University of
Strathclyde
Glasgow

UNIVERSITY OF STRATHCLYDE

Faculty of Engineering

Development of Low-cost Field-Programmable Gate Array based Time-to-Digital Converters for Time-resolved Measurements

by

Yu Wang

Abstract

Time-to-digital converters (TDCs) are high-precision stopwatches that convert a time interval (TI) into a digital code. They measure TIs with resolutions from nanoseconds (ns) to femtoseconds (fs). Due to their high resolutions, they are key components in time-of-flight (ToF) applications such as light detection and ranging (LiDAR) and positron emission tomography (PET). They are also used in fluorescence lifetime imaging microscopy (FLIM), quantum random number generation (QRNG), high-energy physics, etc.

For digital TDCs, both field programmable gate array (FPGA)-TDCs and application-specific integrated circuit (ASIC)-TDCs are capable of achieving picosecond (ps)-level resolutions, benefiting from advances in complementary metal-oxide semiconductor (CMOS) manufacturing technologies. However, FPGA-TDCs have shorter developing cycles and lower developing costs than ASIC-TDCs. Hence, FPGA-TDCs are more appropriate for rapid prototype verification. FPGA-TDCs are usually composed of coarse and fine counters to achieve a wide measurement range and high resolution simultaneously. A coarse counter can be easily implemented using a clock-driven counter. Hence, most research focuses on the architecture and calibration methods for fine-time measurements. For a fine counter, the primary parameter is the resolution (also known as the least significant bit, LSB). It is the minimal TI that can be detected. Ideally, all time bins in a TDC should have the same bin width. However, they are not uniform due to the imperfect CMOS manufacturing and clock skews. The variations of bin widths are characterised by differential nonlinearity (DNL) and integral nonlinearity (INL). Besides, measurement results fluctuate for a fixed TI due to jitters and quantization errors. Hence, precision or root-mean-square (RMS) precision is also a parameter of concern for TDCs.

This thesis proposes four innovative FPGA-TDCs with novel architectures and calibration methods, aiming to enhance TDCs' resolution, linearity and hardware utilisation efficiency. The first design proposes an automatic calibration architecture implemented in a Zynq-7000 system on chip (SoC). This design uses the programmable

logic (PL) in the SoC to build a 16-channel TDC and uses the processing system (PS) to calibrate all TDC channels in the PL. This TDC offers a resolution of 9.83 ps with good uniformity, achieving an averaged peak-to-peak DNL (DNL_{pk-pk}) of 0.38 LSB and an averaged peak-to-peak INL (INL_{pk-pk}) of 0.63 LSB.

The second design proposes several new methods and architectures, with the sampling matrix architecture being the most significant contribution, dramatically enhancing the resolution of the Gray-code oscillator (GCO)-TDC with low hardware utilisation. Besides, the virtual bin calibration method is also proposed and hardware-implemented, aiming at high linearity and variable resolutions. With these innovations, this design can offer a 20.97~80.45 ps resolution with a better than 0.18 LSB averaged DNL_{pk-pk} in the 16-channel TDC implemented in a Zynq UltraScale+ MPSoC. Simultaneously, each channel only uses 456 look-up tables (LUTs) and 368 D-type flip-flops (DFFs), indicating substantial potential for multi-channel applications. Moreover, this design is also implemented in Kintex-UltraScale and Virtex-7 FPGAs, showcasing its universality.

The third design combines the wave union (WU) and dual-sampling methods and proposes a bidirectional encoder and a manually-routed WU launcher. The WU launcher generates a four-transition wave pattern to enhance the resolution. The bidirectional encoder is then designed to encode the four transitions in real time. Combined with the sub-tapped-delay line (sub-TDL) method, the proposed TDC implemented in a Zynq UltraScale+ MPSoC achieves a 0.46 ps ultra-high resolution with a 4.44 ns dead time.

The fourth design is a two-stage interpolation TDC consisting of a Vernier GCO-TDC (VGCO-TDC) and a TDL-TDC. This architecture uses the TDL-TDC to measure the overtaking residue from the VGCO-TDC. Hence, the TDL-TDC only needs to cover the resolution of the VGCO-TDC, significantly reducing the hardware utilisation of the TDL-TDC. When implemented in a Kintex-UltraScale FPGA, the TDC can achieve an average resolution of 4.57 ps, only consuming 440 LUTs and 570 DFFs. Moreover, this design is also implemented in a Virtex-7 FPGA, showcasing its universality.

