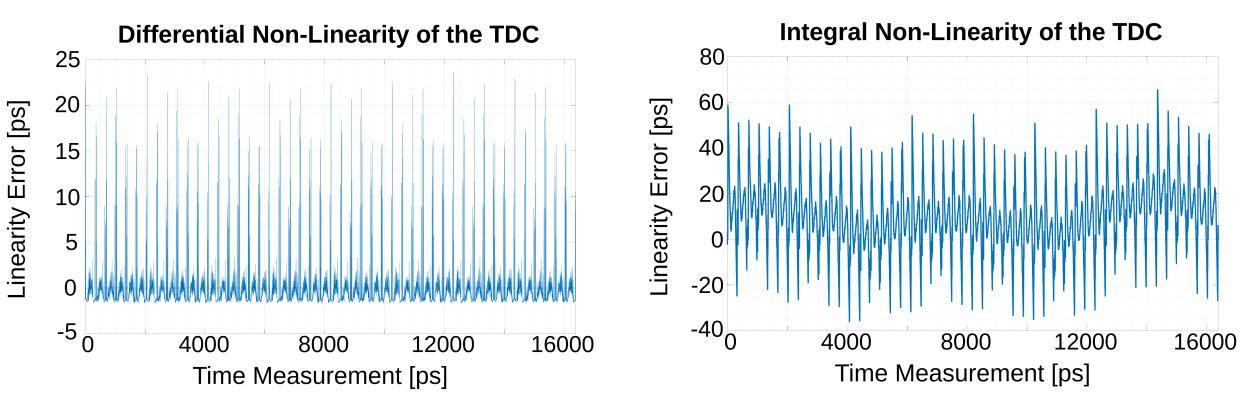


Figure 64: Absolute DNL (on the left) and INL (on the right) for a TDC with a V-TDL composed of 1 FULL-DSP-chain.

Figure 62 graphically shows the signals provided by the Waveform Generators.

To obtain the measurement of the DNL and the INL of the TDC, a histogram with $LSB_{histo} = LSB = 2 \text{ ps}$ and $FSR_{histo} \leq FSR = 999.984 \text{ ns}$ (in order to cover the entire measurement range) has been acquired. In this experiment, a $FSR_{histo} = 16 \text{ ns}$ and a total number of measurements equal to $N_{SAMPLE} = 10^7$ have been chosen.

5.6.3. Experimental results

The $dnl_{Abs}[n]$ and the $inl_{Abs}[n]$ have been measured in four different cases, involving four different V-TDL implementations. Figures 63, 64, 65, and 66 show the DNL and the INL for, respectively: a TDL composed of 1 CARRY-chain; a V-TDL composed of 1 FULL-DSP-chain; a V-TDL composed of 4 CARRY-chains in parallel; a V-TDL composed of 2 CARRY-chains + 1 FULL-DSP-chain in parallel. By calling DNL_{Abs} and INL_{Abs} the maximum value of the modulus of $dnl_{Abs}[n]$ and $inl_{Abs}[n]$ respectively, we can conclude that:

- The TDC with 1 FULL-DSP-chain has a worse non-linearity (i.e. $DNL_{Abs} = 23 \text{ ps}$, $INL_{Abs} = 65 \text{ ps}$), due to its huge “ultra-bins”.
- The SuperWU Sub-Interpolation algorithm on the TDL improves the overall linearity of the TDC, as we can see by comparing the TDC based on 1 CARRY-chain (i.e. $DNL_{Abs} = 0.82 \text{ ps}$, $INL_{Abs} = 8 \text{ ps}$) and the one with 4 CARRY-chains in parallel (i.e. $DNL_{Abs} = 0.22 \text{ ps}$, $INL_{Abs} = 5.8 \text{ ps}$).
- The TDC using 4 CARRY-chains in parallel is characterized by the most linear behavior (i.e. $DNL_{Abs} = 0.22 \text{ ps}$, $INL_{Abs} = 5.8 \text{ ps}$), followed by the one using 2 CARRY-chains + 1 FULL-DSP-chain in parallel, which has slightly worse linearity (i.e. $DNL_{Abs} = 0.3 \text{ ps}$, $INL_{Abs} = 12 \text{ ps}$).

