IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 22, 2019, accepted August 1, 2019, date of publication August 9, 2019, date of current version August 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933496

All-Digital Time-to-Digital Converter Design Methodology Based on Structured Data Paths

RUI MACHADO [1], JORGE CABRAL [1], AND FILIPE SERRA ALVES [2]

[1] Algoritmi Centre, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
[2] International Iberian Nanotechnology Laboratory, 4715-330 Braga, Portugal

Corresponding author: Rui Machado (id6010@alunos.uminho.pt)

This work was supported in part by the Portuguese Scholarship from Fundação para a Ciência e Tecnologia (FCT) and in part by the Bosch Car Multimedia through the Advanced Engineering Systems for Industry (AESI) Doctoral Program (Scholarship ID: PDE/BDE/114562/2016) and FCT within the Project Scope: UID/CEC/00319/2019.

ABSTRACT Time-to-Digital Converters (TDC) are popular circuits in many applications, where high resolution time measurements are required, for example, in Positron Emission Tomography (PET). Besides its resolution, the TDC's linearity is also an important performance indicator, therefore calibration circuits usually play an important role on TDCs architectures. This paper presents an all-digital TDC implemented using Structured Datapath to reduce the need for calibration circuitry and cells custom design, without compromising the TDC's linearity. The proposed design is fully implementable using a Hardware Description Language (HDL) and enables a complete design flow automation, reducing both development time and system's complexity. The TDC is based on a Delay Locked Loop (DLL) paired with a coarse counter to increase measurement range. The proposed architecture and the design approach have proven to be efficient in developing a high resolution TDC with high linearity. The proposed TDC was implemented in TSMC 0.18um CMOS technology process achieving a resolution of 180ps, with Differential Non-Linearity (DNL) and Integral Non-Linearity (INL) under 0.6 LSB.

INDEX TERMS Structured data path, time-to-digital converters, TDC, ASIC.

I. INTRODUCTION

Time-to-Digital Converters (TDC) are devices used to convert the difference between the arrival time of two signals to a digital value [1], [2]. TDCs have long been used in physics experiments for Time-of-Flight (TOF) measurements, and laser range finders [1]–[5]. Lately, TDCs are becoming more popular and its range of applications has been growing. In CMOS image sensors, TDCs are used in combination with Analog-to-Digital Converters (ADC) to shorten the conversion time without decreasing the dynamic range of the ADC [6]. With the appearance of LiDAR systems in the automotive industry, the high resolutions achieved by TDCs enable for higher precision in object detection and tracking [5]. Finally, time-based accelerometers also benefit from the use of a TDC, as it enables to further enhance the overall system resolution due to a more precise pull-in time measurement [7].

The main performance figures of a TDC are resolution, differential non-linearity (DNL), and integral non-linearity (INL) [6]. Recently, some research works have reported resolutions under 10ps with DNL and INL under ±1 LSB [1], [3]. Perktold and Christiansen [3], claims to be the first to achieve a single-shot precision of 3 ps-rms. The research work by Ko *et al.* [1], proposes a Vernier sub-ranging TDC with DNL calibration, achieving a 5 ps resolution. A more recent work [2] presents a hybrid architecture TDC based on residual time extraction and amplification. According to the results reported, high linearity was achieved, although the 320 ps resolution is far from the current state of the art.

Modern applications, like the use of LiDAR in the automotive industry, demand for readout systems capable of achieving both high resolution and long measurement range. As the number of sensors and sub-systems in a car increases, small die size and low power consumption are usually desired in automotive applications as well. With the ever-increasing market competition, short development cycles are also an issue to consider when designing a new system. The current CMOS TDC architectures, although achieving high

The associate editor coordinating the review of this manuscript and approving it for publication was Gian Domenico Licciardo.

resolutions and linearity, usually require custom made cells or blocks to calibrate the system to compensate for power, voltage, and temperature (PVT) variations. Therefore, although some architectures are fully digital, the design of transistor-level specific cells is required. This greatly increases the development cycle when compared to a fully digital Verilog or VHDL implementation.

This paper proposes an all-digital TDC architecture based on a Delay Locked Loop (DLL) that takes advantage of the computer aided design (CAD) tools Structured Data Path feature, implemented in Cadence Innovus, to achieve high linearity. With this approach, no extra hardware is required for the TDC calibration, improving both die size and power consumption. Moreover, as the design is fully implemented using a Hardware Description Language (HDL), the system's design and validation time is greatly reduced. In order to increase measurement range, the proposed TDC was paired with a coarse counter, implementing a two-stage measurement. The proposed TDC was implemented in the TSMC 0.18 μm CMOS process.

The remainder of this paper is structured as follows. Section II presents the current state of the art for CMOS TDCs. In Section III, the proposed architecture and design process are explained, together with a description of the major blocks that compose the proposed TDC. The results achieved are presented in Section IV. Finally, Section V provides the main conclusions and future research paths.

II. STATE-OF-ART

Several architectures have been proposed to address the increasing time measurement resolution and linearity requirements of TDCs. While in the beginning, high resolution TDC systems were only possible using Application-Specific Integrated Circuits (ASIC), with the evolution of Field-Programmable Gate Arrays (FPGA) technology, some architecture started to be ported to these platforms. The fast prototyping cycles enabled by FPGA implementations, results in less expensive solutions. In the following subsections, the main architectures for ASIC and FPGA implementations are summarized, together with the major results reported in the literature.

A. TIME-TO-AMPLITUDE CONVERTER (TAC) TDC

A simple implementation of a TDC can be achieved by pairing a Time-to-Amplitude Converter (TAC) and an ADC. In this architecture, a capacitor is charged by a current source during the time interval (Ti), resulting in a capacitor charge proportional to Ti. This value is then converted to a digital value by the ADC. Although simple to implement, this architecture has a large deadtime and the final system resolution is limited by the ADC resolution. Due to its analogue nature, this architecture is not implementable in FPGA platforms.

The work from Cossio [8] reported a 50 ps resolution using this architecture with a power consumption of 10 mW per channel and a deadtime under 17μs.

B. TAPPED-DELAY LINE (TDL) TDC

The TDL architecture uses the intrinsic propagation delay of a digital cell to create multiple interpolation stages. By sampling a signal that has been delayed by a chain of multiple digital cells with similar propagation delays, it is possible to obtain a so-called thermometer code which has information regarding the time difference between the sampling and the delayed signal. The thermometer code is then decoded to a binary format. As the cells propagation delays are affected by PVT conditions, and the TDC linearity deteriorates as the delay chain length increases, it is usual to implement this architecture in a Delay-Locked Loop (DLL) or Phase-Locked Loop (PLL) schema in ASIC platforms [9], [10]. In this configuration, the delay chain is built as a Voltage Controlled Oscillator (VCO) with a fixed frequency, controlled within the PLL. To reduce the size of the chain, a digital counter, clocked by the VCO, is also implemented. When a sampling signal arrives, the state of the delay chain, together with the VCO clocked counter, have the time measurement information. Since it is not possible to implement variable delay cells in FPGA, this architecture is often used together with a calibration mechanism, most commonly decimation [11], [12] or histogram-based calibration [13], [14], to compensate for the PVT conditions and increase system linearity. Nevertheless, TDL architectures are by far the most used for both ASIC and FPGA platforms. As the achievable resolution is closely coupled with the technology being used, resolutions in the range of 500 ps [15], [16] and 50 ps [17]–[19] are frequently reported. When this architecture is used with second-stage subgate delay techniques, resolutions of 5 ps with DNL and INL values in the range of $+/-1$ LSB can be achieved [20].

