

# High-resolution time-to-digital converters (TDCs) with a bidirectional encoder

Yu Wang   Wujun Xie   Haochang Chen   David Day-Uei Li

**Abstract:** A high-resolution time-to-digital converter (TDC) based on wave union (four-edge WU A), dual-sampling, and sub-TDL methods is proposed and implemented in a 16-nm Xilinx UltraScale+ field-programmable gate array (FPGA). We combine WU and dual-sampling techniques to achieve a high resolution. Besides, we use the sub-TDL method and the proposed bidirectional encoder to suppress bubbles and encode four-transition pseudo thermometer codes efficiently. **Experimental results indicate the proposed TDC achieves a 0.4 ps resolution with a 450 MHz sampling clock and a less than 9 ps RMS precision from 0 ns to 100 ns (achieving a 3.06 ps RMS precision in the best-case scenario).**(R3, comment 7) These characteristics make this design suitable for particle physics, biomedical imaging (such as positron emission tomography, PET), and general-purpose scientific instruments.

*Keywords*—time-to-digital converter (TDC), wave union (WU), dual-sampling, Sub-TDL, bidirectional encoder.

## I. Introduction

**Time-to-digital converters (TDCs) are high-precision time sensors converting a time interval (TI) into a digital code.**(R3, comment 6) They are widely used in time-resolved applications, such as particle physics [1–3], positron emission tomography (PET) [4,5], random number generation [6,7], Raman spectroscopy [8,9], and light detection and ranging (LiDAR) [10,11].

We can use application-specific integrated circuits (ASICs) or field-programmable gate arrays (FPGAs) [12] to implement TDCs. FPGA-TDCs are widely applied in scientific research and prototype designs, as they have a shorter development cycle and lower development costs than ASIC-TDCs. Most FPGA-TDCs use cascade carry

chains to form the tapped delay line (TDL) since carry chains are standard in modern FPGAs and have dedicated routing resources.

The resolution (the minimal detectable TI, usually called the least significant bit, LSB) is the primary metric for a TDC. For TDL-TDCs, it is determined by the carry chain's propagation delay and can be defined as:

$$Q = \frac{T}{n}, \quad (1)$$

where  $T$  is the period of the sampling clock and  $n$  is the number of time bins. Although TDL-TDC's resolution is improved with the advances in complementary metal-oxide-semiconductor (CMOS) manufacturing technologies, the achievable resolution is still limited. Architectures, such as Vernier [13] and multi-chain merging [14], have been proposed to break this process-related limitation. However, the Vernier architecture suffers from deadtime, and the multi-chain architecture consumes significant hardware resources. Hence, these two architectures have a trade-off between the resolution and other metrics.

In 2008, Wu and Shi proposed wave union (WU) TDCs, respectively achieving 25 ps (WU A) and 10 ps (WU B) precisions in a Cyclone II FPGA [15]. The concepts of these two methods are similar to the multi-chain merging architecture, utilizing multi-measurements for the same TI to achieve a better resolution and precision. However, in these two WU methods, multi-measurements are conducted by feeding a train of pulses (including rising and falling edges) rather than implementing parallel TDLs. Hence, WU methods are more effective in hardware utilization than multi-chain merging. The difference between WU A and WU B lies in the length of the pulse train. With a no-feedback wave launcher, WU A generates a finite-length pulse containing a limited number of logic transitions. However, WU B generates an infinite-length pulse train with a feedback wave launcher. Therefore, WU B has a better resolution than WU A but with a longer dead time [15]. Since the WU method was proposed, it has been the mainstream for high-resolution TDL-TDCs. For example, Wang *et al.* implemented a

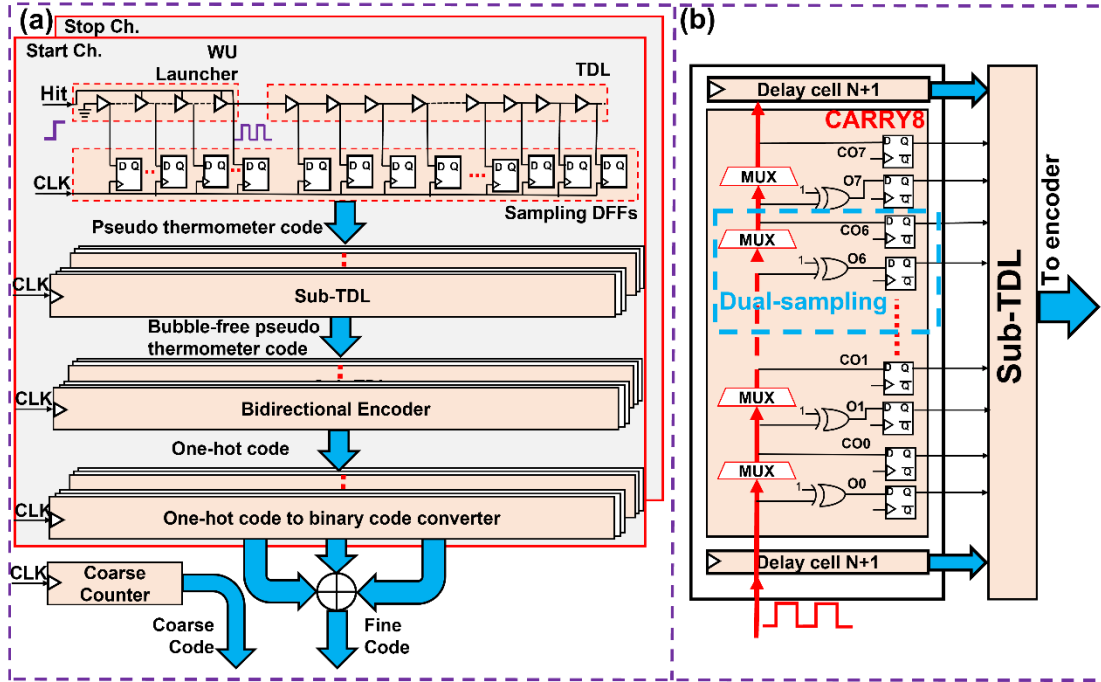


Fig. 1. The block diagram of the (a) proposed FPGA-TDC system, (R1 comment 2c) and (b) dual-sampling method with CARRY8.

4-snapshot TDC (WU B) in a Virtex-4 FPGA [16], Szplet *et al.* implemented a 6-edge TDC (WU A) in a Spartan-6 FPGA [17], and Xie *et al.* implemented a 2-edge TDC (WU A) in an UltraScale FPGA [18].

However, to our knowledge, no WU TDC is implemented in 16-nm FPGA devices. Hence, in this work, we designed and implemented a 4-edge WU TDC (WU A) in a 16-nm UltraScale+ MPSoC device. We combined the dual-sampling and WU methods to achieve a high resolution. Besides, we also used the sub-TDL method to suppress bubbles and the proposed bidirectional encoder to encode bubble-free four-transition pseudo thermometer codes.

The remainder article is structured as follows. Section II describes the architecture and design of the proposed TDC, Section III presents the experimental results, and

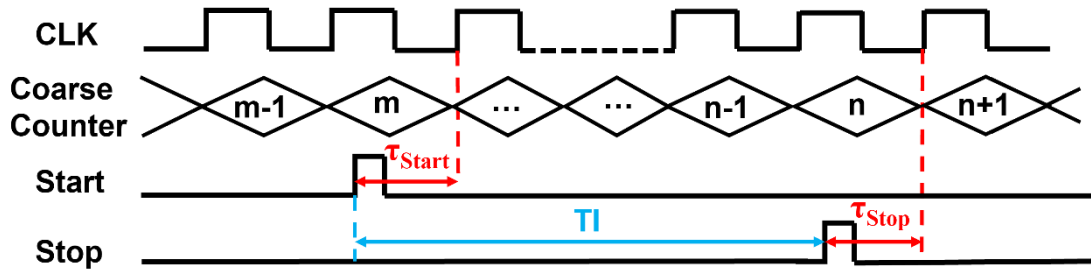


Fig.2. The timing diagram of measurement.

Section IV compares the proposed TDC with other WU TDCs. Finally, Section V concludes this work.