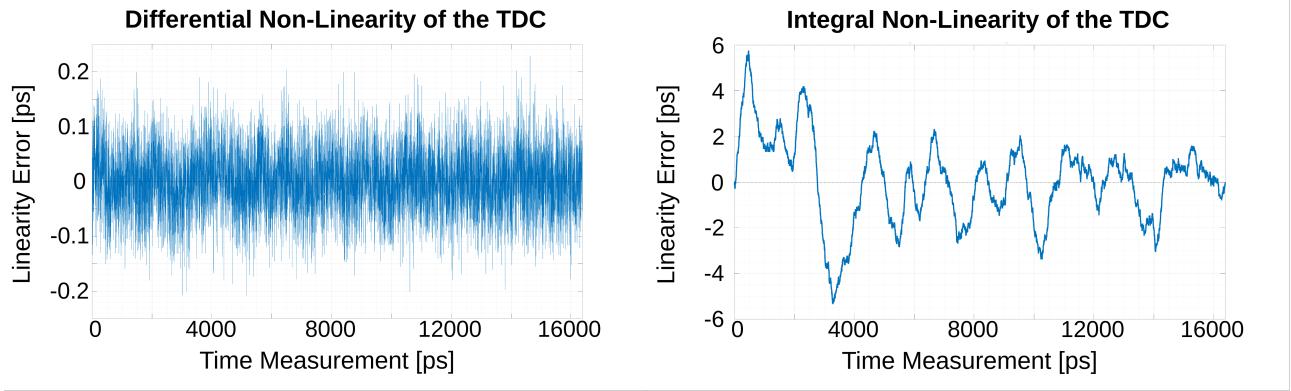


Figure 65: Absolute DNL (on the left) and INL (on the right) for a TDC with a V-TDL composed of 4 CARRY-chains.

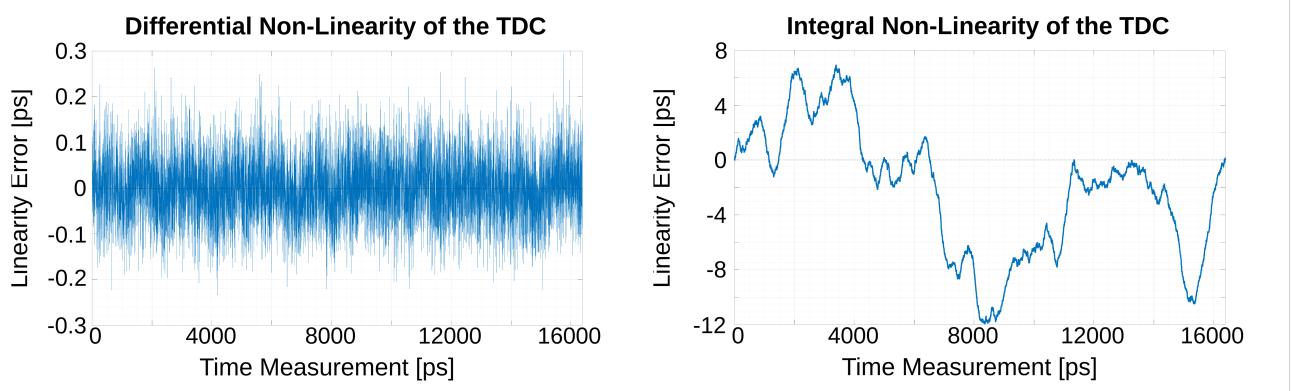


Figure 66: Absolute DNL (on the left) and INL (on the right) for a TDC with a V-TDL composed of 2 CARRY-chains + 1 FULL-DSP-chain.

Implementation	DNL_{Abs}	INL_{Abs}
1 CARRY-chain-based TDC	0.82 ps	8 ps
1 FULL-DSP-chain-based TDC	23 ps	65 ps
4 CARRY-chains-based TDC	0.22 ps	5.8 ps
2 CARRY-chains + 1 FULL-DSP-chain-based TDC	0.3 ps	12 ps

Table 17: Absolute DNL and INL for the TDCs in Figures 63 (i.e., 1 CARRY-chain), 64 (i.e., 1 FULL-DSP-chain), 65 (i.e., 4 CARRY-chains), and 66 (i.e., 2 CARRY-chains + 1 FULL-DSP-chain).