To extend the system measurement range, TDL architectures are frequently coupled with global clock counters, usually called coarse counters. In these, it is possible to have a sampling event at the same time the coarse counter is changing. To avoid metastability issues, two coarse counters are usually implemented. Each counter is incremented in different clock edges, one at rise clock edge and the other at the clock falling edge. The value to use i.e. the value that is not metastable, is chosen depending on the value presented on the delay line [21], [22].

C. VERNIER DELAY LINE (VDL) TDC

One of the problems of the TDL architecture is that the maximum achievable resolution is technology dependant and cannot go below the intrinsic cell's delay. To address this issue, Vernier Delay Lines (VDL) were proposed. This architecture delays both the start and stop signals using two different delay lines. It is therefore possible to achieve subgate delay resolution, given by the difference between the delay of the cells used in the two chains. It is important to assure that the cell's delay of the stop signal chain is smaller than the cell's delay of the start signal, otherwise the stop signal would never be able to catch-up with the start signal, making it impossible to get a valid measure. As the size of the delay chain tend to be much larger than in the TDL architecture, the VDL is usually built in a two DLL or PLL (one for each delay line) schema, paired with a loop counter. Resolutions in the range of 30 ps [9] and 20 ps [23], both with DNL and INL under +/−1 LSB, were already reported. ASIC implementations of VDL are quite challenging, as the PVT conditions are complex to calibrate between delay chains. Regarding FPGA implementations, the VDL architecture is usually implemented using a dual ring oscillator schema, paired with a phased detector. In this architecture, the frequency difference between the two oscillators is the resolution defining element, rather than the difference between cells' propagation delay. Lee and Moon [24] were able to achieve a 1.45 ps resolution using a 0.18 μm CMOS process at 1.8 supply voltage, by combining a VDL architecture with a single time amplifier.

D. PULSE SHRINKING TDC

Pulse shrinking architectures' principle of operation is based on the mismatch between the rising and falling transition time of a cell. Using this mismatch is possible to build a delay line capable of shrinking an input pulse gradually until it ceases to exist. The shrinking factor, i.e. the amount the input pulse is shrunk each iteration, gives the TDC resolution. These architectures are implemented using a delay line configured in a loop with a counter clocked by the last delay stage of the loop. When the pulse starts propagating in the delay line loop, it is shrunk at each loop cycle, until it reaches a point in which the oscillation in the loop extinguishes, stopping the clock of the loop counter. The number of loop iterations, times the shrinking factor, gives the length of the input pulse. Although this architecture offers good resolution and low power consumption, it requires the custom development of a shrinking element. Resolutions close to 40 ps [10] have been reported with linearity values of +/−0.6 LSB.

As in the case of TAC TDCs, this architecture is not suitable for FPGA platforms, since the resources on FPGA are fixed and it is not possible to create custom cells, greatly limiting the control of the shrinking factor.

E. PHASED CLOCKS TDC

A popular architecture, especially for FPGA implementations, when resolution requirements are in the range of a few nanoseconds is phased clocks TDCs. In FPGA platforms, several research works have reported resolutions of 1ns using the FPGA's PLL blocks to generate multiple phased clocks, all sampling the same signal. By identifying which phased clock first sampled the input signal, or by sampling the phased clocks state at the arrival of an input signal's rise edge, it is possible to achieve sub clock frequency resolutions. In this architecture, the phase difference between clocks is the TDC's resolution defining element. This solution does not offer the best resolution, but it has high linearity, does not require a calibration block and has low hardware resources count. Recently, the work from Wang *et al.* [25] reported an ASIC implementation with a 780ps resolution on a 0.13um CMOS technology.

More recently, a novel resistive interpolation TDC was introduced by Mauricio *et al.* [5], [26], reporting a 15ps resolution with +/−0.31 LSB DNL and +/−0.68 LSB INL. The research work reports a system implemented in 0.18 μm CMOS technology, with a power consumption of 11.3 mW.

III. DESIGN OF THE ALL-DIGITAL TDC

From the analysis presented in the previous Section, it can be concluded that the available architectures offer great resolutions, high linearity and large measurement range. Although all ASIC implementations offer both high resolution and good linearity, it demands for a custom-made design process, where specific delay cells and calibration mechanisms must be implemented. This customization process leads to an increase on the development time, power consumption, and circuit area. In applications where the required resolution does not go below the intrinsic gate delay, the use of standard technology cells, like clock buffers, can be advantageous in terms of development time and design complexity. By using these standard cells, a complete HDL design flow can be adopted, taking advantage of the CAD tools functionalities for both optimization and verification.

The proposed design flow aims to create a TDC design with resolutions that can compete with the current state of the art TDCs, without the need for extra calibration circuitry or PVT compensation. The complete TDC architecture is depicted in Fig. 1. The architecture is based on a two-stage counting schema, using a coarse counter (*coarse counter 0°* in Fig. 1) for extended measurement range and a DLL for fine time measurement, below the system clock (sys clk) cycle period. The TDC circuit also includes an edge detector block, a second coarse counter, phased 180° regarding the main coarse counter, for metastability correction (*coarse counter 180°* in Fig. 1), and a merge block.

The results of a timing simulation are presented in the diagram of Fig. 2 (simulation results represented in a hexadecimal notation). The measurement process starts when a rise edge of the signal to be measured (denoted as *hit*) is detected. This event triggers the DLL of the start signal (start DLL). A simplified RTL view of the DLL is shown in Fig. 3. A logic *'1'* starts propagating throughout the delay line cells until it reaches the end of the delay line. At this point, the signal is inverted and used as feedback signal to the beginning of the delay line, triggering the propagation of a logic *'0'*. The DLL will maintain this oscillating behavior until the next system clock rise edge, moment in which the oscillations are stopped, and the state of the DLL, represented by a thermometer code (T0 to T31 in Fig. 3), is sampled. In the presented simulation example (Fig. 2), the hit signal had enough time to propagate until the 8*th* delay cell. On the next system clock cycle, the thermometer code is stored in a second set of registers (*Start Therm. Code* in Fig. 2) where it is kept stable for proper thermometer to binary conversion, performed by a priority decoder. Two 4-bit binary counters (see Fig. 3) are clocked by the last stage of the DLL. One incremented every time a *'1'* finishes to propagate in the

