

Multi-Channel FPGA Time-to-Digital Converter With 10 ps Bin and 40 ps FWHM

Davide Portaluppi¹, Klaus Pasquinelli¹, Iris Cusini¹, and Franco Zappa², *Senior Member, IEEE*

Abstract—We present a novel architecture for multi-channel time-to-digital converters (TDCs) to be implemented into low-cost field-programmable gate arrays (FPGAs), achieving 10-ps least significant bit (LSB), 164- μ s full-scale range, and good linearity both in terms of differential nonlinearity (DNL) and integral nonlinearity (INL). The conceived architecture is based on the carry chain delay line model and wave union A method: the positions of both rising and falling edges that propagate in multiple parallel carry chains are recorded each time there is an HIT input. This technique effectively subdivides the ultrawide bins improving the measurement precision and, combined with the sliding-scale technique and continuous code density calibration, improves the TDC linearity. Employing the proposed architecture, we have implemented in a Xilinx Artix-7 FPGA a TDC with 20 timestamp units and validated the device in a time-correlated single photon counting (TCSPC) setup, when connected to an array chip with 5×5 single-photon avalanche diodes (SPADs).

Index Terms—Single-photon avalanche diode (SPAD), tapped delay line (TDL), time-correlated single photon counting (TCSPC), time-to-digital converter (TDC).

I. INTRODUCTION

TIME-TO-DIGITAL converters (TDCs) are devices able to convert time delays into digital numbers, aiming at high time resolution, precision, conversion speed, and low dead time. Among many other applications, TDCs are exploited in many scientific fields requiring the precise measurement of the arrival time of photons, such as in particle and high-energy physics [1], laser ranging [2], [3], time-of-flight positron emission tomography (ToF-PET) [4], fluorescence lifetime imaging (FLIM), quantum imaging [5], and so on. Besides TDC resolution, i.e., the least significant bit (LSB), which determines the shortest time delay difference that can be measured single shot, other parameters, such as precision, accuracy, full-scale range (FSR), speed, and number of parallel channels, strongly depend on the specific application.

TDCs developed as application-specific integrated circuits (ASICs) can be extremely performing, since they allow

the customization of the architecture and the fine tailoring of specific target performance. They reach single-shot time-jitter precision better than 1 ps [6] and allow the parallelization of many channels [7]. On the other hand, an ASIC design implies high nonrecurrent engineering (NRE) costs, design time, and long manufacturing time. Instead, field-programmable gate array (FPGA) implementations offer faster deployment time, lower development costs, easier reconfigurability, and parameters adjustment and can be more easily migrated to newer FPGA releases. Eventually, FPGAs find applications in widespread fields [8]–[10] because, even if they may not attain the same performance of custom ASICs, their programmability, design portability, and fast prototyping are definitely key factors.

A typical FPGA implementation of a TDC employs a clock counter for the coarse most-significant bits (MSBs) time-stamping and a gates chain delaying an input signal for the fine LSB quantization. When a valid signal triggers the delay line, the next clock pulse samples the position in the delay line and the value of the free-running coarse counter [11], [12]. Usually, in general-purpose FPGAs, logic gates are connected through dedicated carry lines, which offer short (tens of picoseconds) and fixed propagation delays. These elements are well suited for implementing tapped delay lines (TDLs) in FPGAs [11], [13]. Combining dedicated carry lines with a coarse counter, the achievable dynamic range widens. However, the unpredictable nonuniformity of the taps along the delay chain impairs the TDC linearity.

Various strategies to improve linearity and resolution can be found in literature. For example, Shen *et al.* [14] proposed a multichain measurements averaging method to achieve better performance. In [13], resolution and precision are instead increased by performing multiple measurements, and the conversion linearity is improved by a semicontinuous calibration, while, in [15], the clock skew at the border of the clock regions is exploited to enhance resolution and precision. However, all those approaches add complexity and dead time.

In this article, we present a low-cost FPGA-based TDC architecture aiming at fine resolution and linearity and high channel density. Unlike most TDCs already reported in literature, which are based on high-level high-gate count FPGAs, we employ a low-cost development board (OpalKelly XEM7310-A200) with a Xilinx Artix-7 FPGA reaching performance comparable with the one of more expensive counterparts. We discuss the design and implementation of each TDC stage, and we present a novel decoding strategy that

Manuscript received September 25, 2021; revised January 24, 2022; accepted January 28, 2022. Date of publication February 18, 2022; date of current version March 7, 2022. The Associate Editor coordinating the review process was Dr. Ziqiang Cui. (*Corresponding author: Klaus Pasquinelli.*)

Davide Portaluppi was with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milan, Piazza Leonardo da Vinci 32, 20133 Milan, Italy. He is now with Analog Devices, Inc., Wilmington, MA 01887, USA.

Klaus Pasquinelli, Iris Cusini, and Franco Zappa are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milan, Piazza Leonardo da Vinci 32, 20133 Milan, Italy (e-mail: klaus.pasquinelli@polimi.it; franco.zappa@polimi.it).

Digital Object Identifier 10.1109/TIM.2022.3152324

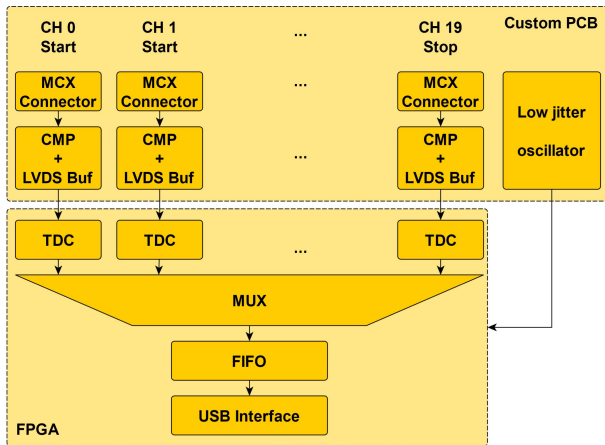


Fig. 1. Block diagram of the printed circuit board (PCB) and FPGA.

performs a subbranching of the TDC. By decoding only the section of the TDC that has recorded the event, both dead time and resources usage are reduced. Moreover, the proposed stage dynamically rearranges the sampled delay and solves the bubbles issue (explained in detail in Section II) without discarding any information. We report on the measured performance and linearity improvements achieved due to the proposed continuous calibration and bin merging. Eventually, we show the validation of our 19-channel TDC together with a single-photon avalanche diode (SPAD) array of 5×5 pixels.