## II. Architecture and Design

The TDC system is shown in Fig. 1a. It consists of the start and stop channels (Start Ch. and Stop Ch. in Fig 1a). These two channels are driven by the same clock and responsible for recording starting and stopping timestamps for a TI. According to the timing diagram shown in Fig. 2, the starting and stopping timestamps can be respectively defined as:

$$timestamp_{start} = m \times T - \tau_{start}, \quad (2)$$

and

$$timestamp_{stop} = n \times T - \tau_{stop}, \quad (3)$$

where  $T$  is the period of the coarse counting clock,  $m$  and  $n$  are coarse codes output from the coarse counter, and  $\tau_{start}$  and  $\tau_{stop}$  are time intervals corresponding to respective fine codes. Hence, the TI can be calculated as:

$$TI = timestamp_{stop} - timestamp_{start} = (\tau_{start} - \tau_{stop}) + (n - m) \times T. \quad (4)$$

A binary or gray code counter can easily achieve the coarse counter. Here, we focus on fine-time interval measurements in this paper.

### A. CARRY8 and Dual-sampling

When a hit comes, the WU launcher generates a pulse train in each channel and feeds it into the TDL. Then, when a rising edge of the sampling clock comes, the TDL's outputs are registered by D flip-flops (DFFs) to evaluate the time interval between rising edges of the hit and the sampling clock ( $\tau_{start}$  and  $\tau_{stop}$  in Fig. 2).

Hence, as the delay cell, the CARRY8 is the backbone for TI measurements. Implemented by CARRY8s, the dual-sampling method [18–20] was firstly proposed by Wang and Liu [21]. (R3, comment 2) As shown in Fig. 1b, the TDL (highlighted in red) is based on cascaded CARRY8s, and each CARRY8 contains eight multiplexer (MUX)-based delay elements [22]. Like its predecessors (CARRY4s), each delay element in CARRY8s has two outputs (O and CO in Fig. 1b). Differently, in each slice, these two outputs can be simultaneously sampled by DFFs in the CARRY8 rather than either “O” or “CO” sampled in the CARRY4 [23]. Therefore, with the dual-sampling method, eight delay elements can output sixteen taps, equivalent to subdividing one bin into two bins. Thus, the resolution is improved without increasing the system complexity.

## B. Sub-TDL and Wave Union Launcher

Although the TDL’s outputs are sampled by DFFs, they cannot be encoded directly as there are unexpected logic transitions (for example, unexpected “0” among “1”s or “1” among “0”s), usually called bubbles. Bubbles are caused by clock skews and TDLs’ uneven propagation delays [24], and they cause coding errors. The Wallace tree encoder [25], bin realignment [26], and ones-counter [27] were proposed to resolve bubble problems. However, the Wallace tree encoder and ones-counter increase hardware utilization, and iterations for bin realignment are time-consuming. Hence, these methods are unsuitable for a TDC with many time bins.

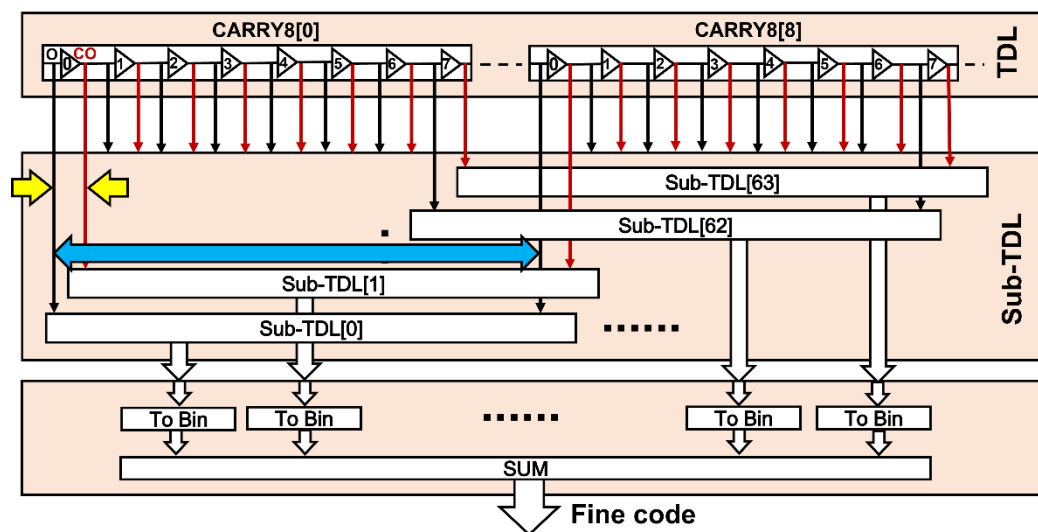


Fig. 3. The architecture of Sub-TDL.

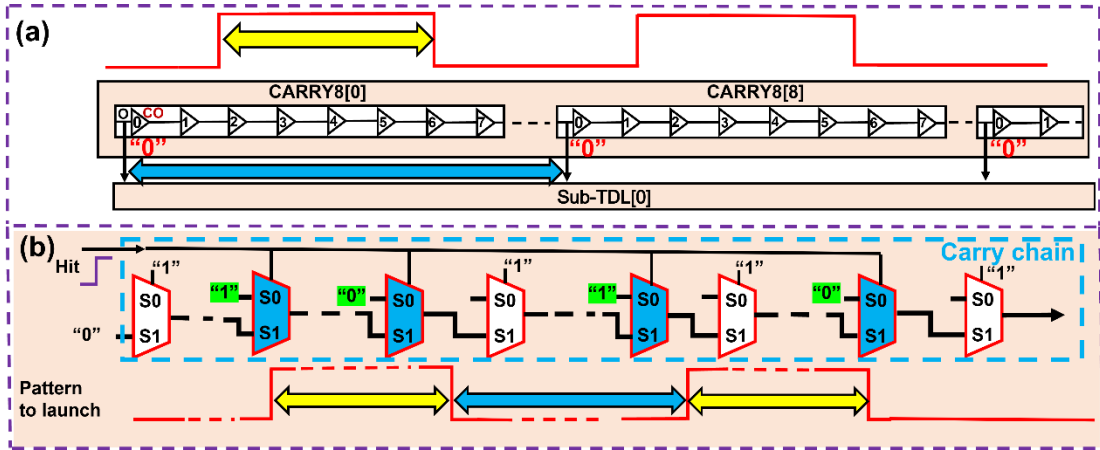


Fig.4. (a) The concept of undetectable wave patterns in the Sub-TDC. (b) The block diagram of wave union (WU A) launcher.

To resolve bubble problems without extra design complexity, we used the sub-TDL method [24] (or the decomposition method [28]) in our design. The sub-TDL architecture is shown in Fig. 3. The bin-width of the dual-sampling TDL is highlighted in yellow, and the sub-TDL's bin-width is highlighted in blue. The sub-TDL elongates time intervals between taps by decomposing TDL's sampling taps to minimize the impact of clock skews and obtain bubble-free outputs. Then all subsets (results from sub-TDLs) are summed and interpolated to maintain the TDC's resolution. The number of sub-TDLs depends on the maximal bubble depth (MBD). In Ref. [24], there are four sub-TDLs for a Virtex-7 FPGA and eight sub-TDLs for a Kintex UltraScale FPGA. However, in the UltraScale+ MPSoC device, the observed MBD is sixty for the dual-sampling TDL. Hence, we build sixty-four Sub-TDLs for our design.