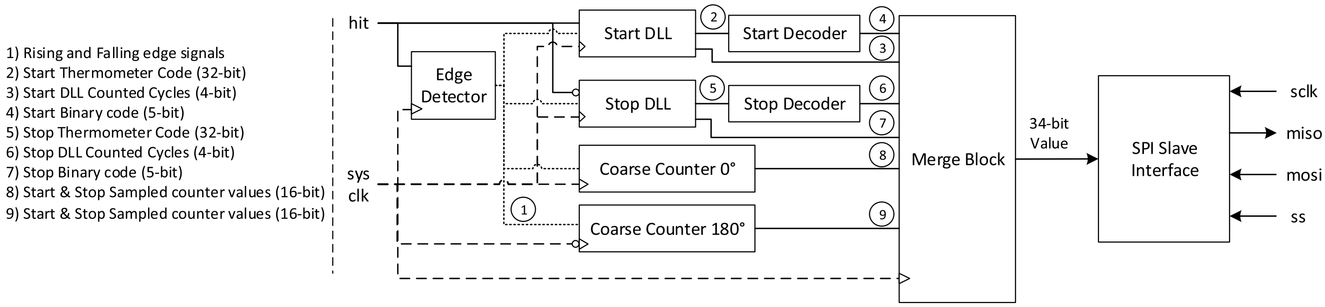

1) Rising and Falling edge signals
2) Start Thermometer Code (32-bit)
3) Start DLL Counted Cycles (4-bit)
4) Start Binary code (5-bit)
5) Stop Thermometer Code (32-bit)
6) Stop DLL Counted Cycles (4-bit)
7) Stop Binary code (5-bit)
8) Start & Stop Sampled counter values (16-bit)
9) Start & Stop Sampled counter values (16-bit)
hit
sys clk
Edge Detector
Start DLL
Stop DLL
Start Decoder
Stop Decoder
Coarse Counter 0°
Coarse Counter 180°
Merge Block
34-bit Value
SPI Slave Interface
sclk
miso
mosi
ss

FIGURE 1. ASIC block diagram.

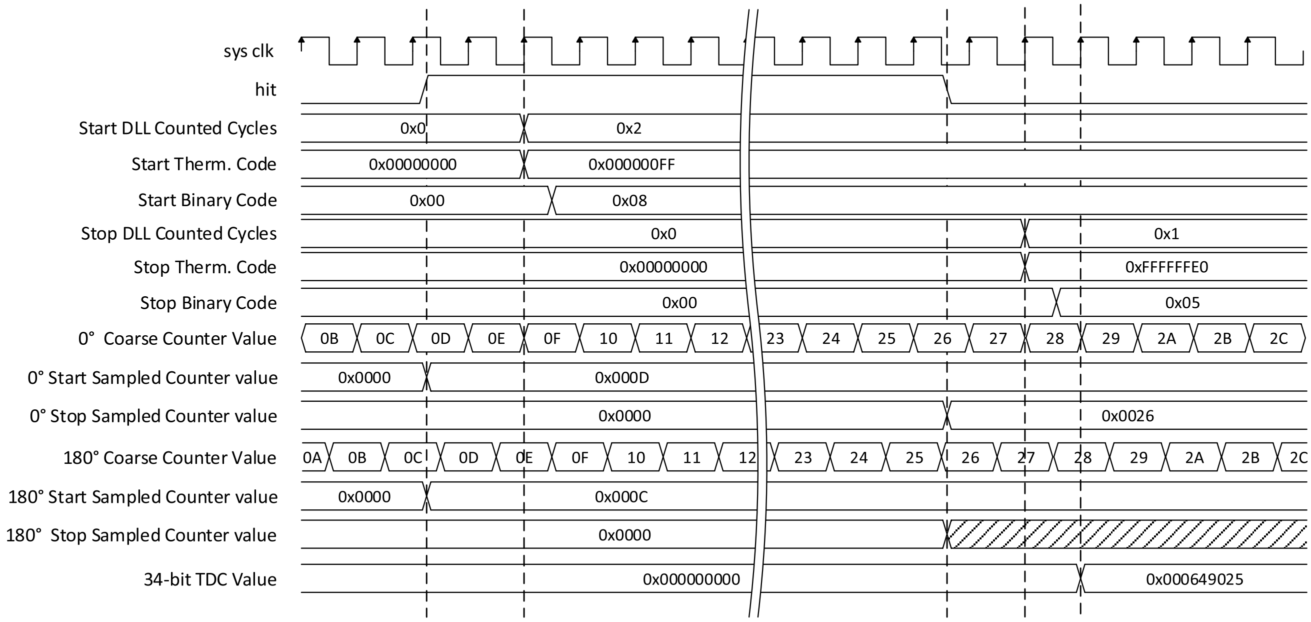


FIGURE 2. Timing simulation results (waveform diagram representation).

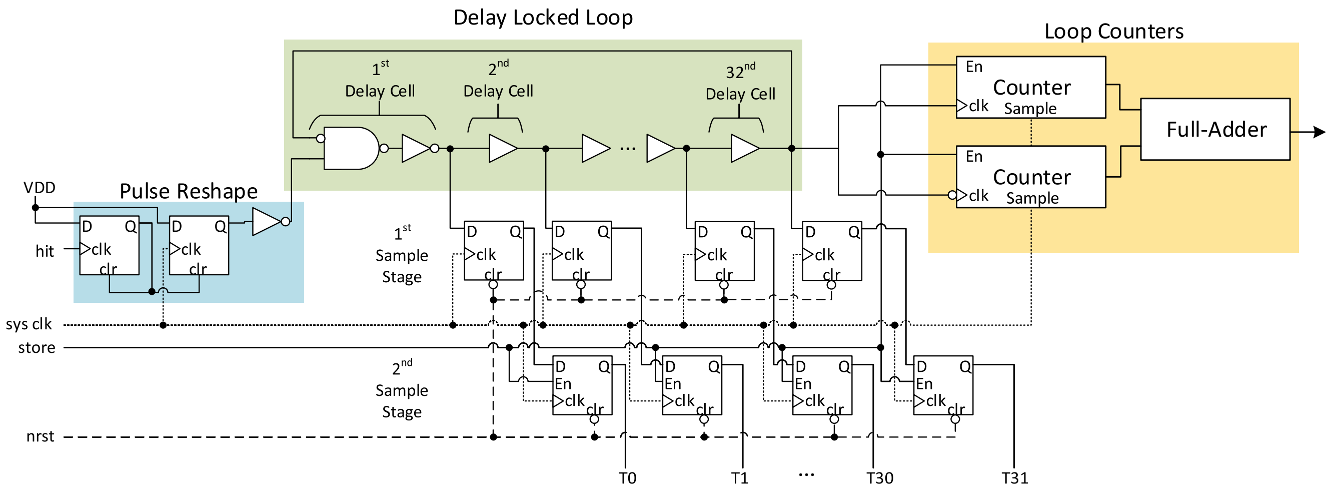


FIGURE 3. Delay locked loop architecture simplified RTL view.

33 18	17 14	13 9	8 5	4 0
Coarse Value	Start DLL Counted Cycles	Start Binary Code	Stop DLL Counted Cycles	Stop Binary Code

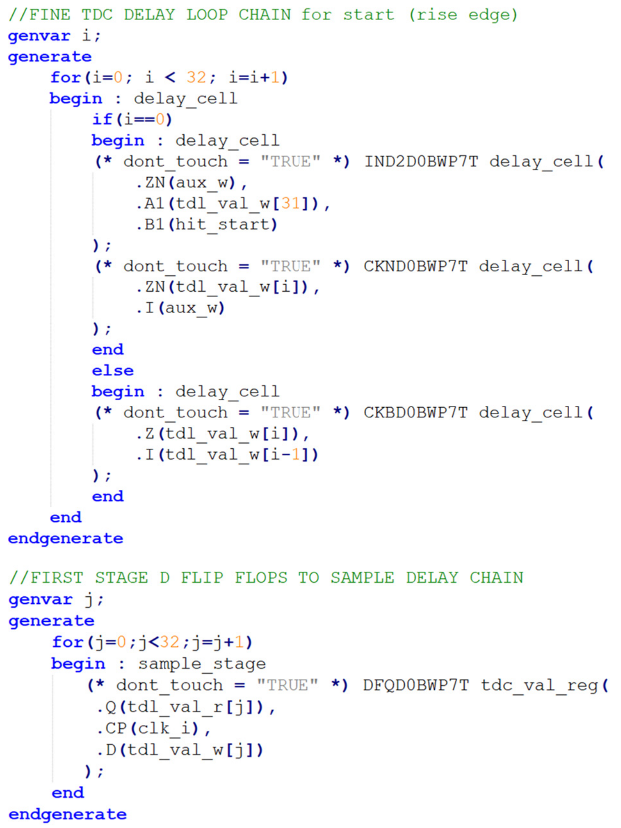
FIGURE 4. Measurement 34-bit data packet structure.