This article is organized as follows. Section II describes the TDC architecture and implementation choices. Section III shows the characterization results. Section IV discusses the validation of the multi-channel TDC in a specific application. Section V summarizes the conclusion.

II. TDC ARCHITECTURE

The core of the proposed TDC is 20 identical and independent timestamp units; then, we have exploited the 20 units as a 19-channel TDC, with the purpose of measuring the time intervals between 19 START signals and a global STOP signal. Thus, we have employed 19 units to sample the different STARTs and one channel to sample the global STOP (Fig. 1). Using two different delay lines for the START and STOP signals allows us to take advantage of the sliding scale technique [16] and, thus, to increase the TDC linearity since the same time interval can be measured over different regions of the channel delay lines. Note, however, that each timestamp unit can be employed independently of the others or in any START–STOP combination. In the following, we refer to START and STOP inputs as HIT inputs.

The most intuitive and simplest way to implement a TDC is by using just high-speed counters. However, this approach limits the resolution to a clock period, e.g., to a few nanoseconds in low-cost 28-nm FPGAs (such as the Xilinx Artix 7). A common way to improve resolution is to take advantage of TDLs and exploiting dedicated carry lines within the FPGA fabric. As shown in Fig. 2, the HIT input feeds a fast TDL.

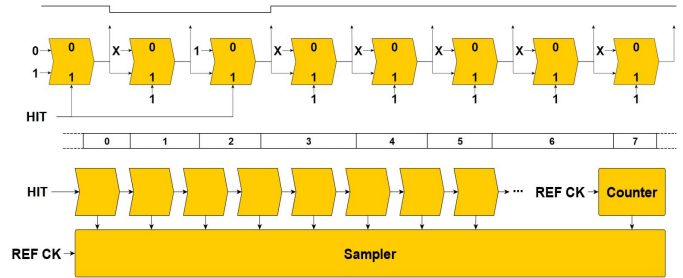


Fig. 2. Signal propagation through the carry chain and sampling provide the fine LSBs quantization, while a counter gives the coarse MSBs time-stamping.

When this signal is HIGH (logical “1”), the second input of the first and third taps propagates. The first input of the other taps is unknown (“X”) and does not affect the TDL behavior since they always propagate the second input (selecting input set to “1”). In this way, the code “101” moves through the carry chain, whose outputs are then sampled to extract the fine position of the propagating waveform, while a digital counter gives the coarse time-stamping. Once the propagation delay of each tap is known, the propagation time (t_{Fine}) is computed and combined with the coarse counter value (t_{Coarse}) to provide the signal arrival time.

More specifically, t_{Fine} is subtracted from t_{Coarse} , as the TDL measures the time elapsed between the HIT input and the next sampling clock pulse. In a START–STOP delay measurement, the time interval between the two HIT inputs is computed as the difference between the STOP arrival time and the START one

$$t = (t_{\text{coarseSTOP}} - t_{\text{fineSTOP}}) - (t_{\text{coarseSTART}} - t_{\text{fineSTART}}).$$

In our design, the TDLs are implemented by chaining several arithmetic carry propagation primitives (CARRY4) available in the Artix 7 slices, though the maximum achievable delay is limited by routing restrictions given the limited area of each clock region. Indeed, a TDL crossing clock region boundaries could compromise linearity, due to sudden variations in clock routing. To avoid trespassing clock regions, we have implemented chains with a maximum of 200 taps, and a 400-MHz sampling clock (2.5-ns period) has been chosen experimentally verifying that the delay line is sufficiently longer than 2.5 ns. This limits the FSR of the TDC; hence, the free-running 16-bit coarse counter has been added.

The actual length of the TDLs and the individual tap delays are not known a priori. Moreover, they can vary with process, voltage, and temperature (PVT) variations and from device to device, and they have no fixed relationship to the sampling clock. For this reason, each tap is calibrated through a code density test: a random HIT signal is fed to each channel and a histogram of the resulting TDL codes is collected.

Each tap has a width $T_{\text{bin}_i} = T_{\text{clock}} \times (N_{\text{bin}_i} / N_{\text{sample}})$, where T_{clock} is the sampling period of the delay line, N_{bin_i} is the number of events within bin i , and N_{sample} is the total number of samples. As shown in Fig. 3, taps’ widths are not constant across the delay line. Since the delay time corresponding to each bin is the cumulative sum of all preceding tap widths

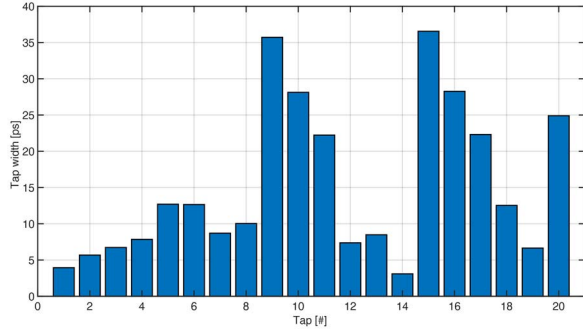


Fig. 3. Tap widths of a representative section of a single TDL before bin merging and calibration.

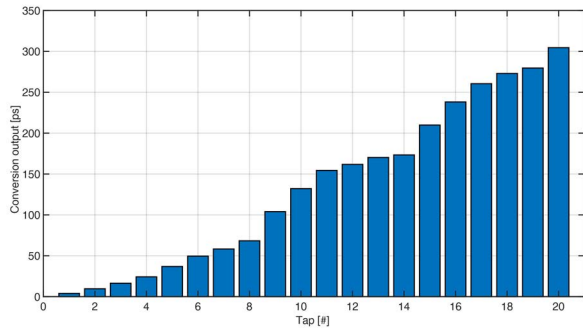


Fig. 4. Cumulative sum of the TDL tap widths.