Table 17 summarizes the achieved results for the four implementations in Figures 63, 64, 65, and 66.

6. Conclusions

To cope with the increasing demand for performant Time Interval Meters (TIMs) in time-resolved experiments, more and more performances such as resolution, single-shot precision, and multi-hit detection are required for these kinds of instruments. In this thesis work, a high-resolution Time-to-Digital Converter (TDC) Intellectual Property Core (IP-Core) for Field Programmable Gate Arrays (FPGAs) has been implemented. Thanks to the user-friendly IP-Core's Graphical User Interface (GUI), high portability and tunability are guaranteed. The GUI allows the user to set the TDC's parameters as FSR, resolution (LSB), and the number of channels in the instantiation stage; besides, portability on different FPGA systems is allowed thanks to a simple drag-and-drop action on the IP-Core. The system is also compatible with different FPGA technology nodes, thus further increasing its portability, and exploiting more scaled and performant technologies. By simply choosing an option from the GUI, the user can implement the TDC either on 28-nm Xilinx 7-Series technology (X7S) or on 20/16-nm Xilinx Ultrascale/Ultrascale+ (XUS/XUS+) technologies, the latter pushing even more the TDC's performances thanks to the scaling. Two different TDCs have been implemented, both using the Super Wave Union (SuperWU) Sub-Interpolation algorithm with four Tapped Delay-Lines (TDLs) to improve the LSB and the single-shot precision of the TDC. These two implementations aimed to study the possible trade-offs existing between performance and area/power consumption. Eventually, the choice for the first implementation fell on a TDC based on a 4 CARRY-chains V-TDL, which maximizes the single-shot precision; on the other hand, a 2 CARRY-chains + 1 FULL-DSP-chain TDC has been chosen as the second implementation, and it is characterized by an area/power relaxation. Both implementations have been tested on Xilinx's KCU105 Evaluation Board, hosting a Kintex-UltraScale™ FPGA (xcku040-ffva1156-2-e). The following experimental results have been obtained:

- a higher clock frequency, equal to 500 MHz, has been reached in both implementations thanks to the improved scaling of the XUS technology, allowing it to lower the Dead-Time (i.e., 2 ns) and improve the multi-hit capability of the TDC;
- a better single-shot precision has been achieved on the 4 CARRY-chains implementation (i.e., 2.8 ps) than on the 2 CARRY-chains + 1 FULL-DSP-chain one (i.e., 3.8 ps), making the former more suitable for those applications requiring very high precision;
- lower power consumption (i.e., 0.1 W less) and hardware occupancy have been achieved (i.e., 1000 FFs less) on the 2 CARRY-chains + 1 FULL-DSP-chain implementation than in the 4 CARRY-chains, allowing the former to have a greater number of channels (i.e., 74). It is thus more suitable for those applications requiring a higher number of physical events to be detected;
- both implementations have negligible non-linearity since the DNLs are equal to 0.22 ps and 0.3 ps for the 4 CARRY-chains and the 2 CARRY-chains + 1 FULL-DSP-chain implementation, respectively, and the INLs are equal to 5.8 ps and 12 ps, respectively.