DLL (a 0-to1 transition on the last buffer of the DLL), and another one incremented when '0' finishes to propagate in the DLL (a 1-to-0 transition on the last buffer of the DLL). This loop counting mechanism increases the TDC channel flexibility since it allows a lower system clock frequency, which increases the maximum dynamic range of the system, while maintaining the same resolution. The counters' values are sampled on the first system clock cycle subsequent to the hit signal arrival. After that, the values are added, and the result is stored in a second set of registers (*Start DLL Counted Cycles* in Fig. 2). In the presented example, the sum of the two loop counters is 2, which means that the hit signal had enough time to propagate two times throughout the entire DLL. Together with the binary value of the DLL state (*Start Binary Code* in Fig. 2) provides the fine measurement value for the start event. The rise edge of the hit signal also triggers the sampling of the values presented in both coarse counters (*0° Start Sampled Counter Value* and *180° Start Sampled Counter Value* in Fig. 2). These counters are free-running counters being one incremented at every system clock rise edge and the other incremented at each system clock fall edge.

The end of the measurement is triggered by the fall edge of the hit signal. This event starts the DLL of the stop signal (stop DLL), and its value will be sampled in the next system clock rise edge and stored for decoding in the second system clock rise edge (*Stop Therm. Code* in Fig. 2). At the same time, the stop DLL loop counters are sampled and stored (*Stop DLL Counter Cycles* in Fig. 2). Once more, the final value of the loop counters and the binary value of the DLL state (*Stop binary Code* in Fig. 2) are used to get the fine measurement value for the stop event. The fall edge of the hit signal also triggers the sampling of the coarse counters (*0° Stop Sampled Counter Value* and *180° Stop Sampled Counter Value* in Fig. 2). The final measurement value is obtained on the third clock cycle after the hit's fall edge event, and it is the result of merging all the mentioned values in a single 34-bit frame (see Fig 4). The 34-bit TDC value is retrieved as follows: 1 - the stop and start coarse counter values are subtracted (in this example 0x26-0x0D) and the result is placed in the first 16 Most Significant Bits (MSB) of the output word. The process of selecting which coarse values to use between the 0° and 180° counters will be further explained in sub-section B; 2- the values from the start and stop DLLs are merged in the remaining bits in the order depicted in Fig 4. In the presented example, the final TDC value is 0x000649025.

A. TWO-STAGE TIME MEASUREMENT

The time measurement unit is composed by a coarse counter and a 32-interpolation stage DLL. The course counter is a simple binary counter, incremented at each system clock cycle and sampled by the signal to be measured (the hit signal in the architecture block diagram of Fig. 1).



```
//FINE TDC DELAY LOOP CHAIN for start (rise edge)
genvar i;
generate
    for(i=0; i < 32; i=i+1)
    begin : delay_cell
        if(i==0)
        begin : delay_cell
        (* dont_touch = "TRUE" *) IND2D0BWP7T delay_cell(
            .ZN(aux_w),
            .A1(tdl_val_w[31]),
            .B1(hit_start)
        );
        (* dont_touch = "TRUE" *) CKND0BWP7T delay_cell(
            .ZN(tdl_val_w[i]),
            .I(aux_w)
        );
        end
        else
        begin : delay_cell
        (* dont_touch = "TRUE" *) CKBD0BWP7T delay_cell(
            .Z(tdl_val_w[i]),
            .I(tdl_val_w[i-1])
        );
        end
    end
endgenerate

//FIRST STAGE D FLIP FLOPS TO SAMPLE DELAY CHAIN
genvar j;
generate
    for(j=0;j<32;j=j+1)
    begin : sample_stage
        (* dont_touch = "TRUE" *) DFQD0BWP7T tdc_val_reg(
        .Q(tdl_val_r[j]),
        .CP(clk_i),
        .D(tdl_val_w[j])
        );
    end
endgenerate
```

FIGURE 5. Verilog representation of the DLL TDC channel.

Because the hit signal is completly asynchronous to the system clock it is possible to have a scenario where the value in the counter is sampled while it is being incremented (depicted in Fig. 2, the *180° Stop Sampled Counter Value* is shadowed due to a metastable state). As it is not possible to guarantee that all the bits in the counters update simultaneously and since the hit signal can arrive inside the flip-flops setup and hold timing window, it is mandatory to implement a method to rectify these uncoherent states of the counter's value. Therefore, a second binary counter, incremented at the falling edge of the clock signal, is used.

The sub-clock time measurement, hereafter referred as fine measurement, is done by an interpolation circuit, built by connecting buffers in chain, forming a loop. Each interpolation stage output is connected to a flip-flop that samples its state at each clock cycle, and to the next delay element on the chain (see Fig. 5 for the Verilog code responsible for generating the DLL and respective sampling stage). The last delay element of the loop is used as clock on two 4-bit binary counters. One of the counters is incremented at the rising edge and the other at the falling edge. The simplified version of the RTL view of the described delay locked loop is depicted in Fig. 3. The time measurement can be obtained according to the following equations:

$$t_{fine} = (start - stop)\tau + (Cy_{start} - Cy_{stop})32\tau \tag{1}$$

$$t_{meas} = (Cnt_{stop} - Cnt_{start})T + t_{fine} \tag{2}$$

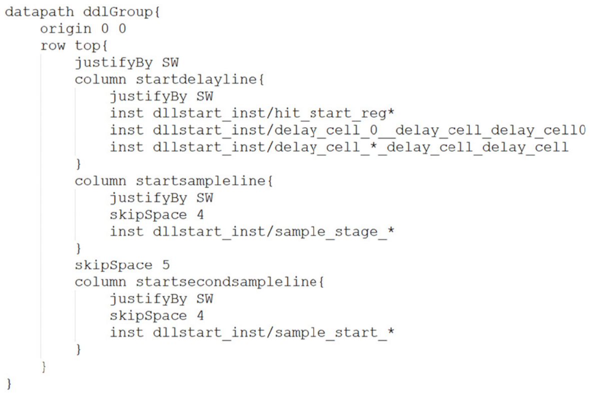
being t_{fine} the time measured by the DLL, *start* and *stop*, the binary value from the fine measurement for start and stop signals respectively (*Start Binary Code* and *Stop Binary Code* in Fig. 2), τ, the delay line basic element propagation delay, Cy_{start} and Cy_{stop} the values of the binary loop counters at the start and stop sampling event respectively (*Start DLL Counter Cycles* and *Stop DLL Counter Cycles* in Fig. 2), t_{meas}, the time interval measured by the TDC, Cnt_{start} and Cnt_{stop}, the coarse counter binary value sampled on the hit rise and fall edge respectively, and T the period of the system clock.