Acknowledgements

The journey through my PhD is particularly challenging over the last four years, especially during the COVID-19 pandemic. I would like to express my sincere appreciation to everyone who encouraged and supported me. Without your support, completing this thesis would be impossible.

First, I am deeply grateful to my supervisor, Dr. David Li, who is always approachable. Without his constant encouragement, I cannot persist in my PhD. His guidance is also crucial in enhancing my writing and communication skills, refining my ideas, and addressing design issues.

Second, I am profoundly thankful to the China Scholarship Council and the University of Strathclyde for providing the funding to cover my costs and the necessary resources to conduct my research. Their financial support allowed me to focus my research, and the equipment facilitated my experimental set-up.

Third, I would like to acknowledge my parents for their endless love and unconditional trust, which makes my PhD journal possible. My sincere appreciation also goes to my fiancée, Lexi Tian. Her encouragement and company guide me towards the light during the darkest time.

Last, I am forever indebted to my colleagues and friends, including Dr. Wujun Xie, Dr. Dongxiao, Dr. Ming Zhang, Dr. Quan Wang, Dr. Ziao Jiao, Dr. Yang Huang, ZhenYa Zang, Miaomiao Chang, Yifei Wan, Henqing Tian, Xiang Li, YinHua Liu, Xi Chen, Xinda Li, Yulin Liu, for their understanding and selfless assistance. You all enrich my life in Glasgow, making it vibrant and memorable.

List of Figures

List of Tables

List of Abbreviations

Auto. Cali.	**Automatic calibration**
ADC	**Analog-to-digital converter**
ALU	**Arithmetic logic unit**
ASIC	**Application-specific integrated circuit**
Asyn. Output BRAM	**Asynchronous output block random access memory**
AXI	**Advanced eXtensible interface**
BCFC	**Bin-width calibration factor calculation**
BES III	**Beijing Spectrometer**
BRAM	**Block random access memory**
C&C BRAM	**Compensation and calibration block random access memory**
C&C core	**Compensation and calibration core**
Calc.&Output	**Calculation and output**
CC	**Coincidence circuit**
CCS	**Coarse_clk_sync**
CFC	**Compensation factor calculation**
CMOS	**Complementary metal-oxide semiconductor**
CMS	**Compact muon solenoid**
comp.&cali. factor	**Compensation and calibration factor**
DESPEC	**Decay spectroscopy**
DFF	**D-type flip-flop**
DNL	**Differential nonlinearity**
DNLpk-pk	**Peak-to-peak differential nonlinearity**
DSP	**Digital signal processing**
dToF	**Direct time-of-flight**
EDA	**Electronic design automation**
ETL	**Endcap timing layer**
ETROC	**Endcap timing readout chip**
FLIM	**Fluorescence lifetime imaging microscopy**
FPGA	**Field programmable gate array**
fs	**Femtosecond**
GCO	**Gray-code oscillator**
GRO	**Gated ring oscillator**
HDL	**Hardware description language**
Histo. BRAM	**Histogram block random access memory**
INL	**Integral nonlinearity**
INLpk-pk	**Peak-to-peak integral nonlinearity**
IP	**Intellectual property**
ISA	**Input_shaper_start**
LGAD	**Low-gain avalanche detector**
LHC	**Large hadron collider**

LiDAR	**Light detection and ranging**
LoR	**Line of response**
LSB	**Least significant bit**
LSPR	**Large-scale parallel routing**
LUT	**Look-up table**
MBD	**Maximum bubble depth**
MMCM	**Mixed-mode clock manager**
MPGRO	**Multi-path gated ring oscillator**
MUX	**Multiplexer**
ns	**Nanosecond**
NUMMP	**Nonuniform monotonic multiphase**
Para. Core	**Parameter core**
PC	**Personal computer**
PET	**Positron emission tomography**
PL	**Programmable logic**
PLL	**Phase-locked loop**
ps	**Picosecond**
PS	**Processing system**
QRNG	**Quantum random number generation**
RMS	**Root mean square**
RO	**Ring oscillator**
SCDT	**Static code density test**
SI	**International System of Units**
SoC	**System on chip**
SPAD	**Single photon avalanche diode**
SRS	**Stanford Research System**
TDC	**Time-to-digital converter**
TDL	**Tapped-delay line**
TI	**Time interval**
TIM	**Time interval meter**
ToF	**Time-of-flight**
UART	**Universal asynchronous receiver-transmitter**
USB	**Universal serial bus**
VBCM	**Virtual bin calibration method**
VGCO-TDC	**Vernier Gray code oscillator time-to-digital converter**
VRO	**Vernier ring oscillator**
VTDL	**Vernier tapped-delay line**
WU	**Wave union**