References

- [1] ARM. *AMBA4 AXI4-Stream Protocol Version 1.0*, march 2010. 21
- [2] Joseph M. Beechem and Ludwig Brand. Time-resolved fluorescence of proteins. *Annual Review of Biochemistry*, 54(1):43–71, 1985. 3
- [3] B Chance, J S Leigh, H Miyake, D S Smith, S Nioka, R Greenfeld, M Finander, K Kaufmann, W Levy, and M Young. Comparison of time-resolved and -unresolved measurements of deoxyhemoglobin in brain. *Proceedings of the National Academy of Sciences*, 85(14):4971–4975, 1988. 3
- [4] Chun-Chi Chen, Shih-Hao Lin, and Chorng-Sii Hwang. An area-efficient cmos time-to-digital converter based on a pulse-shrinking scheme. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(3):163–167, 2014. 15
- [5] N. Corna, E. Ronconi, F. Garzetti, S. Salgaro, N. Lusardi, L. Tavazzani, and A. Geraci. High-performance physical-independent address-based communication interface for fpga in custom scientific equipment. In *2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–4, 2020. 40
- [6] Nicola Corna, Fabio Garzetti, Nicola Lusardi, and Angelo Geraci. Digital instrument for time measurements: Small, portable, high-performance, fully programmable. *IEEE Access*, 9:123964–123976, 2021. 20
- [7] Andrea Costa, Nicola Corna, Fabio Garzetti, Nicola Lusardi, Enrico Ronconi, and Angelo Geraci. High-performance computing of real-time and multichannel histograms: A full fpga approach. *IEEE Access*, 10:47524–47540, 2022. 40
- [8] Ke Cui and Xiangyu Li. A high-linearity vernier time-to-digital converter on fpgas with improved resolution using bidirectional-operating vernier delay lines. *IEEE Transactions on Instrumentation and Measurement*, 69(8):5941–5949, 2020. 12, 13
- [9] Ryuichi Enomoto, Tetsuya Iizuka, Takehisa Koga, Toru Nakura, and Kunihiro Asada. A 16-bit 2.0-ps resolution two-step tdc in 0.18- μ m cmos utilizing pulse-shrinking fine stage with built-in coarse gain calibration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(1):11–19, 2019. 15
- [10] F. Garzetti, N. Lusardi, A. Geraci, E. Dobovicnik, G. Cautero, C. Dri, R. Sergo, and L. Stebel. Fully fpga-based and all-reconfigurable tdc for 3d (x, y, t) cross delay-line detectors. In *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, pages 1–3, 2018. 19
- [11] Fabio Garzetti, Nicola Corna, Nicola Lusardi, and Angelo Geraci. Time-to-digital converter ip-core for fpga at state of the art. *IEEE Access*, 9:85515–85528, 2021. 6, 9, 12, 16, 20, 24
- [12] Ltd. Hangzhou Junce Instrument Co. *JDS2900 series Dual-Channel Arbitrary DDS Signal Generator Quick Start Guide*, June 2019. p. 2. 56
- [13] S. Henzler. *Time-to-Digital Converters*. Springer, August 2010. 6
- [14] J. Kostamovaara, K. Maatta, T. Rahkonen, and R. Rankinen. Ecl and cmos asics for time-to-digital conversion. In *Proceedings., Second Annual IEEE ASIC Seminar and Exhibit,*, pages P5–2/1, 1989. 9
- [15] Paweł Kwiatkowski. Digital-to-time converter for test equipment implemented using fpga dsp blocks. *Measurement*, 177:109267, 2021. 34