Each DLL outputs a thermometer code for the decoder block. The decoder has two priority encoders, one for detecting falling edge transitions, corresponding to the cycle in which the DLL is propagating a *'1'* logic level, and another for detecting rising edge transitions, for the case in which the DLL is propagating a *'0'* logic level.

The design of the DLL is the most critical part of the TDC system development, since it is the block responsible for defining the system resolution and linearity. Therefore, some aspects must be taken into consideration. The longer the delay chain, the higher it's the non-linearity, as the mismatch between gates propagation delay are adding across the entire chain (i.e. PVT conditions). Also, due to non-uniform gates' rise and fall times, pulse stretching/shrinking effects are magnified on longer delay chains.

To minimize non-uniformities on the cells' rise and fall times, clock buffers were selected as the delay element. These cells are available in all technologies design kits and are known for having a close match between rise and fall time, in order to have very limited impact on the clock's distribution along a design's clock tree. The selected gate to implement the delay chain was a clock buffer from TSMC 0.18 μm technology, with a 150 ps propagation delay, according to TSMC documentation.

As technology scales down, and with it, the gates' propagation delay time decreases, the RC parasitics of the connecting wires can no longer be ignored. These play an important role on the gates' load capacity, affecting its charge and discharge speeds, and therefore, its propagation delay. To tackle this issue and secure a uniform delay across the delay chain, the placement and routing of the delay elements must be done precisely. Even with small delay chains, manually placing and connecting all the elements is a monotonous and time-consuming task. Furthermore, a fully scripted design process is impossible to achieve. To address this issue, Structured Data Path (SDP) can be used to describe how the CAD tools should place a given group of gates. According to Cadence's Innovus user guide [27], the main advantage of using Structured Data Path capabilities is that it ensures a uniform routing, which in the case of implementing a TDC, is a much desired feature. With the gates placed in a structured manner, the CAD tools are capable of automatically perform a structured routing, that, although not identical between cells, have similar effect in terms of RC parasitics, resulting in gates with identical propagation delays. The SDP function allows the placement of a group of cells in rows and/or columns improving clock latencies, area usage and system performance. The main disadvantage of using SDP is the need for a deep knowledge on the design being implemented, although, this is a common issue in all TDC designs.



```
datapath ddlGroup{
    origin 0 0
    row top{
        justifyBy SW
        column startdelayline{
            justifyBy SW
            inst dllstart_inst/hit_start_reg*
            inst dllstart_inst/delay_cell_0__delay_cell_delay_cell0
            inst dllstart_inst/delay_cell_*_delay_cell_delay_cell
        }
        column startsampleline{
            justifyBy SW
            skipSpace 4
            inst dllstart_inst/sample_stage_*
        }
        skipSpace 5
        column startsecondsampleline{
            justifyBy SW
            skipSpace 4
            inst dllstart_inst/sample_start_*
        }
    }
}
```

FIGURE 6. SDP file example.

A Structured Datapath can be described using a SDP file, which can then be read by the CAD tool using Tool Command language (TCL) commands. Therefore, a full scripted design flow can be adopted. Part of the used SDP file is presented in Fig 6 and the resultant ASIC layout is depicted in Fig 7. In the example file, a row (named *top*) is created with 3 columns, one for the start DLL elements (*startdelayline*), and two for the first and second sampling stages (*startsampleline* and *startsecondsampleline respectively*). In each column, the name of the instances to be placed are defined using the keyword *inst*. Wildcards can be used to define multiple instances in one line. For example, the line *inst dllstat_inst/delay_cell_∗_delay_cell_delay_cell* will place all the instances that belong to module *dllstart_inst*, as well as all the cells which the name starts by *delay_cell_* and ends with *_delay_cell_delay_cell*. When multiple instances are defined in the same column, the placement is done in the following order: the first instance to be described is the first to be placed, the following instances are placed on top of the previously placed instances. If multiple instances are defined in the same line, by means of wildcards, the placement is done alphabetically, so in the aforementioned scenario, the *delay_cell_0* would be placed first, followed by *delay_cell_1* and so on. In the layout, the physical placement is done from the bottom of the floorplan to the top

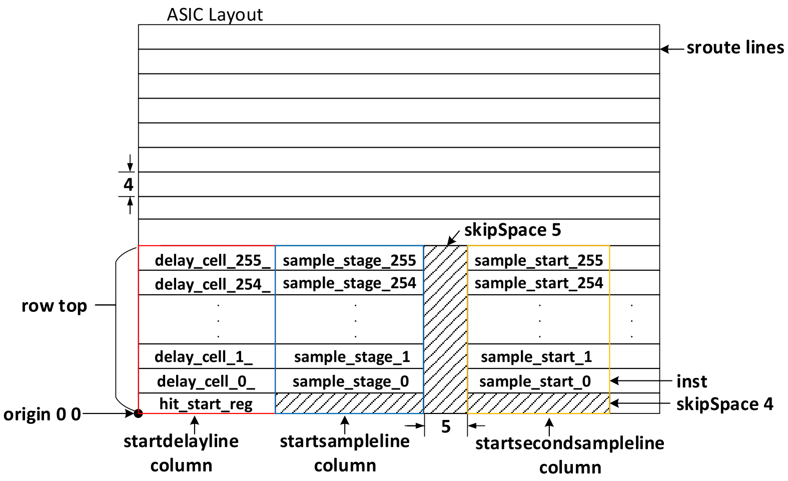


FIGURE 7. Placement result using the SDP file example.

and the width of the column is defined by the width of the largest cell placed in that column.

The influence of SDP placement and routing on the linearity of the delay chain was studied and the results will be presented and discussed in Section IV.

B. EDGE DETECTOR AND MERGE BLOCK

The edge detector block is responsible for generating control signals at every rising and falling edge event. These signals are used to enable the store stage and to sample the value on the loop counters of the DLL. The falling edge event signal is also used to identify the end of conversion.

The merge block receives all the measurement values and performs the required calculations and synchronization checks. This block is responsible for checking in which stage of the DLL the hit signal arrived. This information is used to decide which of the coarse counters has the correct value. To ensure a stable coarse counter, a large synchronization window was implemented with a time interval of 4 ns (2 ns before or 2ns after the system clock rise edge), which corresponds to approximately 20% of the system's clock period. The process used to choose the synchronization window is as follows: First, a time interval greater than the setup and hold time of the used cells must be added before and after the clock rise edge. Since we are using TSMC 180nm technology, 500 ps are more than enough. Second, because the output bits of the counter have different routing, it is needed to secure an extra time to account for skew between these nets. The higher the value selected the easier is to attain a skew in-between our synchronization window. A quick layout was made to understand the typical skew in these nets, and based on the results, it was decided that an extra 1.5 ns should be added to each side of the synchronization window. Therefore, all the DLL values sampled in the range equal to 2 ns before or 2 ns after the clock rising edge, indicate that the main coarse counter may be metastable. Following this methodology, we can be sure that if any value from the DLL of the TDC is outside the synchronization window the value on the main coarse counter is completely stable and it is safe to use it. Otherwise, the value on the second coarse counter is used since it has a phase difference of 180° and therefore is far from metastable when the values of the DLL are inside the synchronization window.