However, as shown in Fig. 4a, when the WU method is combined with sub-TDLs, logic transitions may be undetectable in sub-TDLs if the pulse width (highlighted in yellow in Fig. 4a) is shorter than sub-TDL's bin-width (highlighted in blue in Fig. 4a). To avoid this and precisely control the pulse width, we implemented the WU launcher with CARRY8s (shown in Fig. 4b). By using the input signal ("hit" in Fig. 4b) as the MUX-based delay elements' select signal, the WU launcher can work in the "Standby Mode" and "Launch Mode". When the input is "0" (low-logic level), the WU launcher works in the "Standby Mode" and stores the wave pattern in the **carry chains**. Then the stored pattern is launched when the input changes to "1" (high-logic level). The stored pattern is configured by the "S0" input (highlighted in green in Fig. 4b) of the delay

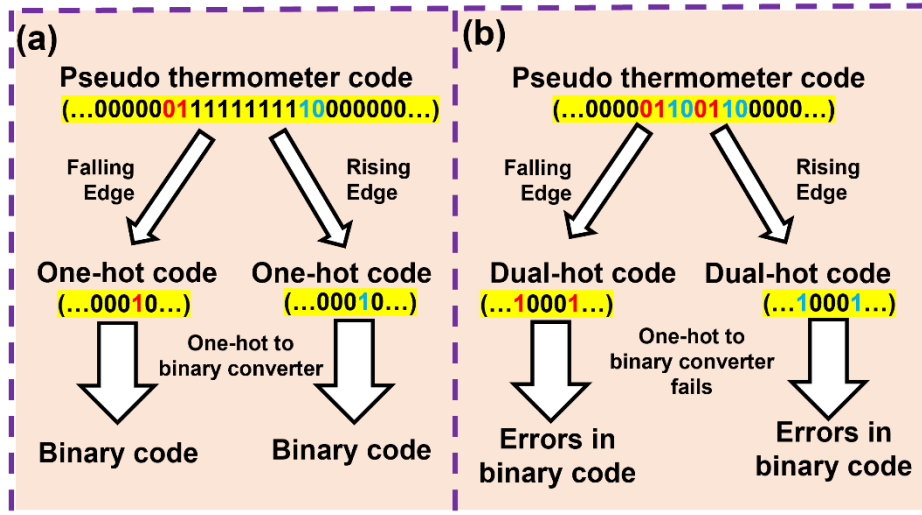


Fig.5. Encoding workflow of (a) two-edge WU TDC and (b) four-edge WU TDC.

element. The pulse width is configured by the number of delay elements among adjacent “configuring elements” (highlighted in blue in Fig. 4b). Our design configures the wave pattern as “01010” to contain four logic transitions. In the designed WU launcher, the width of the positive pulse (highlighted in yellow in Fig. 4b) is configured as 80 taps (5 CARRY8s) in the dual-sampling TDL. However, rising edges propagate faster than falling edges in the TDL, causing the width of the negative pulse (highlighted in blue in Fig. 4b) to decrease when a pulse train propagates along the TDL [18]. Hence, we configure the negative pulse width as 112 taps (7 CARRY8s) in the dual-sampling TDL to ensure the negative pulse is always detectable in every sub-TDL. (R1, comment 1)

### C. Bidirectional Encoder

With the well-designed WU launcher, four logic transitions (two rising edges and two falling edges) are detectable in every sub-TDL. In our previous work (a TDC with two-edge WU A) [29], rising and falling edges in sub-TDLs’ outputs are respectively detected and converted to one-hot code by positioning “1-0” and “0-1” edges. Then every one-hot code is converted to the corresponding binary code for final result calculations. However, that strategy is out-of-work in the proposed TDC since there are more than one rising and falling edges in every sub-TDL’s output (the comparison is shown in Fig. 5). Hence, we propose a bidirectional encoder for the four-edge WU A

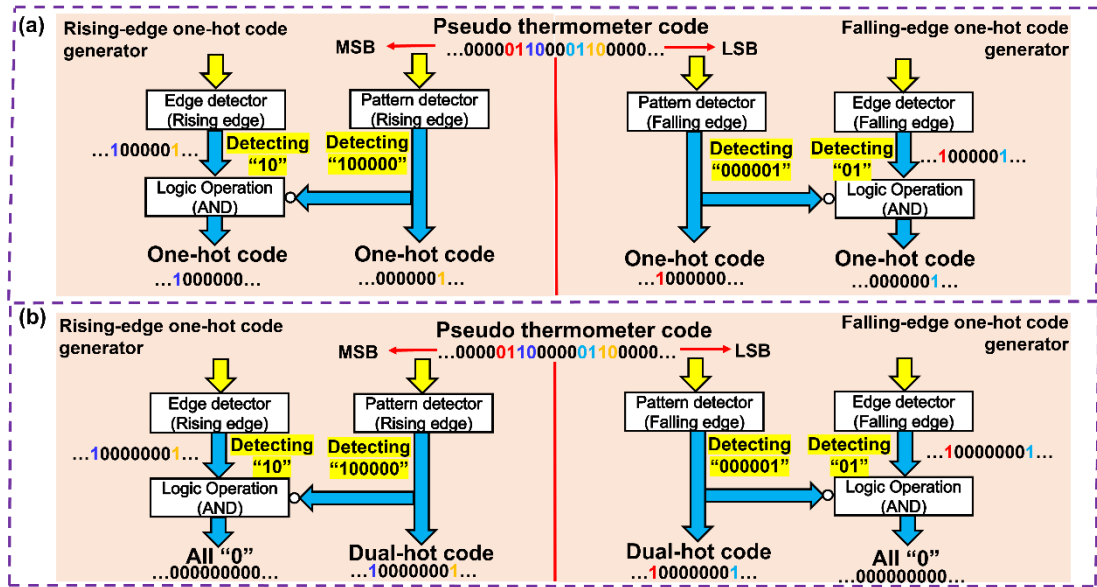


Fig.6. Block diagram and encoding flow of the bidirectional encoder when the width of the negative pulse in the sub-TDL is (a) less than 5 bits and (b) more than 5 bits.(R1, comment 2a)

TDC.

The bidirectional encoder's block diagram and encoding flow are shown in Fig. 6a. It consists of the rising-edge and falling-edge one-hot code generators, responsible for converting a four-transition pseudo thermometer code to four one-hot codes. For a one-hot code generator, it contains a pattern detector and an edge detector, respectively responsible for detecting the specific logic-transition pattern and the specific logic-transition edge. For example, in the rising-edge one-hot code generator, the pattern detector locates the pattern "100000", whereas the edge detector detects the rising edge "10".(R1, comment 2e) Then the pattern detector can generate a one-hot code since the width of the negative pulse (highlighted in blue in Fig. 4b) is precisely controlled to less than 5 taps in every sub-TDL. However, two "10" edges can be found in the pseudo thermometer code as shown in Fig.6a. Hence, we input the edge detector's output and the pattern detector's output to a AND gate to get the other one-hot code from the rising edge. The falling-edge one-hot code generator is similar to the rising-edge one-hot code generator (shown in Fig.6a). Differently, for the falling-edge detection, the pattern detector identifies the pattern "000001", and the edge detector identifies the edge "01".(R1, comment 2e) Then, similar to our previous work [29], we can convert four one-hot codes to four binary codes and then generate the final code. Besides, Fig. 6b shows the



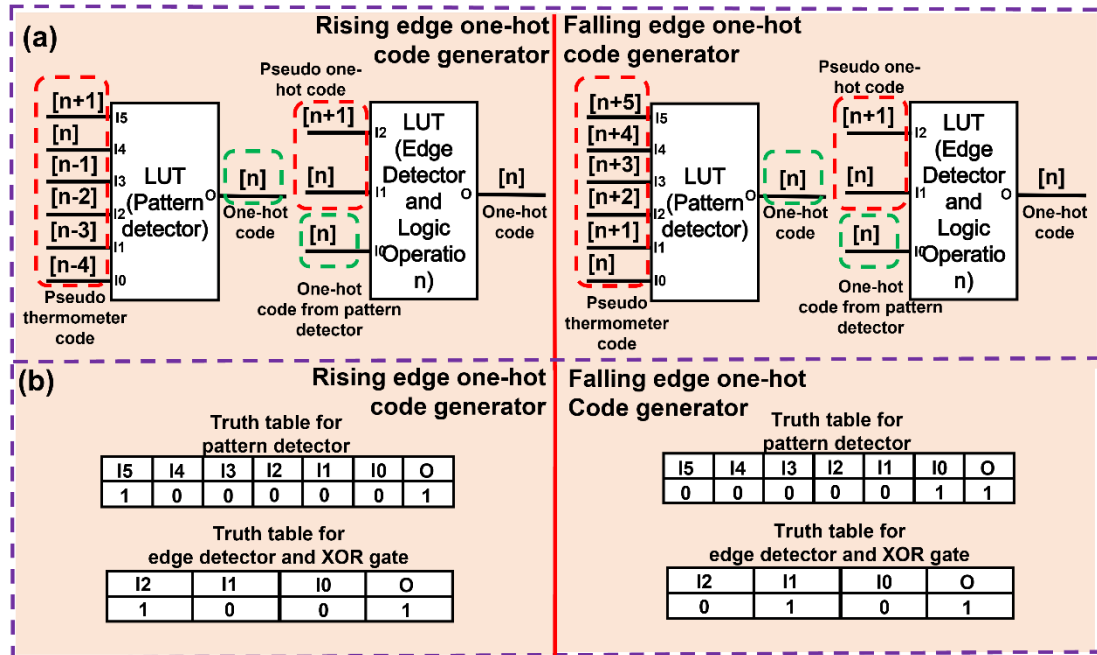


Fig.7. (a) Hardware implementation of the bidirectional encoder. (b) Truth tables for the bidirectional encoder. (R1, comment 2b)