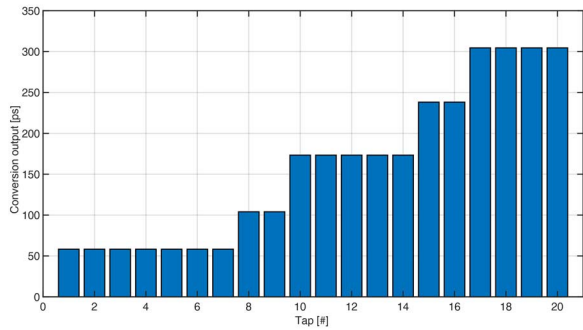


Fig. 5. Rebinning of the histogram in order to improve the DNL.

from Fig. 3, it is possible to compute the cumulative sum, as shown in Fig. 4. The final differential nonlinearity (DNL) can be improved by requantizing the bins, with a wider uniform step (bin merging) [17]. This technique consists in merging adjacent smaller bins into larger ones, in order to obtain a more uniform bin size. In this way, the TDC linearity increases at the expense of resolution and single-shot precision [18], [19]. Fig. 5 shows the results: for example, bins from 1 to 7 are merged in a single 52-ps bin.

To this purpose, we implement a LabVIEW subVI that, every 38 000 acquired samples, computes the cumulative sum of the arrival time histogram, executes bin-merging, and creates a conversion table for the next arrival events. However, the DNL improvement attainable through bin-merging is limited

to the width of the extrawide bins (e.g., 37 ps in Fig. 3). One solution to overcome this issue could be to manually change the position of TDLs within the FPGA, trying to avoid extrawide bins. However, this impacts repeatability, since each TDC should be tested and manually moved in a proper position within the FPGA. Another solution could be to repeat the delay measurement many times, until the HIT input does not fall in an extrawide bin, but this is not possible in all applications.

Instead, the most common solution exploits the wave union method [13], whose different implementations are divided into two subgroups. In wave union type A, a group of signal edges propagates through the TDL each time the HIT input is set, and their position along the chain is sampled just once. In wave union type B, the HIT input starts a ring oscillator, whose output is fed into the TDL, which in turn is sampled for several consecutive clock cycles. The major cons of type B are longer dead time, more complicated decoding network, extra time jitter sources, and calibration complexity. Thus, we have chosen type A as starting point for our implementation.

In the scheme proposed by Wu [20], each HIT input generates a packet of two rising edges and one falling edge propagating through the TDL. This method aims at splitting ultrawide bins; however, also small bins get split, with the extra benefit of having more data granularity for requantization, hence an improved DNL. Theoretically, an increased number of edges (falling, rising, or both) allow further bin splitting. However, increasing the number of edges in the packet generates decoding challenges, as code bubbles (better described later) make it difficult to reconstruct the position of two edges that are too close together. In a limit case, if two consecutive edges reach each other, they will disappear. It should also be noted that all edges must be still in the chain when the sample is taken. For the above reasons, edges cannot be placed too close, but the edge packet cannot be too long either. A longer packet will result in a shorter effective TDL length before parts of the packet start dropping out of the end of the chain; and, as already seen, the chain length within the same FPGA clock region is limited. Given all these limitations, instead of increasing the number of edges, we opted to have only two edges propagating in a TDL and to use four delay lines in each channel. A pulse is fed to all four chains each time HIT is set, and all four chains are sampled at the same time.

Fig. 6 shows the implementation of the 20 channels, each one made up of four TDLs with 50 CARRY4 blocks (total of 200 taps). The origin coordinates of each channel must be manually specified to ensure that each channel starts at the boundary of a clock region, and custom scripts lock the placement of CARRY4s and sampling flip-flops (FFs) around each channel's origin, as shown by the orange shaded elements, giving a repeatable TDL placement for each channel.

The main issue in using parallel chains instead of a longer TDL capable of accommodating more edges is that not all chains start at the same precise time: e.g., sampling may happen when chain A has already started propagating, but chain B is still idle. To mitigate this issue, the HIT input is sampled by a D FF with clock enable and asynchronous clear (FDCE) and then distributed to all TDLs using a horizontal clock region

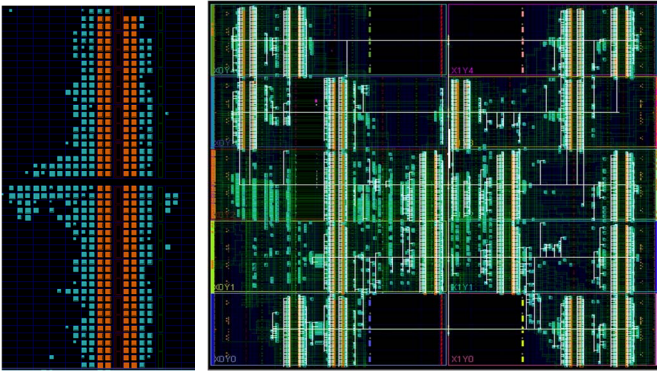


Fig. 6. Implementation of one TDL (left) and of the 20 position-locked TDLs (right). Each orange square is a slice containing a CARRY4.

buffer (BUFH), which can drastically reduce the skew between signals. Moreover, our implementation employs a validator that signals if the HIT input is propagating in all the four parallel chains. If an HIT input starts propagating close to the successive clock pulse, the validator will give the valid flag in the next clock cycle pulse and not in the current one. Thus, the tapped delay must last more than the clock period.

A decoding network is then used at the output of the four TDLs to locate the positions (i.e., the tap numbers) of the “1-0” and “0-1” changes along the TDL since the location of these “edges” is proportional to the amount of time between the HIT input itself and the TDL being sampled.