The correct thermometer decoded value is selected based on the value from the two DLL loop counters. If the loop counter has an even number, then the DLL was propagating a *'1'* and the value from the decoder responsible for detecting a 1-to-0 transition in the thermometer code should be used. When the loop counter has an odd number, then the DLL was propagating a *'0'* and the value from the decoder that detects a 0-to-1 transition is the one to be used. These values are then merged together, and a final 34-bit measurement value is created. The final 34-bit measurement value has the format presented in Fig 4: the 16 most significant bits hold the result from the subtraction of the start and stop coarse timestamps; the remaining bits have the value of the start and stop fine measurements.

C. SYSTEM INTEGRATION

The layout of the DLLs using Structured Datapath is depicted in Fig. 8-a, where the two DLLs (start and stop) are highlighted. In Fig. 8-b, the layout without using Structured Datapath is shown. It is possible to verify that, in the case were the the DLL placement was not constrained, the dispersion of the DLL's building cells is higher, resulting in different routing patterns, which ultimately leads to higher nonlinearities across each interpolation stage. The DLL blocks are less than 25% of the total circuit area. The full TDC layout including the TDC channels and some auxiliary circuits, like the Serial Peripheral Interface (SPI) protocol circuit, has a 0.17 mm × 0.17 mm total area, and was implemented using TSMC 0.18μm, six metals technology process.

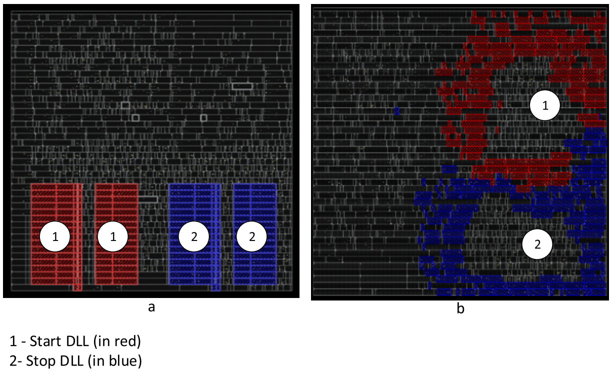


FIGURE 8. **Highlight of the TDC channels placement (a) using Structured Datapath placement (b) using default placement.**

In the next Section, the results for post-layout simulation are presented and discussed to address the effectiveness of the proposed design methodology for all-digital TDCs.

IV. RESULTS

The design of the proposed TDC architecture was entirely implemented using Verilog HDL. To synthesize the architecture, the Synopsys' Design Compiler was used, and the extracted netlist was then used as the input for Cadence's Innovus Place & Route tool, together with the SDP file responsible for constraining the TDC placement. Once finished, the layout's parasitics were extracted in the form of a Standard Delay Format (SDF) file which was used latter in Cadence's NCVerilog to perform post-layout timing simulations (Fig. 2).

To validate the proposed design method, two layouts were used. The first one was constrained by a SDP file and the second was left completely unconstrained. The post layout simulations were performed assuming a 50 MHz system clock, and a power supply voltage of 1.8 V. In such conditions, the measurement range of the TDC is 1,31 ms, corresponding to the 16-bit coarse counter implemented. The resolution of the TDC is defined by the propagation delay of the cells used in the Delay Locked Loop, which, in the voltage conditions aforementioned and at ambient temperature, is approximately equal to 180 ps in the worst-case conditions (30 ps more than the documented value).

To validate system's operation, post-layout simulations were performed using the SDF files with the design extracted timing information. This simulation with annotated timings also allows the evaluation of the proposed design methodology. The layout of the synthetized design was done as follows: first, a typical place and route flow was used, with no constraints applied to the design tool regarding cells' placement; after extracting the post-layout timing information, another run was made. In this run, a SDP file with information regarding the DLL placement was used in the Innovus tool, leaving the remaining design unconstrained. Again, the post-layout timing information was recorded. The collected data was then analysed using a MATLAB script in order to calculate the DNL and INL, assuming a LSB of 180 ps, which corresponds to the DLL cell's mean propagation delay. The worst-case scenario, for start and stop DLLs, was considered for both layouts. The results are presented in Fig. 9.

Ideally, the time between two consecutive delay line steps should be equal. However, due to changes in PVT conditions, these values change. The differential non-linearity is the deviation of a given cell's propagation delay from its ideal value, and can be calculated, referring to LSB, as follows:

$$DNL_i = \frac{t_{pi} - \tau}{\tau} \tag{3}$$

being t_{pi} the real propagation delay of the ith delay line cell, and τ the ideal propagation delay. The Integral non-linearity can be defined as the cumulative error across the delay line, and can be obtained by adding the delay line DNL values, as follows:

$$INL_i = \sum_{i=0}^{N-1} \frac{(t_{pi} - \tau)}{\tau} \tag{4}$$

in which N is the total number of cells used to build the DLL.

The Structured Datapath design shows a very similar behaviour between start and stop chains. This similarity results in a reduction on the measurement offset error. Moreover, the smaller DNL values across the DLL were achieved on the Structured Datapath layout. The maximum INL value was also greatly reduced.

As expected, the unconstrained design presents a much higher non-linearity, with DNL reaching 0.84 LSB and a maximum INL of 1.24 LSB. The constraint design, on the other hand, has a maximum DNL of 0.6 LSB and INL lower than 0.6 LSB for the worst-case conditions. These values represent a 50% improvement on the system's INL and a 30% improvement on the delay locked loop DNL. The results obtained highly support the premise that Structured Datapath should be used as a design methodology in order to achieve high linearity TDC without requiring extra calibration circuitry.

From the results presented, it is possible to notice, for both TDC layouts, that the higher DNL values are obtained at the first and last interpolation stages. This is because the first interpolation stage is built using the NAND and NOT gate, while the remainder of the DLL is built using clock buffers. In the case of the last interpolation stage, the discrepancy is due to the routing pattern used, which is different from the pattern of the remaining DLL, as well as the additional input load of the NAND gate used in the first stage. Again, these results highlight the impact of routing on the chain's linearity.

Another important aspect on the design is the clock skew between the flip-flops sampling the DLL. The system's clock tree was generated to guarantee a skew below 50 ps between the sampling flipflops. This way, it is secured by design that the TDC will not have thermometer codes with bubbles, which would make the decoding circuitry much more complex.