encoding flow when the negative pulse width (highlighted in blue in Fig. 4b) is more than 5 bits in a sub-TDL. The width exceeding the limit (5 bits in sub-TDLs) causes the pattern detector to generate a one-hot code unsuccessfully, which also leads to the incorrect output from the AND gate.

The hardware implementation of the bidirectional encoder is shown in Fig. 7a. We used a 6-input LUT to implement the pattern detector and a 3-input LUT to implement the edge detector and the AND gate. In our design, all LUTs for the bidirectional encoder are instantiated by Vivado Primitive [30]. However, LUTs are configured differently for rising and falling-edge one-hot code generators. The truth tables for LUTs are shown in Fig. 7b.

As shown in Fig. 1, one-hot codes from the bidirectional encoders are input to one-hot code to binary code converters and generate corresponding binary codes following

```

Input  [Width-1:0]      One_hot_in,
Output [Celi(√Width)-1:0] Binary_out,

For (i=0; i< Width; i=i+1) begin
    if(One_hot_in[i])
        Binary = i;
end
    
```

Fig.8. Pseudo codes of the one-hot code to binary code converter.

the pseudo codes shown in Fig. 8. Then, all binary codes from sub-TDLs are summed together through 4-input adders, and the final result is sent to the PC through universal asynchronous receiver-transmitter (UART). On the PC, the timestamp of each bin can be calculated with the code density tests [31] and bin-by-bin calibration method [32] (details are shown in III. A and B). Finally,  $\tau$  shown in Fig.2 can be evaluated on the PC with the bin's timestamps.(R1, comment 2c) In our design, we use sequential logical circuits for the TDL's sampling, sub-TDLs, bidirectional encoders, one-hot code to binary code converters, and adders (shown in Fig. 1a). Benefiting from this sequential logical design, the encoding flow is in pipeline, and the total latency (from sampling DFFs to adders) of our design is 8 system clocks (It takes 1 clock to output results for sampling TDLs, sub-TDLs, bidirectional encoders and one-hot code to binary code converters. However, parallely summing all 64x4 subsets together by the 4-input adder consumes 4 clocks). (R1, comment 2d)

### III. Experimental Results

We implemented the proposed TDC in the ZCU104 evaluation board [33] and closely placed two channels (Start Ch. and Stop Ch. in Fig. 1a) to reduce the offset. To evaluate TDC's performances, we used an SRS CG-635 as an external signal source. The same signal was simultaneously fed into two channels to reduce measurement errors and jitters from cables connecting the signal source and the evaluation board. For code density tests [31], this input can be treated as a random hit for two channels since it is asynchronous with the TDC's sampling clock. We used the input signal's period as a TI for RMS precision tests. Meanwhile, we can also measure the offset between two channels by calculating the difference between the same edge's timestamps recorded by two channels. The sampling clock (also known as the system clock) was sourced from an onboard crystal oscillator (IDT-8T49 [33]), and the frequency of the sampling clock was configured as 450MHz.

#### A. Bin-Width and Linearity

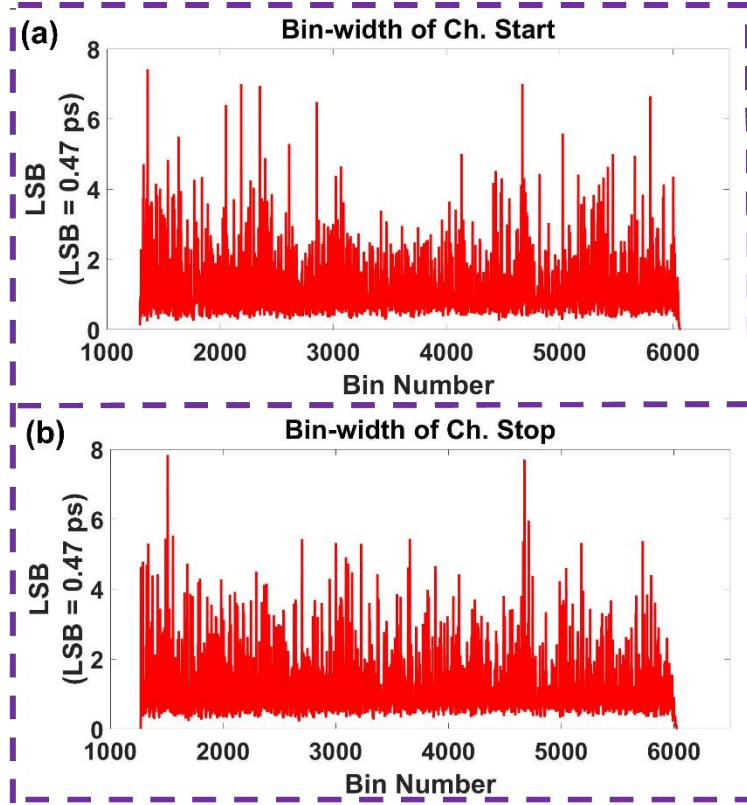


Fig.9. Bin-width of (a) the start channel and (b) the stop channel.

The bin-width is the quantization step of each time bin. To ensure the TDL can cover a sampling period, we increased the length of the TDL until two continuous outputs can be detected when the hit appears close to the rising edge of the sampling clock. Then, twenty million random hits (with 99.777777MHz frequency) were fed into two channels for code density tests to estimate the fine codes' bin-width of the designed TDC.(R3, comment 4) According to the number of hits collected at the  $k$ -th bin ( $n_k$ ), the bin-width can be estimated from:

$$W[k] = T \times \frac{n_k}{N} \quad (5)$$

where  $N$  is the number of random hits and  $T$  is sampling clock period (2.222ns with a 450MHz sampling clock).(R1, minor a) However, suffering from clock skews and mismatches, the widths of time bins differ. These differences lead to a transfer curve rather than the desired linear quantization steps [12]. With only fine codes, a TDC's linearity can be characterized by differential nonlinearity (DNL) and integrated nonlinearity (INL) as [12]: (R1, minor b)

$$DNL[k] = W[k] - Q, \quad (6)$$

and

$$INL[k] = \sum_{j=0}^k DNL[j]. \quad (7)$$

Besides, Wu also proposed the equivalent bin-width ( $\omega_{eq}$ ) and its deviation ( $\sigma_{eq}$ ) to evaluate the TDC's linearity [34]. They are defined as:

$$\sigma_{eq}^2 = \sum_{i=1}^n \left( \frac{W[i]^2}{12} \times \frac{W[i]}{W_{total}} \right) \quad (8)$$

and

$$\omega_{eq} = \sigma_{eq} \times \sqrt{12} = \sqrt{\sum_{i=1}^n \frac{W[i]^3}{W_{total}}}, \quad (9)$$

where  $W_{total} = \sum_{i=1}^n W[i]$ .