Ideally, with the edge packet used in our implementation, the code recorded at the TDL outputs would look like ...111100...0001111... (where an edge propagates from left to right). However, in actual implementation, “bubbles” in the TDL outputs, such as those underlined in the followings, are frequently observed:

```

...11111101010000000001000111111...
...1111110100000000000101111111...
...1111110010000000000111111111...
...1111110000000000000110111111...

```

Possible reasons for these bubbles are nonuniformities in the routing delays from CARRY4 outputs to the corresponding sampling FFs or possibly hidden optimizations of the carry network compared to the simple datasheet functional diagram.

In our Artix-7 implementation, we observed that the bubble patterns differ in case of either falling or rising edges and, in some channels, they are nonstatic, thus impairing the feasibility of simply reordering the TDL taps.

The decoder presented in this work solves the bubble issue without discarding information and without relying on any bubble suppression or mitigation strategy at design time. First, a boundary between “rising edge bubbles” and “falling edge bubbles” is established by locating a sufficiently long string of uninterrupted zeros in the TDL. The decoder then counts the number of ones present between the first TDL tap up to this boundary position, so to dynamically rearrange the position of 1s and 0s in the sequence, thus generating a clean thermometric code with the position of the first edge. Lastly, the decoder counts the overall number of 0s in the TDL,

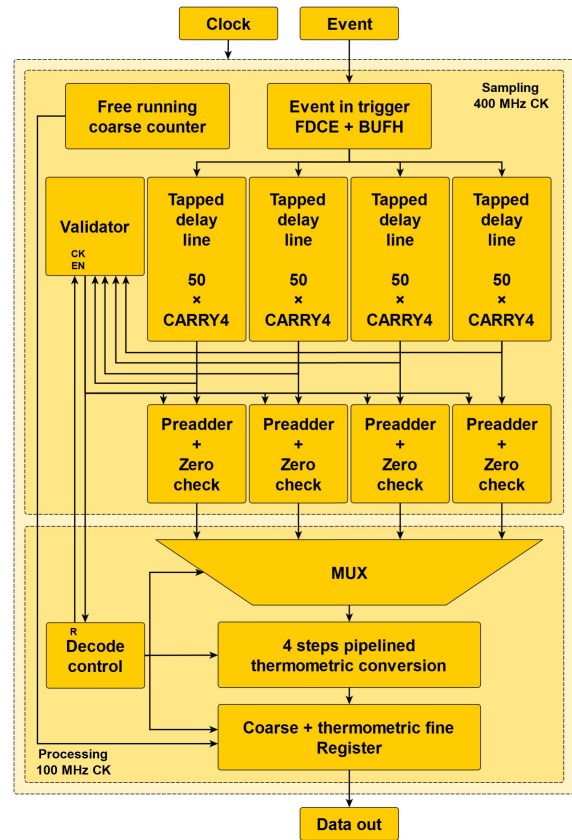


Fig. 7. Block diagram of a single TDC channel with four TDLs in parallel, a coarse counter, and the sampling logic.

corresponding to the distance between the two edges (again, after dynamically rearranging the sequence to obtain a bubble-free code). The position of the second edge is simply obtained by summing the number of 0s (distance) with the previously described first edge position.

A block diagram of the time-stamping channel is shown in Fig. 7. Most of the decoding logic is shared between all TDLs. This was done as a resource saving measure, but it represents the primary tradeoff with maximum event rate: the decode must process the four TDLs sequentially, and it also operates at a slower 100-MHz processing clock rate due to its relative size and complexity.

The TDL outputs are first sampled by a cascade of two D-type FFs (2DFF) to reduce metastability events at the inputs of the downstream circuitry. This is performed at every cycle of the sampling clock, irrespective of the HIT input.

Concurrently, the validator examines the 2DFF outputs to check whether the edge packet has moved from its starting position, which implies an HIT input. If such an HIT input is found, the sampling process is stopped to store TDL data for decoding.

The most resource-efficient way to store data would have been to act on the clock enable signal of the 2DFFs themselves. However, this was not feasible at 400 MHz due to the large number of elements and physical distance among them and ultimately required the insertion of an additional storage latch for each TDL. The first step of processing

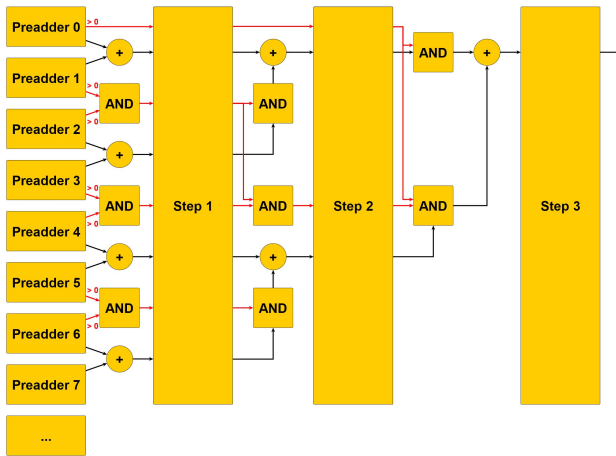


Fig. 8. Block diagram of the pipelined gated adder tree where the control signals are highlighted in red.

(the “preadders”) was then merged into each channel’s storage latch, both to reduce the number of FFs required and to make use of their related lookup tables (LUTs). Each preadder acts on groups of five consecutive TDL taps and generates a 3-bit sum of the inputs and a flag that indicates whether there is at least a zero among the inputs.

The group size of five was chosen as it efficiently uses the Xilinx 7-series slice structure, in which each LUT can generate two functions of five inputs and is associated with two matching FFs.

Following the preadders and storage stage, data move to the 100-MHz processing clock domain. After the validator circuit has identified a valid sampled event, a four-input multiplexer is used to sequentially select the outputs from the four TDLs in the channel at each cycle of the processing clock, so that the pipelined thermometric decoder receives output codes of the TDLs in four consecutive clock cycles. After data from all four TDLs have been sent to the decoder, the validator circuit is reset and the channel is ready for a new event.