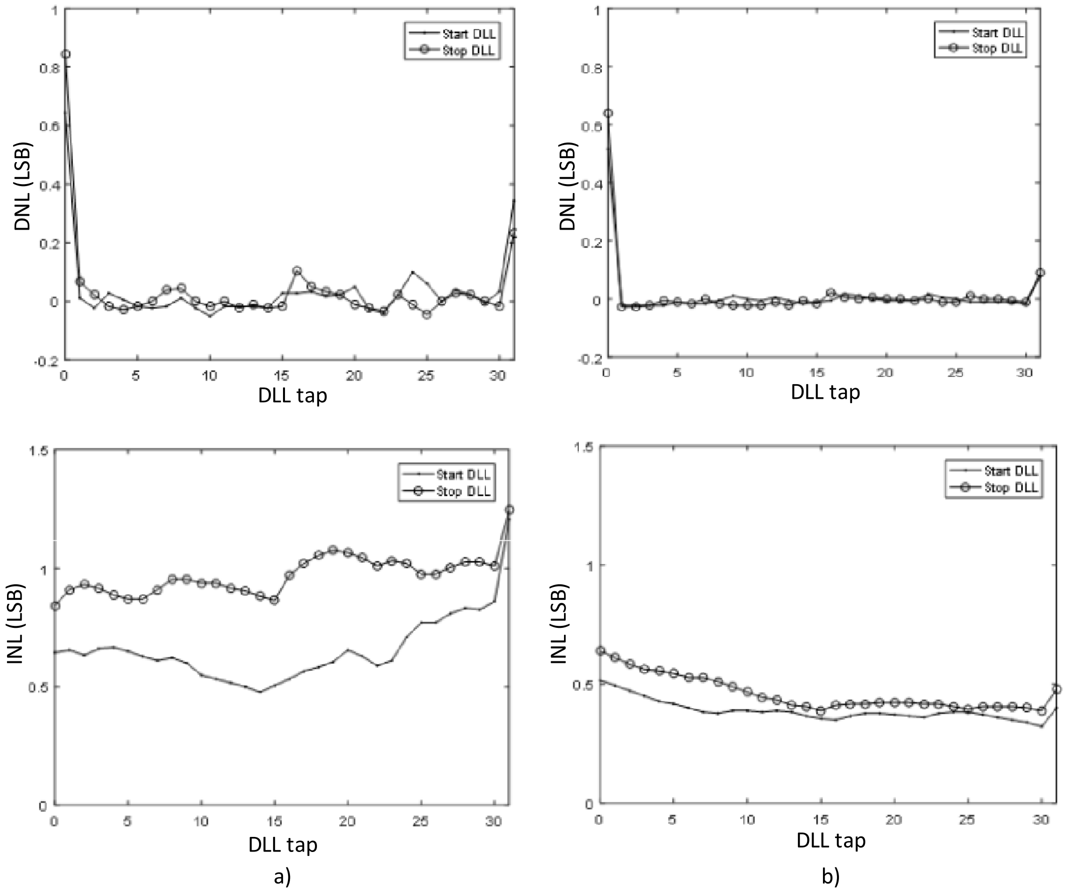


FIGURE 9. a) Non-constrained TDC placement DNL and INL and b) Structured Datapath Placed TDC DNL and INL.

TABLE 1. Performance summary and comparison.

Metric	This Work*	[15]*	[2]	[5]	[1]	[3]
Resolution (ps)	180	201.8	320	15	5	5
Range (us)	1310	6800	2.55	-	3.2	0.00064
DNL (LSB)	-0.02:0.63	0.18	0.68	+/-0.31	0.63	+/-0.9
INL (LSB)	0.32:0.51	-	-1.23:1.19	+/-0.68	1.47	+/-1.3
Supply Voltage (V)	1.8	1.8	3.3	-	1.2	1.3
Operating Frequency (MHz)	50	155	100	800	20	1562.5
Power (mW)	0.375	0.315	10.9	45.2	1.15	42
Die Size (mm)	0.17x0.17	1.25x1.25	0.39x0.39	0.91x0.215	0.837x0.837	1.52x0.85
CMOS Technology (µm)	0.18	0.18	0.35	0.18	0.13	0.13

*Post-layout simulation results

It is important to notice that, with technology scaling down, the use of Structured Datapath can be preponderant in achieving an all-digital TDC without any calibration mechanism. In lower size technologies, the effect of the routing RC parasitics has higher impact on gates' propagation delay. In such cases, DNLs higher than 1 LSB are expected when SDP is not use. In such scenarios, the existence of missing codes in the TDC is also expected and therefore, calibration mechanisms will be required.

When considering only one TDC channel, comprising the delay locked loop, the sample flip-flops, the loop counters, and the decoder, the total power consumption (internal and switching) is 0.375 mW, according to the post-layout reports generated by the CAD design tool.

A comparison between the proposed TDC post-layout results and some of the current state of the art TDCs is presented in TABLE 1. The proposed TDC can reach linearity values better than some state of the art TDCs [2], [3], using a fully automated design process, which greatly reduces the system implementation complexity and the development time, when compared to the adopted custom approach from the literature. Regarding resolution, the obtained results can also compete with some of the state of the art TDC works [2], [15]. Nevertheless, even though a smaller area is used, as well as a lower system clock frequency, the power dissipation reported here is higher when compared with [15]. In [15] the delay cells used to build the TDC were designed specifically for that application, and therefore optimized for their needs. In this work, standard clock buffer cells from TSMC 0.18 technology library were used. These cells are not optimized for this application and therefore may justify the worst performance in terms of power dissipation. Although, our approach with the use of standard buffer cells greatly reduces the TDC design complexity, reducing development time.

V. CONCLUSION

In this paper a high linearity all-digital TDC designed using a Structural Datapath approach in TSMC 0.18 μm CMOS process technology was presented. The proposed TDC achieved a LSB resolution of 180 ps and a DNL and INL lower than 0.6 LSB, for the worst-case scenario. The TDC's power consumption is 0.375 mW when supplied at 1.8 V.

The majority of the research that can be found in literature uses a full-custom approach, with PLLs or DLLs schemas to stabilize the delay line and reduce the system susceptibility to PVT conditions and thus increase its linearity. These approaches cannot be fully targeted by a HDL design and greatly increase the design complexity. In this work we propose the use of Structured Datapath which proved to be able to achieve high linearity without the need for extra calibration circuitry, thus reducing circuit area, power consumption and design complexity. The presented design can be fully implemented in HDL enabling for a full digital automated flow. Furthermore, no custom cells need to be designed. These factors contribute for a large reduction on system's development time.

With technology scaling down, and with it, an increasing impact of the routing on gates' propagation delay, the benefits of using SDP is expected to be even greater. The main challenge will be to secure low skew on the system clock tree that samples the registers of the TDC's delay line. If calibration could not be avoided, the proposed design can easily be extended to have software calibration. To do so, it is only needed to add a test input to the delay line that is multiplexed with the real input signal. The test signal can be used to build a histogram in software based on the code density method. These values can then be used to calibrate the output from the TDC when in normal operation.

REFERENCES

[1] C.-T. Ko, K.-P. Pun, and A. Gothenberg, "A 5-ps Vernier sub-ranging time-to-digital converter with DNL calibration," *Microelectron. J.*, vol. 46, no. 12, pp. 1469–1480, Dec. 2015.

[2] J. Wu, W. Zhang, X. Yu, Q. Jiang, L. Zheng, and W. Sun, "A hybrid time-to-digital converter based on residual time extraction and amplification," *Microelectron. J.*, vol. 63, pp. 148–154, May 2017.

[3] L. Perktold and J. Christiansen, "A fine time-resolution (≪ 3 ps-rms) time-to-digital converter for highly integrated designs," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2013, pp. 1092–1097.

[4] P. Fischer, I. Peric, M. Ritzert, and T. Solf, "Multi-channel readout ASIC for ToF-PET," in *Proc. IEEE Nucl. Sci. Symp. Conf. Rec.*, Oct./Nov. 2006, pp. 2523–2527.

[5] J. Mauricio, D. Gascón, D. Ciaglia, S. Gómez, G. Fernández, and A. Sanuy, "MATRIX: A novel two-dimensional resistive interpolation 15 ps time-to-digital converter ASIC," *J. Instrum.*, vol. 11, no. 12, Dec. 2016, Art. no. C12047.