The proposed TDC's bin-width is shown in Fig. 9. In both channels, the first valid time bin (not a zero-width bin) appears at around the 1250<sup>th</sup> bin, caused by the WU launcher. A part of the carry chain constructs the WU launcher, and the pattern is generated and stored before a hit comes. Hence, the TDC's output is not zero, although there is no input and the output increases when an input hit appears. Besides, a cluster of narrow bins appears at the tail of valid time bins in both channels, caused by clock jitters. All of the above parameters for both channels are summarized in TABLE I.

Ch. Start		Ch. Stop	
LSB (fs)	465	LSB (fs)	466
DNL (LSB)	[-0.99,6.42]	DNL (LSB)	[-1,6.84]
INL (LSB)	[-8.79,51.56]	INL (LSB)	[-2.57,72.55]
$\omega_{eq}$ (ps)	1.81	$\omega_{eq}$ (ps)	1.85
$\sigma_{eq}$ (ps)	0.52	$\sigma_{eq}$ (ps)(R3, comment 5)	0.53

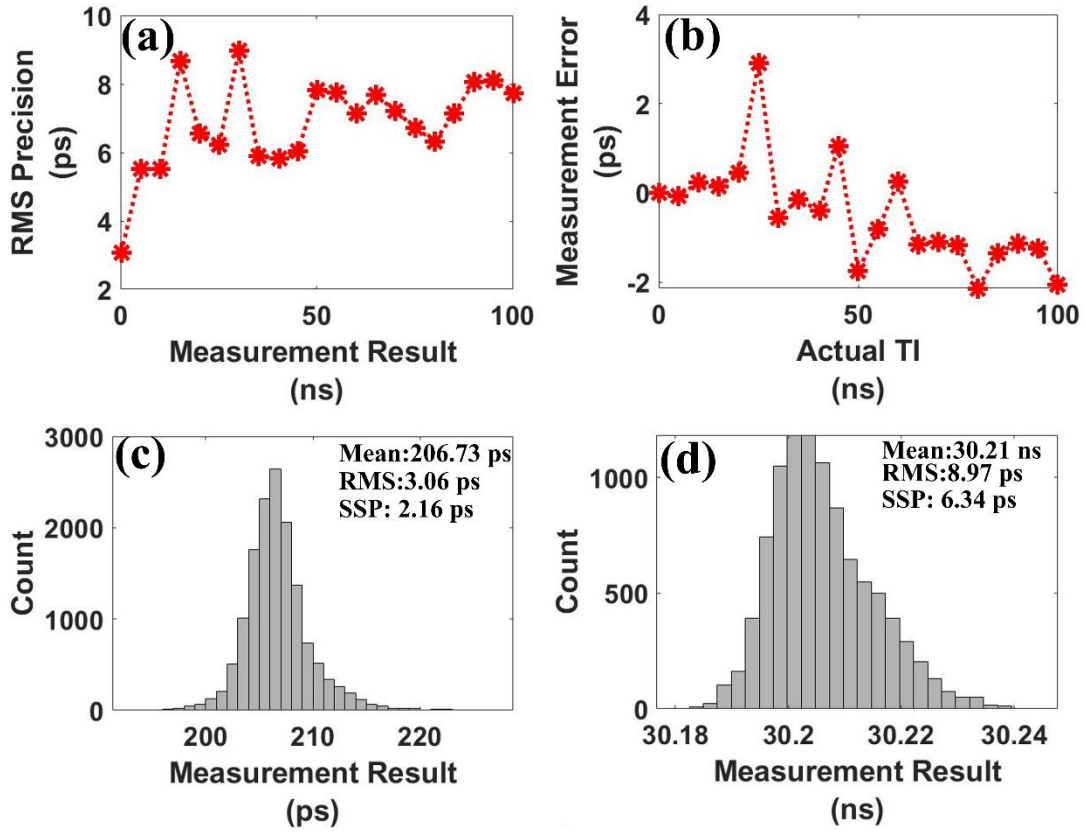


Fig.10. (a) RMS precisions of the proposed TDC. (b) Measurement errors of the proposed TDC. (c) Measurement histogram for the TI = 0 ns. (d) Measurement histogram for the TI = 30 ns.

## B. Time Interval Tests

The RMS precision represents the measurement uncertainty introduced by jitters and quantization errors [31]. It is evaluated by the standard deviation ( $\sigma$ ) of repetitive measurements for a fixed TI and improved by the bin-by-bin calibration [32]. They are respectively calculated as:

$$\sigma^2 = \sum_{i=1}^{N_T} \frac{(x_i - \mu)^2}{N_T - 1}, \quad (10)$$

and

$$t_k = \frac{W[k]}{2} + \sum_{j=0}^{k-1} W[j], \quad (11)$$

where  $x_i$  is the  $i$ -th measurement,  $\mu$  is the average value for  $N_T$  measurements when the TI is constant, and  $t_k$  is the calibrated timestamp corresponding to the center

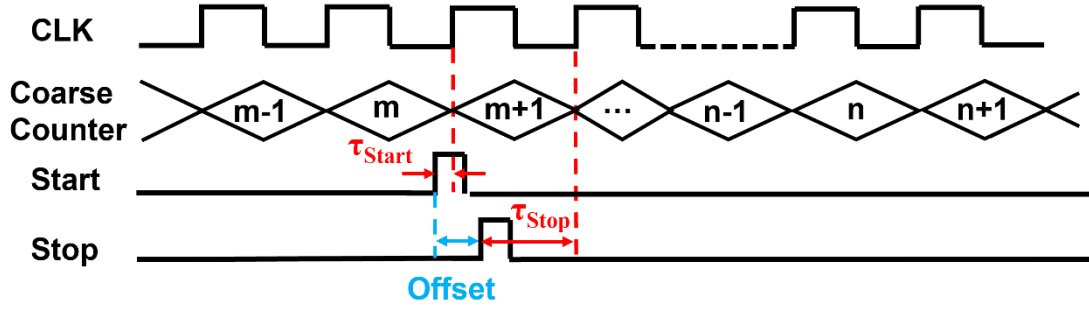


Fig. 11. The timing diagram for the measurement with the coarse counter when  $TI=0$  ns (offset measurement).

of the  $k$ -th time bin.

External measurement errors and jitters are minimized for TI tests by simultaneously feeding the same signal into two channels. Then ten thousand samples are captured for each fixed TI. The RMS precision of the proposed TDC is shown in Fig. 10a. The TI varies from 0 ns to 100 ns with a 5 ns incremental step. Among all measured TIs, the best RMS precision appears when the TI equals 0 ns, achieving 3.06 ps corresponding to 2.16 ps single-shot precision (SSP,  $SSP=RMS/\sqrt{2}$ ). Besides, the average value of measurements is 206.73 ps in this scenario, equal to the offset between two channels (Ch. start and Ch. Stop). RMS precision deteriorates with the TI increasing, achieving the worst RMS precision of 8.97 ps (corresponding to 6.34 ps SSP) at  $TI = 30$  ns. The RMS precision fluctuates between 5 ps and 9 ps in the measured range except for  $TI = 0$  ns. This phenomenon we speculate is caused by the jitter of the coarse counter's counting clock (influencing  $T$  in eq. (4)) and the jitter of the input signal. In repetitive measurements for  $TI = 0$  ns, only a few measurements are achieved with the coarse counter. And this only happens when the hit appears at the end of Ch. Start in the  $m$ -th coarse-counting period and appears at the beginning of Ch. Stop in the  $(m+1)$ -th coarse-counting period due to the offset (shown in Fig. 11). However, for the rest of the TI values in TI tests, the coarse counter is always required because TIs are more than one counting period. Besides, the delay is only introduced by the internal offset between channels when  $TI = 0$  ns. However, delay of the external signal source is required for other situations (such as  $TI = 5$  ns), causing the jitter of signal source to be introduced to the measurement. Due to these two reasons, the RMS precision at TI