The first decoding step is a subranging operation. The “contains a zero” preadder flags are inspected, and only a portion of the entire TDL is extracted, which contains the rising and falling edges. Our implementation extracts a subset of 14 from the overall 40 preadders needed to cover the full TDL, significantly reducing the size of the following adders. The selected portion of the TDL is then processed by two pipelined adder trees. One is a simple binary adder tree that computes the total number of “1s” within the selected portion: this information is then used for calculating the total number of “0s” (i.e., distance between rising and falling edges) as the input width of the adder tree is known. The second tree is a gated adder tree and is shown diagrammatically in Fig. 8. The purpose of this circuit is to calculate the number of “1s” present from the starting position (preadder 0) until the previously mentioned “sufficiently long string of uninterrupted zeros.”

In our characterization, the longest string between bubbly bits was three positions long; for this reason, we chose to look at preadder outputs and stop the summing when we reach

a preadder sum equal to zero (i.e., five consecutive zeros in the TDL). This approach to optimizing decode results in conflicting constraints on the distance of rising and falling edges in the starting packet. The size of the subset extracted by the subranging operation needs to be wide enough to always capture the full rising and falling edges as well as their bubbles; therefore, an edge packet with more spaced edges will require a wider subrange and will have less benefit in terms of resource reduction. Conversely, since the preadders operate at a step of five taps, one must ensure that there are always at least $(5 \times 2 - 1)$ consecutive zeros in the sampled data; otherwise, there is the possibility of the zeros being distributed in such a way that no preadder has a sum equal to zero and the decode will fail. This requires an initial characterization step to evaluate the “raw” TDL behavior in order to choose a safe distance between the edges and subranging width. It should also be noted that the first adder stage of this gated tree is actually identical to a normal adder tree and is in fact shared with the nongated tree. Moreover, the greater than zero comparison and AND gates used in the first stage are simple functions of six inputs and can be implemented with a single LUT6 for each pair of preadders.

At the output of the adder trees, the position of the various edges is easily reconstructed: the second (falling) edge in the TDL is given by the position of the first (rising) edge plus distance (number of zeros); the position of the first edge is the result of the gated adder tree plus number of taps skipped by the subranging operation, which is in general different for each TDL being decoded. Lastly, the value of the virtual tap for the channel is simply calculated as the sum of the eight edges across the four TDLs.

The calibration is executed by a LabVIEW software on the cumulative result from the four TDLs, giving better overall calibration compared to individually calibrating the eight edges’ positions and then averaging. Finally, the time information from both the counter and the TDL is packed together. Concurrently to acquisitions, the LabVIEW software takes advantage of the measurements to calibrate the virtual TDLs. This is possible since external events are uncorrelated to the internal sampling clock, so they can be assumed to be uniformly distributed over the sampling clock period. Each fast calibration is performed on 38 000 samples: longer calibrations provide more accurate results but require either higher event rates or longer intervals between calibrations.

III. TDC CHARACTERIZATION

The 20-channel TDC has been evaluated employing a custom board (Fig. 9) connected to an off-the-shelf XEM7310-A200 board by Opal Kelly, which hosts a Xilinx Artix-7 A200T FPGA speed grade 1. The custom board hosts a low-jitter crystal oscillator (Si570 by Silicon Lab) that generates a 300-MHz clock with 2-ps rms period jitter. This signal is fed to a phase-locked loop (PLL) inside the FPGA to obtain the TDLs 400-MHz clock. Measurements are uploaded to a remote computer through a USB link. The signal sources feed 20 MCX connectors and comparators, with adjustable thresholds (0–2.5-V range), with 0–5-V dynamics and minimum 10-ns pulsewidth before reaching the FPGA.

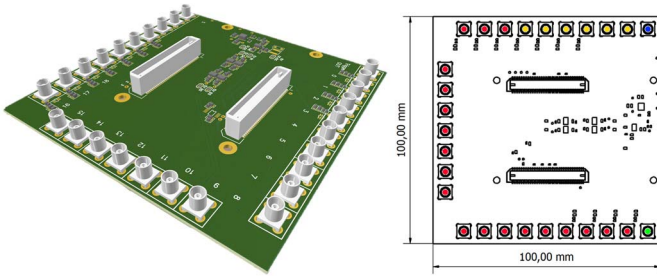


Fig. 9. Custom PCB (size 100 mm \times 100 mm) with connectors for driving the 19 START channels (red pins), the STOP one (blue pin), plus six more channels for debugging purposes (yellow pins), and an output trigger (green pin) for synchronizing the system with external circuitries (e.g., a pulsed laser).

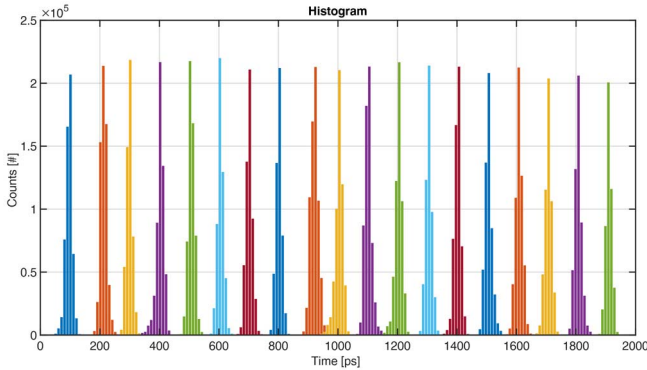


Fig. 10. Acquisitions the 19 channels, showing 40-ps FWHM precision.

We tested the single-shot precision by feeding a fixed START–STOP delay, as shown in Fig. 10. Since the sources have a jitter lower than 10 ps and the input buffers have a jitter of around 1 ps, the most significant contribution to the time jitter comes from the FPGA resulting in a 40-ps full-width at half-maximum (FWHM). The preadders and decoders are all pipelined and the resulting dead time is shorter than 60 ns. Even operating with a USB 3.0 link, the maximum throughput is limited by data transmission since a shortage of FPGA resources does not allow us to optimize the communication with the computer. This limits the throughput to 56 MB/s, with a maximum conversion rate of 700 kHz per channel.