- [16] N. Lusardi, F. Garzetti, N. Corna, R. De Marco, and A. Geraci. Very high-performance 24-channels time-to-digital converter in xilinx 20-nm kintex ultrascale fpga. In *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–4, 2019. 19
- [17] N. Lusardi, F. Garzetti, R. De Marco, and A. Geraci. Implementation issues of a high-performance multi-channel time-to-digital converter in xilinx 20-nm ultrascale fpgas. In *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, pages 1–4, 2018. 19
- [18] N. Lusardi, S. Salgaro, A. Costa, N. Corna, G. Garzetti, E. Ronconi, and A. Geraci. High-channel count fpga-based single-phase shift-clock fast-counter time-to-digital converter. In *2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–4, 2021. 11
- [19] N. Lusardi, S. Salgaro, F. Garzetti, N. Corna, G. Ticozzi, and A. Geraci. Fpga-based multi-phase shift-clock fast-counter time-to-digital converter for extremely-large number of channels. In *2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–4, 2020. 12
- [20] Nicola Lusardi, Nicola Corna, Fabio Garzetti, Simone Salgaro, and Angelo Geraci. Cross-talk issues in time measurements. *IEEE Access*, 9:129303–129318, 2021. 55, 56
- [21] Nicola Lusardi, Fabio Garzetti, Andrea Costa, Marco Cautero, Nicola Corna, Enrico Ronconi, Gabriele Brajnik, Luigi Stebel, Rudi Sergio, Giuseppe Cautero, Sergio Carrato, and Angelo Geraci. High-resolution imager based on time-to-space conversion. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022. 20
- [22] Nicola Lusardi, Fabio Garzetti, and Angelo Geraci. The role of sub-interpolation for Delay-Line Time-to-Digital Converters in FPGA devices. *Nuclear Instruments and Methods in Physics Research A*, 916:204–214, Feb 2019. 18, 19, 44, 48
- [23] Jan Nissinen, Ilkka Nissinen, and Juha Kostamovaara. Integrated receiver including both receiver channel and tdc for a pulsed time-of-flight laser rangefinder with cm-level accuracy. *IEEE Journal of Solid-State Circuits*, 44(5):1486–1497, 2009. 4
- [24] Santi Nonell and Silvia E. Braslavsky. [4] time-resolved singlet oxygen detection. In *Singlet Oxygen, UV-A, and Ozone*, volume 319 of *Methods in Enzymology*, pages 37–49. Academic Press, 2000. 3
- [25] R. Nutt. Digital time intervalometer. *Review of Scientific Instruments*, 39:1342–1345, 1968. 17, 31
- [26] L Perktold and J Christiansen. A multichannel time-to-digital converter asic with better than 3 ps rms time resolution. *Journal of Instrumentation*, 9(01):C01060, jan 2014. 15
- [27] Ankur Pokhara, Jatin Agrawal, and Biswajit Mishra. Design of an all-digital, low power time-to-digital converter in 0.18m cmos. In *2017 7th International Symposium on Embedded Computing and System Design (ISED)*, pages 1–5, 2017. 14, 15
- [28] Schaart D. R. Physics and technology of time-of-flight pet detectors. In *Physics in medicine and biology*, volume 66, 2021. 4
- [29] T. E. Rahkonen and J. T. Kostamovaara. *The use of stabilized cmos delay lines for the digitization of short time intervals*, volume 28. IEEE Journal of Solid-State Circuits, August 1993. 12
- [30] Marco Renna, Mauro Buttafava, Anurag Behera, Marta Zanoletti, Laura Di Sieno, Alberto Dalla Mora, Davide Contini, and Alberto Tosi. Eight-wavelength, dual detection channel instrument for near-infrared time-resolved diffuse optical spectroscopy. *IEEE Journal of Selected Topics in Quantum Electronics*, 25(1):1–11, 2019. 5