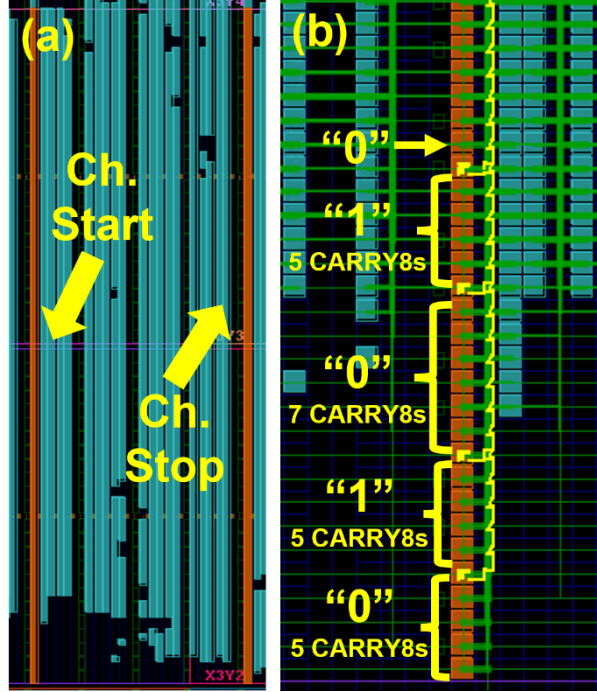


Fig.12. (a) The placement of the start and the stop channels. (b) Hardware implementation of the WU launcher.(R1, comment 1)

= 0 ns is much better than that of the rest TIs.(R3, comment 8) Besides, we analyzed measurement errors (shown in Fig. 10b) as:

$$E = (\mu - T_{offset}) - T_{actual}, \quad (12)$$

where  $T_{offset}$  is the offset between two channels, and  $T_{actual}$  is the actual value of the measured TI controlled by the external signal source. Results indicate our design has less than 3 ps measurement errors in the measurement range from 5 ns to 100 ns.

### C. Hardware implementation

The **resource usage (R3, comment 9)** of the proposed TDC is summarized in TABLE II. **Each channel has 1920 taps from the dual-sampling TDL, consuming 234 CARRY8s, 13547 DFFs and no more than 11782 LUTs. Among them, 22 CARRY8s are used to construct the WU launcher, and 98 CARRY8s are used as the TDL. (R1, minor c) Besides, extra 1920 DFFs and 960 LUTs are used as sampling circuits to sample the WU launcher and TDL's outputs. (R1, comment 1; R2, comment 1)** In addition to these, the rest hardware resources are mainly used for the encoding logic. The proposed bidirectional encoder's **resources usage (R3, comment 9)** is significant of the encoding logic. **For the proposed 30-bit-input rising edge (or falling edge) one-hot code generator, the pattern detector consumes 29 LUTs, and the combination of the edge detector and the AND gate consumes 28 LUTs. However, the Xilinx Vivado tool automatically inserts LUTs when DFFs sample outputs of the bidirectional encoders; hence, the global bidirectional encoder consumes 8976 LUTs and 7296 DFFs (including combinatorial logics and sequential logics for all 64 sub-TDLs' bidirectional encoders). We speculate the automatically inserted 1680 LUTs are used to improve the timing closure and fan-out capability.(R1, comment 2f)** Moreover, the hardware usage of the encoding logic also depends on tap numbers and implemented devices. **With the dual-sampling method, the proposed TDC needs at least 1250 taps (about 78 CARRY8s) to cover the sampling clock's whole period due to the low propagation delay of the CARRY8s. Besides, extra 352 taps (22 CARRY8s) are required for wave pattern generation. They both cause the number of time bins to increase, leading to the**

Resource	Utilization (%)	
	Ch. Start	Ch. Stop
Tap Number	1920 (-)	1920 (-)
CARRY8	234 (0.81%)	234 (0.81%)
DFF	13547 (2.94 %)	13547 (2.94 %)
LUT	11773 (5.11%)	11782 (5.11%)



encoder's significant **resource usage(R3, comment 9)**.

Figure 12a shows the placement of the start and stop channels. We placed these two channels close to reduce the offset. Moreover, we also routed manually for the WU launcher to ensure steady wave pattern generation. The hardware implementation of the

TABLE III  
COMPARISON OF RECENTLY PUBLICATIONED WU TDCs

Ref-year	Method	Devi.	LSB (ps)	RMS (ps)	LUT (%) <sup>1</sup>	DFP (%) <sup>1</sup>	CARRY (%) <sup>1</sup>	Real-time/Post Encoding
[15]-08	WU-A (2 edges)	Cyclone II	30	25	-	-	-	Post Encoding
	WU-B		2.44	10	-	-		
[35]-11	WU-A (2 edges), Bin-by-bin Cali.	Virtex-4	6<	8.8 <sup>2</sup>	-	-	-	Real-time Encoding
[16]-11	WU-B, Multichain-ave.	Virtex-4	12	9	-	-	-	-
[17]-16	Super WU (6 edges , 3 coding lines)	Spartan-6	0.90	<6	-	-	-	-
[36]-19	WU-A (4 edges), Bin-by-bin Cali.	Kintex-7	2.65 <sup>3</sup>	3.5	1410 <sup>3</sup> (1.38)	2732 <sup>3</sup> (1.34)	-	Real-time Encoding
	WU-A (8 edges), Bin-by-bin Cali.		1.33 <sup>3</sup>	3.0	2005 <sup>3</sup> (1.98)	3751 <sup>3</sup> (1.85)	-	
[37]-19	Super WU (2 edges , 8 coding lines)	Kintex-UltraScale	0.31	12.32	- <sup>4</sup>	- <sup>4</sup>	-	Real-time Encoding
[38]-21	MSWU Bin-by-bin Cali.	Kintex-7	0.39	3.30 <sup>5</sup>	-	-	-	Post Encoding
[18]-22	WU-A (2 edges), Sub-TDL, Dual-sampling	Kintex-UltraScale	1.23	5.19 <sup>5</sup>	2460 (1.01)	3463 (0.71)	88 <sup>6</sup> (0.29)	Real-time Encoding
[39]-22	WU-A (5 edges) DSP-chain, Chunk Encoding	Artix-7	-	16.25	- <sup>7</sup>	- <sup>7</sup>	- <sup>7</sup>	Post Encoding
This Work	WU-A (4 edges), Sub-TDL, Dual-sampling, Bidirectional Encoder	UltraScale+MPSoC	0.46	<9 (3.06) <sup>2</sup>	11773 (5.11)	13547 (2.94)	234 <sup>6</sup> (0.81)	Real-time Encoding

<sup>1</sup> Percentage of resource utilization for the implemented devices; <sup>2</sup> RMS precision in the best-case scenario; <sup>3</sup> Calculate from the literature; <sup>4</sup> 3200 SLICES, 400 kbit RAM and 1 DSP are used; <sup>5</sup> Value calculated from SSP ; <sup>6</sup> CARRY8s; <sup>7</sup> 10% DSPs are used in Artix-7 XC7A200T.

four-edge WU launcher is shown in Fig. 12b.

## IV. Comparisons and Discussions

TABLE (R3, comment 10) III summarizes recently published WU TDCs. As shown in TABLE III, most TDCs aim at the resolution from 1 ps to 10 ps, and the TDCs in Ref. [17], [37], [38] and the proposed TDC aim at a sub-picosecond resolution. The TDCs in Ref. [17] and [37] are based on the super WU method (combining WU A and multi-chain merging) and consume significant hardware resources. Although the multi-sampling WU method (MSWU, a technique combining WU A and WU B) in Ref. [38] can achieve a high resolution without increasing much resource usage (R3, comment 9), the encoding is complex, and it is hard to perform real-time encoding in hardware platforms. Hence, our design has a trade-off between encoding complexity and hardware utilization. And it achieves real-time encoding for a high-resolution TDC with the proposed bidirectional encoder.

Compared with other four-edge WU TDCs with real-time encoding (such as that in Ref. [36]), the proposed TDC also has competitive hardware utilizing efficiency. Our design is similar to the four-edge WU TDC in Ref. [36], apart from the encoder. Due to the TDL's high propagation delay, the TDC in Ref. [36] requires 288 taps to build the WU launcher and cover the whole sampling period (with a 554 MHz sampling frequency), consuming 1410 LUTs and 2732 DFFs for each channel. By contrast, due to the TDL's low propagation delay, our design requires more than 6.5-fold taps (1920) for the WU launcher and the dual-sampling sub-TDL, consuming 8.35-fold LUTs and only 4.95-fold DFFs. The comparisons of resource usage (R3, comment 9) between the proposed TDC and the TDC in Ref. [36] indicate that these two designs have similar hardware utilization efficiency. However, our TDC only requires configuring LUTs for four cases (truth tables are shown in Fig. 6b), reducing complexity significantly compared with the TDC in Ref. [36].