We performed a code density test to assess TDCs nonlinearity [21] after calibration. To provide an equal probability of the HIT input in any position of the clock period, the random START events were generated by a SPAD module, providing a digital pulse for every detected photon originated by a constant light source, while the STOP was generated by the PLL inside the FPGA, in spread spectrum mode, fed by a fixed frequency clock. Fig. 11 shows the distributions of 2×10^9 time interval measurements: the computed DNL and integral nonlinearity (INL) are 1.04 ps (rms) and 58 ps (peak-to-peak value), respectively, equal to 0.1 and 5.8 LSB. Being the entire architecture open loop, the chains' time delays vary with temperature and power supply. However, the proposed TDC shows good stability performance, as TDL calibration is constantly performed in the background using

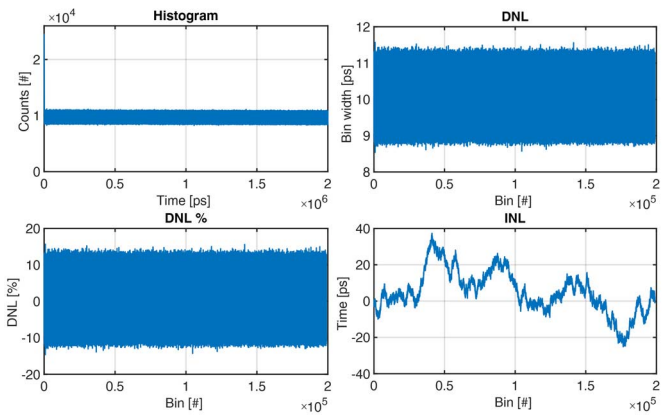


Fig. 11. Code density test histogram (top-left), bin width in picoseconds (top-right), percent DNL (bottom-left), and INL (bottom-right) of one representative channel of the TDC; the LSB is 10 ps.

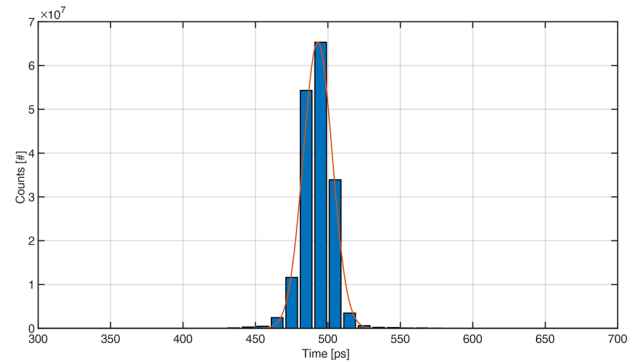


Fig. 12. 24-h stability of channel 0 (2000 cps).

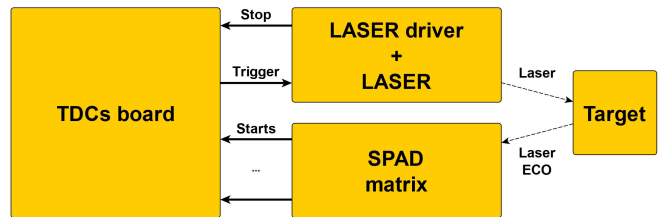


Fig. 13. Block representation of the setup employed to validate the proposed multi-channel TDC in a TCSPC application.

the live measurement data. The calibration is updated every 40 000 events to compensate for slow drifts. Fig. 12 shows a stability measurement carried out feeding two signals with a constant time delay to two timestamp units and computing the time difference between the two. The data have been acquired for 24 h with 2000 events per second and show how the calibration compensates for temperature and supplies drifts ensuring a constant FWHM of 30 ps.

IV. EXPERIMENTAL VALIDATION

In a typical time-correlated single photon counting (TCSPC) setup, employing many detectors (e.g., SPADs) and one common reference source (e.g., the laser excitation trigger), the

TABLE I
COMPARISON OF THE MAIN PERFORMANCE PARAMETERS AMONG MULTI-CHANNEL FPGA-BASED TDCS PRESENTED IN LITERATURE

	[22]	[23]	[12]	[24]	[24]	[25]	This work
Number of TDCs	48	9	161****	96	96	NS	19
INL (LSB)	NS	[-9/+8]	[-2.25/+1.16]**	[-0.09/0.11]**	[-0.15/+0.48]**	[-2.750/+1.238]**	[-2.26/+3.54]
DNL (LSB)	NS	[-1/+1.1]	[-1/+1.5]**	[-0.05/+0.08]**	[-0.12/+0.11]**	[-0.953/+1.185]**	[-0.13/+0.15]
RMS (ps)	9 *	9 ***	19.6*****	14.59 **	7.80 **	26.04 **	17
Conversion rate (per channel)	NS	NS	300 MSample/s expected **	NS	NS	125 MSample/s **	700 kSample/s
LSB (ps)	6 *	12 ***	10 **	10.54 **	5 **	22.2 **	10
FPGA	Virtex-4	Virtex-4	Virtex-6	Virtex-7	UltraScale	Artix-7	Artix-7
Resources	NS	2081 slice register + 3280 LUTs**	NS	1145 LUTs + 1916 FFs ** 55790 LUTs+91968 FFs	703 LUTs + 1195 FFs ** 68357 LUTs+114761FFs	216 LUTs + 638 FFs**	21591 LUTs + 51609 FFs

* performance of 32 channels
 ** single-TDC implementation
 *** 4-time averaging
 **** placed and tested only one by one
 ***** at 40 ns time difference

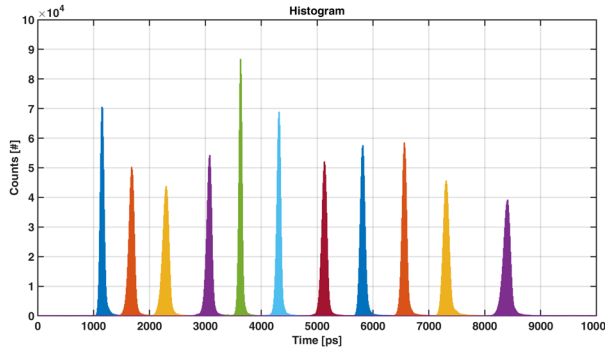


Fig. 14. Histograms with the time-of-flights recorded by a 5×5 SPAD array with a 45-ps FWHM pulsed laser. Note that peaks are shifted in time by software, just to ease readability. Each peak has an 80-ps FWHM.

multi-channel TDC can sample 19 different START (detectors) signals and one global STOP (sync laser) reference. That way, the TDC measures the 19 time intervals between each START pulse and the global STOP. We validated the 20 channel TDC in a real application, together with a SPAD array with 5×5 pixels, which provides 25 independent low-jitter digital pulses signaling the detection of a photon by the corresponding pixel; 19 of these outputs have been fed to the TDCs while the delayed laser sync was used as STOP.