[6] R. Turchetta, *Analog Electronics for Radiation Detection*, 1st ed. Boca Raton, FL, USA: CRC Press, 2018.

[7] R. A. Dias, F. S. Alves, M. Costa, H. Fonseca, J. Cabral, J. Gaspar, and L. A. Rocha, "Real-time operation and characterization of a high-performance time-based accelerometer," *J. Microelectromech. Syst.*, vol. 24, no. 6, pp. 1703–1711, Dec. 2015.

[8] F. Cossio, "A mixed-signal ASIC for the readout of gas electron multiplier detectors design review and characterization results," in *Proc. 13th Conf. Ph.D. Res. Microelectron. Electron. (PRIME)*, Jun. 2017, pp. 33–36.

[9] P. Dudek, S. Szczepanski, and J. V. Hatfield, "A high-resolution CMOS time-to-digital converter utilizing a Vernier delay line," *IEEE J. Solid-State Circuits*, vol. 35, no. 2, pp. 240–247, Feb. 2000.

[10] C.-C. Chen, S.-H. Lin, and C.-S. Hwang, "An area-efficient CMOS time-to-digital converter based on a pulse-shrinking scheme," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 3, pp. 163–167, Mar. 2014.

[11] Y. Wang and C. Liu, "A 4.2 ps time-interval RMS resolution time-to-digital converter using a bin decimation method in an UltraScale FPGA," *IEEE Trans. Nucl. Sci.*, vol. 63, no. 5, pp. 2632–2638, Oct. 2016.

[12] Y. Wang and C. Liu, "A nonlinearity minimization-oriented resource-saving time-to-digital converter implemented in a 28 nm Xilinx FPGA," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 5, pp. 2003–2009, Oct. 2015.

[13] C. Liu and Y. Wang, "A 128-channel, 710 M samples/second, and less than 10 ps RMS resolution time-to-digital converter implemented in a kintex-7 FPGA," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 3, pp. 773–783, Jun. 2015.

[14] J. Y. Won, S. I. Kwon, H. S. Yoon, G. B. Ko, J.-W. Son, and J. S. Lee, "Dual-phase tapped-delay-line time-to-digital converter with on-the-fly calibration implemented in 40 nm FPGA," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 1, pp. 231–242, Feb. 2016.

[15] A. Pokhara, J. Agrawal, and B. Mishra, "Design of an all-digital, low power time-to-digital converter in 0.18 μm CMOS," in *Proc. 7th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2017, pp. 1–5.

[16] T. Watanabe and H. Isomura, "All-digital ADC/TDC using TAD architecture for highly-durable time-measurement ASIC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 674–677.

[17] H. Chen, K. Briggl, P. Eckert, T. Harion, Y. Munwes, W. Shen, V. Stankova, and H. C. Schultz-Coulon, "MuTRiG: A mixed signal Silicon Photomultiplier readout ASIC with high timing resolution and gigabit data link," *J. Instrum.*, vol. 12, no. 1, Jan. 2017, Art. no. C01043.

[18] H. Chen, K. Briggl, P. Fischer, A. Gil, T. Harion, Y. Munwes, M. Ritzert, D. Schimansky, H.-C. Schultz-Coulon, W. Shen, and V. Stankova, "A dedicated readout ASIC for time-of-flight positron emission tomography using silicon photomultiplier (SiPM)," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, Nov. 2014, pp. 1–5.

[19] P. Fischer, I. Peric, M. Ritzert, and M. Koniczek, "Fast self triggered multi channel readout ASIC for time- and energy measurement," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 3, pp. 1153–1158, Jun. 2009.

[20] L. Perktold and J. Christiansen, "A multichannel time-to-digital converter ASIC with better than 3 ps RMS time resolution," *J. Instrum.*, vol. 9, no. 1, Jan. 2014, Art. no. C01060.

[21] Y. Yao, Z. Wang, H. Lu, L. Chen, and G. Jin, "Design of time interval generator based on hybrid counting method," *Nucl. Instrum. Methods Phys. Res. A, Accel., Spectrometers, Detectors Associated Equip.*, vol. 832, pp. 103–107, Oct. 2016.

[22] R. Szplet, P. Kwiatkowski, Z. Jachna, and K. Różyc, "An eight-channel 4.5-ps precision timestamps-based time interval counter in FPGA chip," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2088–2100, Sep. 2016.

[23] R. B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. T. Balsara, "1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 3, pp. 220–224, Mar. 2006.

[24] J. Lee and Y. Moon, "A design of Vernier coarse-fine time-to-digital converter using single time amplifier," *J. Semicond. Technol. Sci.*, vol. 12, no. 4, pp. 411–417, Dec. 2012.

[25] J. Wang, Y. Liang, X. Xiao, Q. An, J. W. Chapman, T. Dai, B. Zhou, J. Zhu, and L. Zhao, "Development of a time-to-digital converter ASIC for the upgrade of the ATLAS Monitored Drift Tube detector," *Nucl. Instrum. Methods Phys. Res. A, Accel., Spectrometers, Detectors Associated Equip.*, vol. 880, pp. 174–180, Feb. 2018.

[26] J. Mauricio, D. Gascon, D. Ciaglia, S. Gomez, G. Fernandez, and A. Sanuy, "MATRIX: A novel two-dimensional resistive interpolation 15 ps time-to-digital converter ASIC," in *Proc. IEEE Nucl. Sci. Symp., Med. Imag. Conf. Room-Temperature Semicond. Detect. Workshop (NSS/MIC/RTSD)*, Oct./Nov. 2016, pp. 1–3.

[27] *Innovus User Guide*, Cadence, San Jose, CA, USA, 2016.

RUI MACHADO was born in Guimarães, Portugal, in 1990. He received the M.Sc. degree in electronics engineering and computers from the University of Minho, Campus de Azurém, in 2014, where he is currently pursuing the Ph.D. degree in electronics and digital systems, in a project in partnership with Bosch Car Multimedia. During his Ph.D. degree, he was an Invited Professor with the Technology School, Polytechnic Institute of Cávado and Ave (IPCA) and the Electronics Department, University of Minho. He has been a Scientific Visitor with the International Iberian Nanotechnology Laboratory (INL), since 2017. His current research interests include time-to-digital conversion systems, and embedded and digital systems design.

JORGE CABRAL received the Ph.D. degree in microsystems technology from Imperial College London, London, U.K. He is currently an Assistant Professor with the University of Minho, Campus de Azurém, Portugal. His research interest includes embedded systems applications, and he is in charge of several research projects in this field.

FILIPE SERRA ALVES was born in Valena̧a, Portugal, in 1989. He received the Ph.D. degree in the microelectronics research area with a focus on pull-in based MEMS inclinometers with integrated electronics from the University of Minho. He was an Assistant Professor with the Industrial Electronics Department, University of Minho. He has been a Scientific Visitor with the Delft University of Technology and the International Iberian Nanotechnology Laboratory (INL). He is currently a Research Fellow with the Nanoelectronics Engineering Department, Nanodevices Research Group, INL. His research is focused on the development of integrated inter-ocular eye pressure monitoring systems, based on MEMS pressure sensors. His research interest includes the design and modeling of MEMS sensors to the design of mixed-signal integrated circuits.