List of Publications

Peer-Reviewed Journal Articles

1. **Y. Wang**, W. Xie, H. Chen, and D. D.-U. Li, ‘Multichannel Time-to-Digital Converters With Automatic Calibration in Xilinx Zynq-7000 FPGA Devices’, *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9634–9643, Sep. 2022, doi: 10.1109/TIE.2021.3111563.

2. **Y. Wang**, W. Xie, H. Chen, and D. D.-U. Li, ‘Low-Hardware Consumption, Resolution-Configurable Gray Code Oscillator Time-to-Digital Converters Implemented in 16 nm, 20 nm, and 28 nm FPGAs’, *IEEE Trans. Ind. Electron.*, vol. 70, no. 4, pp. 4256–4266, Apr. 2023, doi: 10.1109/TIE.2022.3174299.

3. **Y. Wang**, W. Xie, H. Chen, and D. Day-Uei Li, ‘High-resolution time-to-digital converters (TDCs) with a bidirectional encoder’, *Measurement*, vol. 206, p. 112258, Jan. 2023, doi: 10.1016/j.measurement.2022.112258.

4. **Y. Wang**, W. Xie, H. Chen, C. Pei, and D. D.-U. Li, ‘A Two-Stage Interpolation Time-to-Digital Converter Implemented in 20 and 28 nm FGPAs’, *IEEE Trans. Ind. Electron.*, pp. 1–11, 2024, doi: 10.1109/TIE.2024.3370941.

5. W. Xie, **Y. Wang**, H. Chen, and D. D.-U. Li, ‘128-Channel High-Linearity Resolution-Adjustable Time-to-Digital Converters for LiDAR Applications: Software Predictions and Hardware Implementations’, *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 4264–4274, Apr. 2022, doi: 10.1109/TIE.2021.3076708.

Chapter 1 Introduction

In this chapter, Section 1.1 introduces the research background and Section 1.2 introduces the research aim. Besides, the contributions and outline of this thesis are introduced in Sections 1.3 and 1.4, respectively.

1.1 Background

Time is one of the seven fundamental physical quantities in the International System of Units (SI), with the second serving as its foundational unit [1]. Historically, a second was defined as 1/86400 of a day, based on the Earth's rotation. However, irregularities in Earth's rotation led to the adoption of the atomic standard in 1967. Now, it is defined as the duration of 9192631770 periods of radiation corresponding to the transition between two hyperfine levels of the ground state of the cesium-133 atom [2]. The passage of time is commonly measured either by fixed time delays, such as for example an hourglass, or by monitoring an oscillation at a fixed frequency, such as the swing of a pendulum in a pendulum clock. The definition of a second given above is based on a microwave oscillation that can be measured with state-of-the-art accuracy. This thesis explores the use of digital electronic delays and oscillators to measure time intervals with picosecond (ps) precision.

However, the reaction and response occur within an extremely short time interval (TI), such as within the ps-level TI, in scientific and industrial TI-measured applications [3], [4], [5]. Hence, these applications require TI meters (TIMs) with an extremely high resolution. Time-to-digital converters (TDCs) are typical TIMs, converting a TI between two electrical pulses into a digital code. Moreover, TDCs are capable of offering resolutions ranging from nanoseconds (ns) to femtoseconds (fs) [6]. Hence, they are crucial components in both scientific and industrial TI-measured applications.

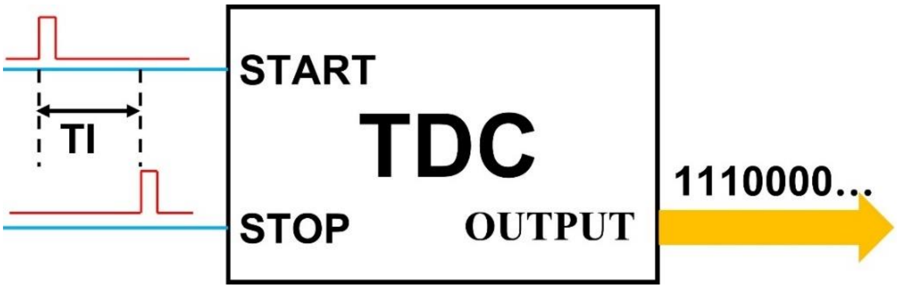


Figure 1.1 Block diagram of TDC measurement.