For process, voltage and temperature (PVT) calibration, this paper does not detailly

discuss it. However, we suggest that the PVT calibration can be conducted on the PC when measuring TIs. As discussed in Sec. III, when the input signal is asynchronous with the TDC's system clock, it can be treated as a random hit for both the start and stop channels. Hence, code density tests, bin-by-bin calibration and measurement can be conducted simultaneously. This online update for bins' timestamps ensures measurements are influenced by PVT as lightly as possible. (R2, comment 2) Although our design's resource usage (R3, comment 9) is acceptable, the hardware utilization efficiency can be further improved. For example, we can use DSP48 to sum all subsets from sub-TDLs to reduce the consumption of DFFs and LUTs. We can also multiplex adders to optimize hardware utilization. Moreover, due to the jitters of the coarse counter's clock, the proposed TDC's RMS precision deteriorates when the TI is measured with the coarse counter. These two aspects still need to be improved in future work. Besides, the designed TDC achieves an ultra-high resolution with the WU method (four edges, WU-A) and the proposed bidirectional encoder. However, for the proposed architecture, it is difficult to encode for more than four logic transitions with satisfactory hardware utilizing efficiency. First, the bidirectional encoder servers for the one-hot code to binary code generator, and it needs to output the same number of one-hot codes as the number of logic transitions. When logic transitions increase, more patterns are required to distinguish different logic transitions' positions and ensure correct one-hot codes can be generated for each logic transition. For example, a new pattern, "100000000", is required for an extra rising edge. However, the pattern detector for "100000" also outputs "1" for the new pattern "100000000". Hence, an extra logical operation between the outputs of pattern detectors for "100000" and "100000000" is required. At the same time, more logical operations are also needed to extract the one-hot code from the output of the edge detector since more edges "10" can also be detected. Moreover, the maximum number of LUTs' inputs is 6 in Xilinx 28-nm and more advanced FPGAs. Therefore, the pattern detector cannot be implemented by a single LUT when the width of the pattern exceeds 6 bits. Besides, more logic transitions require more CARRY8s for the WU launcher. Hence, the bidirectional encoder and the WU launcher's complexity and resource usage increase simultaneously

with the incremental logic transitions. Moreover, with the increased number of logic transitions, the number of one-hot to binary code converters is also proportionately increased. These characteristics limit the proposed architecture to pursue higher performances with acceptable resource usage in the 16-nm FPGA devices (R2, comment 3; R3, comment 1).

## V. Conclusions

Combining the dual-sampling method and the sub-TDL method, we first implemented the four-edge WU TDC in a 16-nm UltraScale+ MPSoC device. We propose a bidirectional encoder to real-time encode the four-transition pseudo thermometer code and achieve a 0.46 ps resolution and less than 9 ps RMS precision with a less than 3 ps measurement error. The hardware implementation of the encoder and WU launcher is also detailed in this report. Experimental results indicate that the proposed TDC is suitable for particle physics, biomedical imaging (such as positron emission tomography, PET), and general-purpose scientific instruments.

### ACKNOWLEDGEMENTS

The research has been supported by the Engineering and Physical Sciences Research Council under EPSRC Grant: EP/L01596X/1, the Royal Society of Edinburgh, and the China Scholarship Council. We would also like acknowledge the support from the Xilinx for donating FPGA develop kits to the research group.

## Reference

- [1] A. Banerjee, M. Wiebusch, M. Poletini, A. Mistry, H. Heggen, H. Schaffner, H.M. Albers, N. Kurz, M. Górska, J. Gerl, Analog front-end for FPGA-based readout electronics for scintillation detectors, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. 1028 (2022) 166357. <https://doi.org/10.1016/j.nima.2022.166357>.
- [2] F. Bouyjou, G. Bombardi, F. Dulucq, A. El Berni, S. Extier, M. Firlej, T. Fiutowski, F. Guilloux, M. Idzik, C. De La Taille, A. Marchioro, A. Molenda, J. Moron, K. Swientek, D. Thienpont, T. Vergine, HGCROC3: the front-end readout ASIC for the CMS High Granularity Calorimeter, *J. Inst.* 17 (2022) C03015. <https://doi.org/10.1088/1748-0221/17/03/C03015>.
- [3] J. Lu, L. Zhao, J. Qin, H. Xu, J. Cao, S. Liu, Q. An, Readout Electronics Prototype of TOF Detectors in CEE of HIRFL, *IEEE Trans. Nucl. Sci.* 68 (2021) 1976–1983. <https://doi.org/10.1109/TNS.2021.3093544>.
- [4] S. Pourashraf, A. Gonzalez-Montoro, J.Y. Won, M.S. Lee, J.W. Cates, Z. Zhao, J.S. Lee, C.S. Levin, Scalable electronic readout design for a 100 ps coincidence time resolution TOF-PET system, *Phys. Med. Biol.* 66 (2021) 085005. <https://doi.org/10.1088/1361-6560/abf1bc>.
- [5] S. Pourashraf, A. Gonzalez-Montoro, M.S. Lee, J.W. Cates, J.Y. Won, J.S. Lee, C.S. Levin, Investigation of Electronic Signal Processing Chains for a Prototype TOF-PET System with 100 ps Coincidence Time Resolution, *IEEE Trans. Radiat. Plasma Med. Sci.* (2021) 1–1. <https://doi.org/10.1109/TRPMS.2021.3124756>.
- [6] M. Grujic, I. Verbauwhede, TROT: A Three-Edge Ring Oscillator Based True Random Number Generator With Time-to-Digital Conversion, *IEEE Trans. Circuits Syst. I.* (2022) 1–14. <https://doi.org/10.1109/TCSI.2022.3158022>.
- [7] Y.-Y. Hu, X. Lin, S. Wang, J.-Q. Geng, Z.-Q. Yin, W. Chen, D.-Y. He, W. Huang, B.-J. Xu, G.-C. Guo, Z.-F. Han, Quantum random number generation based on spontaneous Raman scattering in standard single-mode fiber, *Opt. Lett.* 45 (2020) 6038. <https://doi.org/10.1364/OL.409187>.
- [8] T. Talala, V.A. Kaikkonen, P. Keranen, J. Nikkinen, A. Harkonen, V.G. Savitski, S. Reilly, ukasz Dziechciarzyk, A.J. Kemp, M. Guina, A.J. Makynen, I. Nissinen, Time-Resolved Raman Spectrometer With High Fluorescence Rejection Based on a CMOS SPAD Line Sensor and a 573-nm Pulsed Laser, *IEEE Trans. Instrum. Meas.* 70 (2021) 1–10. <https://doi.org/10.1109/TIM.2021.3054679>.
- [9] J. Holma, I. Nissinen, J. Nissinen, J. Kostamovaara, Characterization of the Timing Homogeneity in a CMOS SPAD Array Designed for Time-Gated Raman Spectroscopy, *IEEE Trans. Instrum. Meas.* 66 (2017) 1837–1844. <https://doi.org/10.1109/TIM.2017.2673002>.
- [10] J. Hu, B. Liu, R. Ma, M. Liu, Z. Zhu, A  $32 \times 32$ -Pixel Flash LiDAR Sensor With Noise Filtering for High-Background Noise Applications, *IEEE Trans. Circuits Syst. I.* 69 (2022) 645–656. <https://doi.org/10.1109/TCSI.2020.3048367>.
- [11] W. Zhang, R. Ma, X. Wang, H. Zheng, Z. Zhu, A High Linearity TDC With a United-Reference Fractional Counter for LiDAR, *IEEE Trans. Circuits Syst. I.* 69 (2022) 564–572. <https://doi.org/10.1109/TCSI.2020.3045731>.
- [12] R. Machado, J. Cabral, F.S. Alves, Recent Developments and Challenges in FPGA-Based Time-to-Digital Converters, *IEEE Trans. Instrum. Meas.* 68 (2019) 4205–4221.