The experimental setup is schematically represented in Fig. 13. The system provides the time-of-flight of photons detected by 19 pixels, which can be accumulated into independent TCSPC histograms for applications like light detection and ranging (LiDAR) [3].

Fig. 14 shows the measurements when the SPAD array is illuminated by a 45-ps FWHM pulsed laser. Note that peaks are shifted in time by software to ease readability. All channels show the expected behavior with an FWHM of 80 ps that reflects the jitter contributions of the laser (45-ps FWHM), the SPADs (60-ps FWHM), and TDCs (40-ps FWHM).

V. CONCLUSION

We have presented a low-cost FPGA implementation of a 19 channel TDC for multiple-sources time-of-flight and time-correlated photon counting applications.

The TDC is composed of 20 independent timestamp units that have been employed as 19 START input channels plus one global STOP reference. The TDC architecture is derived from a carry chain delay line model whose DNL and INL have been improved by exploiting multiple parallel carry chains (four per channel) and recording the position of both rising and falling edges (wave union A method discussed in Section II). This technique effectively subdivides ultrawide bins so to improve measurements precision, both in terms of

maximum bin width and resolution. Combined with a continuous code density calibration and bin merging, this improves TDC linearity, reaching a DNL and INL, respectively, of 1.04 and 58 ps peak-to-peak, equal to 0.104 and 5.8 LSB. The obtained LSB is equal to 10 ps with a single-shot precision of 40-ps FWHM.

Table I provides a comparison among multi-channel FPGA-based TDCs presented in the literature. The architecture presented in this article features high resolution and low non-linearity while keeping the resource usage contained. Note that some implementations do not actually employ the maximum number of channels for system testing and characterization. Moreover, it is important to highlight that the performance of FPGA-based TDCs strongly depends on the FPGA fabrication technology and the number of channels on the FPGA available hardware resources. Thus, the achieved performance should be compared with other low-cost implementations, such as the one presented in [25], which reaches lower resolution and precision and a higher DNL.

The lower conversion rate of our architecture is due to a limit in the data transfer set by the software interface. Indeed, the developed software is still a preliminary version being the aim of this work the validation of our multi-channel design and the novel decoder. We estimate that, by software optimization, 17 MSample/s per channel can be reached.

Considering that the current research trend is toward an increased number of TDCs per measurement system [26], we believe that the presented decoding scheme, through subranging, can reduce the hardware resources required by multi-channel architectures. Indeed, encoding generally consumes significant resources reducing the space for TDLs implementation.

The presented design achieves excellent delay uniformity and stability, ensuring the consistency of delay and the correctness of timing among all channels. The conversion range depends on a coarse clock counter that has been implemented using a 16-bit counter reaching an FSR of about 164 μ s. This value can be easily modified by increasing the bit depth of the coarse counters, with a minor impact on FPGA's resources usage.

The design has been exploited together with a 5×5 SPAD matrix in a TCSPC setup. The necessary calibration and digital postprocessing functions are integrated into the software to provide a plug-and-play device. However, we aim to move the calibration directly into the FPGA since this will allow further processing (e.g., computing the arrival time histograms) before data transmission, drastically decreasing the amount of data transferred to the computer and increasing the conversion rate.