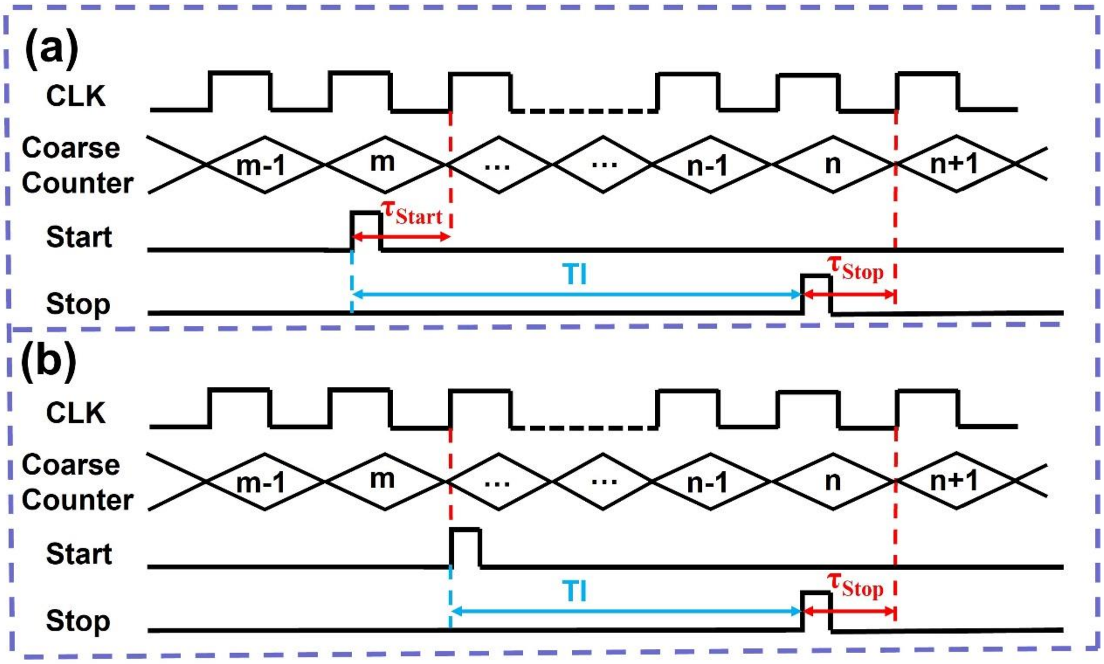


Figure 1.2 Measurement principles of (a) asynchronous Nutt-TDC and (b) synchronous Nutt-TDC.

1.1.1 TDCs and Xilinx FPGAs

Figure 1.1 shows the block diagram of TDC measurement. The TDC shown in Fig. 1.1 has two separate inputs: START and STOP, and the measured TI is the duration between rising edges of pulses input to the START and STOP, respectively (TI shown in Fig. 1.1). TDCs usually use the Nutt method [7], which consists of a coarse counter and a fine counter, to achieve high resolutions and wide measurement ranges simultaneously [8]. The measurement principle of the Nutt-TDC is shown in Fig. 1.2. Fig. 1.2a shows the measurement principle when both the start and stop pulses are asynchronous with the coarse-counting clock (CLK in Fig. 1.2a). In this scenario, the measured TI is calculated as:

$$TI = \left(\tau_{start} - \tau_{stop}\right) + (n - m) \times T, \tag{1.1}$$

where T is the period of the coarse-counting clock, m and n are coarse codes output from the coarse counters, and τ_{start} and τ_{stop} are TIs between rising edges of corresponding pulses and subsequent coarse-counting clocks. However, as shown in Fig. 1.2b, the start pulse can also be synchronous with the coarse-counting clock (CLK in Fig. 1.2b). Hence, the measured TI can also be calculated as:

$$TI = (n - m) \times T - \tau_{stop}. \tag{1.2}$$

In Fig. 1.2a, τ_{start} and τ_{stop} are respectively measured. Therefore, two TDC channels (Start-channel and Stop-channel in Fig. 1.3a) are required in the asynchronous Nutt-TDC shown in Fig. 1.3a. However, only τ_{stop} is measured in the synchronous Nutt-TDC shown in Fig. 1.2b. Hence, as shown in Fig. 1.3b, only the stop-channel is required for the synchronous Nutt-TDC. The synchronous Nutt-TDC measure the arrival time of an input signal relative to the coarse-counting clock, making it increasingly popular in recent years due to its suitability for capturing the arrival times of successive multiple inputs. A clock-driven counter can easily achieve the coarse counter of the Nutt-TDC. Therefore, most research focuses on the fine counter.