- [31] D. Resnati, I. Rech, A. Gallivanoni, and M. Ghioni. *Monolithic time to amplitude converter for time correlated single photon counting*, volume 80. Review of Scientific Instruments, 2009. 10
- [32] Enrico Ronconi, Nicola Corna, Andrea Costa, Fabio Garzetti, Nicola Lusardi, and Angelo Geraci. Multi-cobs: A novel algorithm for byte stuffing at high throughput. *IEEE Access*, 10:78848–78859, 2022. 40
- [33] Erik Sall and Mark Vesterbacka. Thermometer-to-binary decoders for flash analog-to-digital converters. In *2007 18th European Conference on Circuit Theory and Design*, pages 240–243, 2007. 29
- [34] V. Sanchez-Tembleque, L. M. Fraile, and J. M. Udiás. Time over threshold data acquisition system for pet. In *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–3, 2017. 7
- [35] Qi Shen, Shubin Liu, Binxiang Qi, Qi An, Shengkai Liao, Peng Shang, Chengzhi Peng, and Weiyue Liu. A 1.7 ps equivalent bin size and 4.2 ps rms fpga tdc based on multichain measurements averaging method. *IEEE Transactions on Nuclear Science*, 62(3):947–954, 2015. 15, 16
- [36] R Szplet, Z Jachna, P Kwiatkowski, and K Rozyc. A 2.9 ps equivalent resolution interpolating time counter based on multiple independent coding lines. *Measurement Science and Technology*, 24(3):035904, feb 2013. 15, 16
- [37] B.J. Pearson T. Weinacht. *Time-Resolved Spectroscopy An Experimental Perspective*. CRC Press, 2018. 5
- [38] Agilent Technologies. *Agilent 33120A Arbitrary Waveform Generator User's Guide*, March 2002. p. 6. 56
- [39] J. Torres, A. Aguilar, R. García-Olcina, P. A. Martínez, J. Martos, J. Soret, J. M. Benlloch, P. Conde, A. J. González, and F. Sánchez. Time-to-digital converter based on fpga with multiple channel capability. *IEEE Transactions on Nuclear Science*, 61(1):107–114, 2014. 15, 16
- [40] Jinhong Wang, Yu Liang, Xiong Xiao, Qi An, John W. Chapman, Tiesheng Dai, Bing Zhou, Junjie Zhu, and Lei Zhao. Development of a time-to-digital converter ASIC for the upgrade of the ATLAS monitored drift tube detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 880:174–180, feb 2018. 15
- [41] Yonggang Wang, Peng Kuang, and Chong Liu. A 256-channel multi-phase clock sampling-based time-to-digital converter implemented in a kintex-7 fpga. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–5, 2016. 15, 16
- [42] Jun Yeon Won and Jae Sung Lee. Time-to-digital converter using a tuned-delay line evaluated in 28-, 40-, and 45-nm fpgas. *IEEE Transactions on Instrumentation and Measurement*, 65(7):1678–1689, 2016. 23
- [43] Zhong Wu, Yue Xu, and Zhiqiang Ma. A time-to-amplitude converter with high impedance switch topology for single-photon time-of-flight measurement. *IEEE Access*, 9:16672–16678, 2021. 11
- [44] Qin Xi, Feng Changqing, Zhang Deliang, Zhao Lei, Shubin Liu, and An Qi. A low dead time vernier delay line tdc implemented in an actel flash-based fpga. *Nuclear Science and Techniques*, 24, 08 2013. 15, 16
- [45] Xilinx. *7 Series FPGAs Clocking Resources*, aug 2013. User Guide 472. 11
- [46] Xilinx. *7 Series FPGAs Configurable Logic Block*, sep 2016. User Guide 474. 23, 27
- [47] Xilinx. *UltraScale Architecture Configurable Logic Block User Guide*, February 2017. UG574, p. 23.

- [48] Xilinx. *7 Series DSP48E1 Slice User Guide*, March 2018. UG479, p. 14. 35
- [49] Xilinx. *KCU105 Board User Guide*, February 2019. UG917, p. 6. 40
- [50] Xilinx. *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator*, oct 2019. User Guide 994. 21
- [51] Xilinx. *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics*, mar 2021. 24
- [52] Xilinx. *UltraScale Architecture Clocking Resources*, aug 2021. User Guide 572. 11
- [53] Xilinx. *UltraScale Architecture DSP Slice User Guide*, August 2021. UG579, p. 15. 35
- [54] Xilinx. *Virtex-7 T and XT FPGAs Data Sheet: DC and AC Switching Characteristics*, mar 2021. 24
- [55] Xilinx. *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics*, feb 2022. 24
- [56] C.S-C. Yang, E. Brown, U. Hommerich, S.B. Trivedi, A.P. Snyder, and A.C. Samuels. Mid-infrared atomic and molecular laser-induced breakdown spectroscopy emissions from solid substances. In *2009 Conference on Lasers and Electro-Optics and 2009 Conference on Quantum electronics and Laser Science Conference*, pages 1–2, 2009. 5
- [57] Fei Yuan and Parth Parekh. Time-mode all-digital delta-sigma time-to-digital converter with process uncertainty calibration. In *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 489–492, 2019. 9, 18, 32
- [58] Qiuchen Yuan, Bowei Zhang, Jerry Wu, and Mona E. Zaghloul. A high resolution time-to-digital converter on fpga for time-correlated single photon counting. In *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 900–903, 2012. 5
- [59] Jiajun Zheng, Ping Cao, Di Jiang, and Qi An. Low-cost fpga tdc with high resolution and density. *IEEE Transactions on Nuclear Science*, 64(6):1401–1408, 2017. 15, 16
- [60] Sihui Zhu, Yue Xu, Ding Li, and Zhong Wu. A sample and hold time-to-amplitude converter for single photon time-of-flight measurement. In *2019 IEEE 9th Symposium on Computer Applications Industrial Electronics (ISCAIE)*, pages 316–319, 2019. 10