REFERENCES

- [1] A. Alici, "Particle identification with the ALICE time-of-flight detector at the LHC," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 766, pp. 288–291, Dec. 2014.
- [2] H. Seo *et al.*, "Direct TOF scanning LiDAR sensor with two-step multievent histogramming TDC and embedded interference filter," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1022–1035, Apr. 2021.
- [3] F. Villa, F. Severini, F. Madonini, and F. Zappa, "SPADs and SiPMs arrays for long-range high-speed light detection and ranging (LiDAR)," *Sensors*, vol. 21, no. 11, p. 3839, Jun. 2021.
- [4] E. Venialgo *et al.*, "Toward a full-flexible and fast-prototyping TOF-PET block detector based on TDC-on-FPGA," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 3, no. 5, pp. 538–548, Sep. 2019.
- [5] F. Madonini and F. Villa, "Single photon avalanche diode arrays for time-resolved Raman spectroscopy," *Sensors*, vol. 21, no. 13, p. 2487, 2021.
- [6] Y. Seo, J. Kim, H. Park, and J. Sim, "A 0.63 ps resolution, 11b pipeline TDC in 0.13 μ m CMOS," in *Symp. VLSI Circuits-Dig. Tech. Papers*, Kyoto, Japan, Jun. 2011, pp. 152–153.
- [7] K. Hari Prasad, V. B. Chandratre, and M. Sukhwani, "A versatile multi-hit, multi-channel Vernier time-to-digital converter ASIC," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 990, Feb. 2021, Art. no. 164997.
- [8] A. Muthuramalingam, S. Himavathi, and E. Srinivasan, "Neural network implementation using FPGA: Issues and application," *Int. J. Electr. Comput. Energetic Electron. Commun. Eng.*, vol. 2, pp. 2802–2808, Mar. 2008.
- [9] F. Aubépart and N. Franceschini, "Bio-inspired optic flow sensors based on FPGA: Application to micro-air-vehicles," *Microprocess. Microsyst.*, vol. 31, no. 6, pp. 408–419, 2007.
- [10] F. Chekired, C. Larbes, and A. Mellit, "Comparative study between two intelligent MPPT-controllers implemented on FPGA: Application for photovoltaic systems," *Int. J. Sustain. Energy*, vol. 33, no. 3, pp. 483–499, May 2014.
- [11] J. Song, Q. An, and S. Liu, "A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 1, pp. 236–241, Feb. 2006.
- [12] M. W. Fishburn, L. H. Menninga, C. Favi, and E. Charbon, "A 19.6 ps, FPGA-based TDC with multiple channels for open source applications," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 3, pp. 2203–2208, Jun. 2013.
- [13] J. Wu and Z. Shi, "The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay," in *Proc. IEEE Nucl. Sci. Symp. Conf. Rec.*, Oct. 2008, pp. 3440–3446.
- [14] Q. Shen *et al.*, "A 1.7 ps equivalent bin size and 4.2 ps RMS FPGA TDC based on multichain measurements averaging method," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 3, pp. 947–954, Jun. 2015.
- [15] P. Kwiatkowski and R. Szplet, "Efficient implementation of multiple time coding lines-based TDC in an FPGA device," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7353–7364, Oct. 2020.
- [16] C. Cottini, E. Gatti, and V. Svelto, "A new method for analog to digital conversion," *Nucl. Instrum. Methods*, vol. 24, pp. 241–242, Jul./Nov. 1963.
- [17] L.-Y. Hsu and J.-L. Huang, "A multi-channel FPGA-based time-to-digital converter," in *Proc. IEEE 21st Int. Mixed-Signal Test. Workshop (IMSTW)*, Jul. 2016, pp. 1–4.
- [18] S. Kim, W. Kim, M. Song, J. Kim, T. Kim, and H. Park, "15.5 A 0.6 V 1.17ps PVT-tolerant and synthesizable time-to-digital converter using stochastic phase interpolation with $16 \times$ spatial redundancy in 14 nm FinFET technology," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2015, pp. 1–3.
- [19] W. Xie, H. Chen, Z. Zang, and D. D.-U. Li, "Multi-channel high-linearity time-to-digital converters in 20 nm and 28 nm FPGAs for LiDAR applications," in *Proc. 6th Int. Conf. Event-Based Control, Commun., Signal Process. (EBCCSP)*, Sep. 2020, pp. 1–4.
- [20] J. Wu, "Several key issues on implementing delay line based TDCs using FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 3, pp. 1543–1548, Jun. 2010.
- [21] R. Pelka, J. Kalisz, and R. Szplet, "Nonlinearity correction of the integrated time-to-digital converter with direct coding," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 2, pp. 449–453, Apr. 1997.
- [22] E. Bayer and M. Traxler, "A high-resolution (<10 ps RMS) 48-channel time-to-digital converter (TDC) implemented in a field programmable gate array (FPGA)," *IEEE Trans. Nucl. Sci.*, vol. 58, no. 4, pp. 1547–1552, Aug. 2011.
- [23] J. Wang, S. Liu, L. Zhao, X. Hu, and Q. An, "The 10-ps multitime measurements averaging TDC implemented in an FPGA," *IEEE Trans. Nucl. Sci.*, vol. 58, no. 4, pp. 2011–2018, Aug. 2011.
- [24] H. Chen and D. D.-U. Li, "Multichannel, low nonlinearity time-to-digital converters based on 20 and 28 nm FPGAs," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3265–3274, Apr. 2019.

- [25] M. Parsakordasiabi, I. Vornicu, Á. Rodríguez-Vázquez, and R. Carmona-Galán, "A low-resources TDC for multi-channel direct ToF readout based on a 28-nm FPGA," *Sensors*, vol. 21, no. 1, p. 308, Jan. 2021.
- [26] R. Machado, J. Cabral, and F. S. Alves, "Recent developments and challenges in FPGA-based time-to-digital converters," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 11, pp. 4205–4221, Nov. 2019.

Davide Portaluppi received the M.Sc. degree in electronics engineering and the Ph.D. degree in information technology both from Politecnico di Milan, Milan, Italy, in 2014 and 2019, respectively.

He has worked on nuclear instrumentation and radiation detectors and focused his Ph.D. activity on the design of single-photon CMOS imaging arrays and time-measurement circuits. He joined Analog Devices, Inc., Wilmington, MA, USA, in 2019, and his current research interests include advanced architectures for time-of-flight imaging.

Klaus Pasquinelli was born in Seriate, Italy, in 1994. He received the bachelor's and M.Sc. degrees in electronics engineering from Politecnico di Milan, Milan, Italy, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in information technology.

His research interests include the design, development, and testing of systems with single-photon avalanche diodes (SPADs) matrices and digital silicon photomultiplier (dSiPM).

Iris Cusini was born in Tirano, Italy, in 1994. She received the B.Sc. degree (*summa cum laude*) in automation engineering from Politecnico di Milan, Milan, Italy, and from Tongji University, Shanghai, China, within a double degree project, and the M.Sc. degree (*summa cum laude*) in electronics engineering from Politecnico di Milan, in 2019, where she is currently pursuing the Ph.D. degree in information technology.

Her research interests include the design, development, and testing of systems with single-photon avalanche diodes (SPADs) and digital silicon photomultipliers (dSiPMs).

Franco Zappa (Senior Member, IEEE) was born in Milan, Italy, in 1965. He received the master's degree in electronics engineering and the Ph.D. degree from Politecnico di Milan, Milan, Italy, in 1989 and 1993, respectively.

He has been a Full Professor of electronics with Politecnico di Milan, since 2011. His research interests include microelectronic circuitry for single-photon detectors (SPAD) and CMOS and BCD SPAD imagers, for high-sensitivity time-resolved optical measurements, 2-D imaging, and 3-D depth ranging via single-photons' time-of-flight. He is a coauthor of more than 250 papers, published in peer-reviewed journals and in conference proceedings, and eight textbooks on Electronic Design, Electronic Systems, and Microcontrollers. In 2004, he cofounded "Micro Photon Devices" focused on the production of SPAD modules and cameras for single photon-counting and photon-timing.