<https://doi.org/10.1109/TIM.2019.2938436>.

[13] K. Cui, X. Li, A High-Linearity Vernier Time-to-Digital Converter on FPGAs With Improved Resolution Using Bidirectional-Operating Vernier Delay Lines, *IEEE Trans. Instrum. Meas.* 69 (2020) 5941–5949. <https://doi.org/10.1109/TIM.2019.2959423>.

[14] Q. Shen, S. Liu, B. Qi, Q. An, S. Liao, P. Shang, C. Peng, W. Liu, A 1.7 ps Equivalent Bin Size and 4.2 ps RMS FPGA TDC Based on Multichain Measurements Averaging Method, *IEEE Trans. Nucl. Sci.* 62 (2015) 947–954. <https://doi.org/10.1109/TNS.2015.2426214>.

[15] J. Wu, Z. Shi, The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay, in: 2008 IEEE Nuclear Science Symposium Conference Record, IEEE, Dresden, Germany, 2008: pp. 3440–3446. <https://doi.org/10.1109/NSSMIC.2008.4775079>.

[16] J. Wang, S. Liu, L. Zhao, X. Hu, Q. An, The 10-ps Multitime Measurements Averaging TDC Implemented in an FPGA, *IEEE Trans. Nucl. Sci.* 58 (2011) 2011–2018. <https://doi.org/10.1109/TNS.2011.2158551>.

[17] R. Szplet, D. Sondej, G. Grzeda, High-Precision Time Digitizer Based on Multiedge Coding in Independent Coding Lines, *IEEE Trans. Instrum. Meas.* 65 (2016) 1884–1894. <https://doi.org/10.1109/TIM.2016.2555218>.

[18] W. Xie, H. Chen, D.D.-U. Li, Efficient time-to-digital converters in 20 nm FPGAs with wave union methods (the title was suggested by Reviewer 4, different from the original one 'Are wave union methods suitable for 20 nm FPGA-based time-to-digital converters'), *IEEE Trans. Ind. Electron.* (2021) 1–1. <https://doi.org/10.1109/TIE.2021.3053905>.

[19] L. Leuenberger, D. Amiet, T. Wei, P. Zbinden, An FPGA-based 7-ENOB 600 MSample/s ADC without any External Components, in: The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ACM, Virtual Event USA, 2021: pp. 240–250. <https://doi.org/10.1145/3431920.3439287>.

[20] B. Wu, Y. Wang, Q. Cao, Z. Li, X. Li, X. Zhou, Y. Hu, Z. Wang, M. Shao, J. Liu, C. Li, Z. Zhao, Design of Time-to-Digital Converters for Time-Over-Threshold Measurement in Picosecond Timing Detectors, *IEEE Trans. Nucl. Sci.* 68 (2021) 470–476. <https://doi.org/10.1109/TNS.2021.3060069>.

[21] Y. Wang, C. Liu, A 3.9 ps Time-Interval RMS Precision Time-to-Digital Converter Using a Dual-Sampling Method in an UltraScale FPGA, *IEEE Trans. Nucl. Sci.* 63 (2016) 2617–2621. <https://doi.org/10.1109/TNS.2016.2596305>.

[22] UltraScale Architecture Configurable Logic Block User Guide (UG574), (2017) 58.

[23] 7 Series FPGAs Configurable Logic Block User Guide (UG474), (2016) 74.

[24] H. Chen, D.D.-U. Li, Multichannel, Low Nonlinearity Time-to-Digital Converters Based on 20 and 28 nm FPGAs, *IEEE Trans. Ind. Electron.* 66 (2019) 3265–3274. <https://doi.org/10.1109/TIE.2018.2842787>.

[25] F. Kaes, R. Kanan, B. Hochet, M. Declercq, New Encoding Scheme For High-speed Flash ADC's, (n.d.) 4.

[26] C. Liu, 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.* 62 (2015) 773–783. <https://doi.org/10.1109/TNS.2015.2421319>.

[27] Y. Wang, J. Kuang, C. Liu, Q. Cao, A 3.9-ps RMS Precision Time-to-Digital Converter Using Ones-Counter Encoding Scheme in a Kintex-7 FPGA, *IEEE Trans. Nucl. Sci.* 64 (2017) 2713–2718. <https://doi.org/10.1109/TNS.2017.2746626>.

- [28] Z. Song, Y. Wang, J. Kuang, A 256-channel, high throughput and precision time-to-digital converter with a decomposition encoding scheme in a Kintex-7 FPGA, *J. Inst.* 13 (2018) P05012–P05012. <https://doi.org/10.1088/1748-0221/13/05/P05012>.
- [29] Y. Wang, W. Xie, H. Chen, D.D.-U. Li, Multichannel time-to-digital converters with automatic calibration in Xilinx Zynq-7000 FPGA devices, *IEEE Trans. Ind. Electron.* (2021) 1–1. <https://doi.org/10.1109/TIE.2021.3111563>.
- [30] Xilinx, UltraScale Architecture Libraries Guide, (2020). <https://docs.xilinx.com/v/u/2020.1-English/ug974-vivado-ultrascale-libraries>.
- [31] J. Kalisz, Review of methods for time interval measurements with picosecond resolution, *Metrologia.* 41 (2004) 17–32. <https://doi.org/10.1088/0026-1394/41/1/004>.
- [32] J. Wu, Several Key Issues on Implementing Delay Line Based TDCs Using FPGAs, *IEEE Trans. Nucl. Sci.* 57 (2010) 1543–1548. <https://doi.org/10.1109/TNS.2010.2045901>.
- [33] Xilinx, ZCU104 Evaluation Board User Guide (UG1267), (2018). [https://www.xilinx.com/support/documents/boards\\_and\\_kits/zcu104/ug1267-zcu104-eval-bd.pdf](https://www.xilinx.com/support/documents/boards_and_kits/zcu104/ug1267-zcu104-eval-bd.pdf).
- [34] J. Wu, Uneven bin width digitization and a timing calibration method using cascaded PLL, in: 2014 19th IEEE-NPSS Real Time Conference, IEEE, Nara, Japan, 2014: pp. 1–4. <https://doi.org/10.1109/RTC.2014.7097534>.
- [35] E. Bayer, M. Traxler, A High-Resolution ( $\sim 10$  ps RMS) 48-Channel Time-to-Digital Converter (TDC) Implemented in a Field Programmable Gate Array (FPGA), *IEEE Trans. Nucl. Sci.* 58 (2011) 1547–1552. <https://doi.org/10.1109/TNS.2011.2141684>.
- [36] Y. Wang, X. Zhou, Z. Song, J. Kuang, Q. Cao, A 3.0-ps rms Precision 277-MSamples/s Throughput Time-to-Digital Converter Using Multi-Edge Encoding Scheme in a Kintex-7 FPGA, *IEEE Trans. Nucl. Sci.* 66 (2019) 2275–2281. <https://doi.org/10.1109/TNS.2019.2938571>.
- [37] N. Lusardi, F. Garzetti, N. Corna, R.D. Marco, A. Geraci, Very High-Performance 24-Channels Time-to-Digital Converter in Xilinx 20-nm Kintex UltraScale FPGA, in: 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), IEEE, Manchester, United Kingdom, 2019: pp. 1–4. <https://doi.org/10.1109/NSS/MIC42101.2019.9059958>.
- [38] P. Kwiatkowski, D. Sondej, R. Szplet, Bubble-Proof Algorithm for Wave Union TDCs, *Electronics.* 11 (2021) 30. <https://doi.org/10.3390/electronics11010030>.
- [39] S. Tancock, J. Rarity, N. Dahnoun, The Wave-Union Method on DSP Blocks: Improving FPGA-Based TDC Resolutions by 3x With a 1.5x Area Increase, *IEEE Trans. Instrum. Meas.* 71 (2022) 1–11. <https://doi.org/10.1109/TIM.2022.3141753>.