Abstract in lingua italiana

La crescente richiesta di misurazioni temporali ad alta precisione nelle applicazioni scientifiche (le quali spaziano dal campo biomedico a quello industriale), ha portato al bisogno di strumenti, chiamati Misuratori di Intervalli di Tempo (TIMs), caratterizzati da una risoluzione e una precisione nell' ordine dei picosecondi (ps), e da un rate di campionamento nell' ordine delle centinaia di megahertz (MHz), in grado quindi di processare correttamente gli eventi fisici rilevati. Diverse soluzioni per i TIM sono presenti nella letteratura scientifica, implementate sia in ASIC che in FPGA. Per soddisfare bisogni come una veloce prototipazione e un veloce tempo di sviluppo, in questo lavoro abbiamo scelto l' approccio FPGA, implementando un TIM interamente digitale che prende il nome di Convertitore Tempo-Digitale (TDC). Il TIM che presenteremo é un TDC a piú canali basato su una struttura a Tapped Delay-Line (TDL), sottoforma di IP-Core per gli FPGA Ultrascale/Ultrascale+ di Xilinx (XUS/XUS+). L' approccio IP-Core garantisce un' alta portabilitá poiché il TDC puó essere prontamente testato su diversi sistemi con una semplice operazione di trascinamento. Per sfruttare i benefici dello scaling, il nodo tecnologico XUS a 20-nm é stato investigato in questo lavoro, portando infine a ottenere alti rate di misura, un basso Dead-Time, un' alta risoluzione, e una precisione eccellente. In particolare, una precisione nell' ordine dei 3 ps é stata raggiunta, insieme a un massimo rate di misura pari a 200 Msps per ciascun canale, soddisfacendo cosí il bisogno di un alto rate richiesto dai detector allo stato dell' arte. Dal punto di vista della progettazione, il TDC é stato implementato e interamente testato su un Kintex-Ultrascale FPGA, contenuto in una KCU105 Evaluation Board. Con lo scopo di investigare i possibili trade-off tra consumo di potenza e precisione, e tra occupazione d' area e precisione, due diverse tipologie di TDL sono state testate. La prima soluzione é piú performante in termini di precisione, a discapito di una maggiore occupazione d' area, ed é basata sulle primitive di CARRY (cioé la primitiva CARRY8) disponibili sul fabric dell' FPGA. La seconda soluzione é invece meno performante ma piú efficiente a livello di consumo di potenza e risorse hardware, ed é basata sulle primitive di Digital Signal Processing (DSP) (cioé la primitiva DSP48E2). Infine, un' architettura ibrida che sfrutta sia risorse di CARRY che di DSP é stata testata, portando a risultati che mostrano i punti di forza di entrambe le tipologie di TDL (cioé una precisione di 3.8 ps, un consumo di potenza di 0.49 W, e un massimo numero concesso di canali pari a 74). Questa architettura é stata confrontata con quella puramente basata su catene di CARRY, la quale é la piú performante in termini di precisione (pari a 2.8 ps), ma é caratterizzata da un maggiore consumo di potenza (pari a 0.596 W) e di area (che porta ad avere al massimo 62 canali).

Parole chiave: Convertitore Tempo-Digitale (TDC), Tapped Delay-Line (TDL), Digital Signal Processor (DSP), Field-Programmable-Gate-Array (FPGA)

Acknowledgements

Firstly, I would like to thank my thesis supervisor, Prof. Nicola Lusardi, for believing in me so much during all the time I spent at the DigiLAB and for teaching me so many things regarding the digital electronics world. I also want to thank Prof. Angelo Geraci, for giving me the possibility to carry out this thesis work in his laboratory, along with Dr. Fabio Garzetti, Enrico Ronconi, and Andrea Costa, for their fundamental help during all this time. A big thanks also go to my DigiLAB 2 mates André, Gabriele, and Luca for creating a fantastic working atmosphere and making the days lighter.

A special thanks go to my family, for making me the person I am nowadays, and without whom this University career would not have been possible.

I owe also a special thanks to my girlfriend Eleonora for being part of my life and making it better, and for believing in me every single day.

Last but not least, I want to thank all my friends Andre, Fra, Jack, Lisa, Lori, Mark, Marti, Nati, Tessy, and Tona for their presence throughout the years and for all the good times spent together.