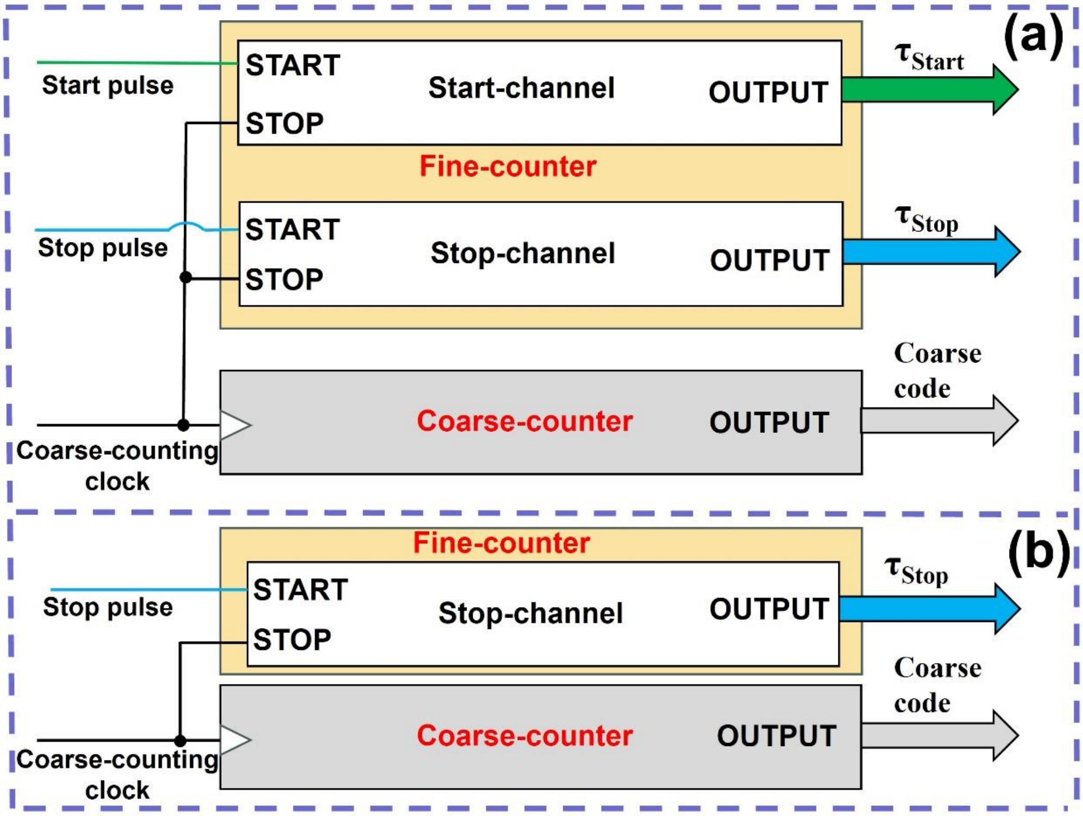


Figure 1.3 Block diagrams of (a) asynchronous and (b) synchronous Nutt-TDCs.

Methods of fine counters can be roughly classified as “analog” and “digital” [8]. Therefore, the corresponding TDCs are named “analog-TDCs” and “digital-TDCs”. The prevalent architecture of analog-TDCs is the double-conversion TDC, as shown in Fig. 1.4. This architecture first converts the measured TI into a voltage by charging a capacitor (*C* in Fig. 1.4) with a constant current (*I* in Fig. 1.4). The output voltage of the capacitor is then converted into a digital code by an analog-to-digital converter (ADC) [9]. Double-conversion TDCs can achieve a high measurement resolution. For example, in 1994, Kalisz *et al.* designed a double-conversion TDC with a 3 ps resolution [10]. In 2015, Kim *et al.* proposed a hybrid-domain two-step double-conversion TDCs with a 0.63 ps resolution [11]. However, this kind of TDCs suffers from high power consumption, large silicon area and significant temperature drift [12]. Hence, digital-TDCs are more widely used in industrial and scientific applications.

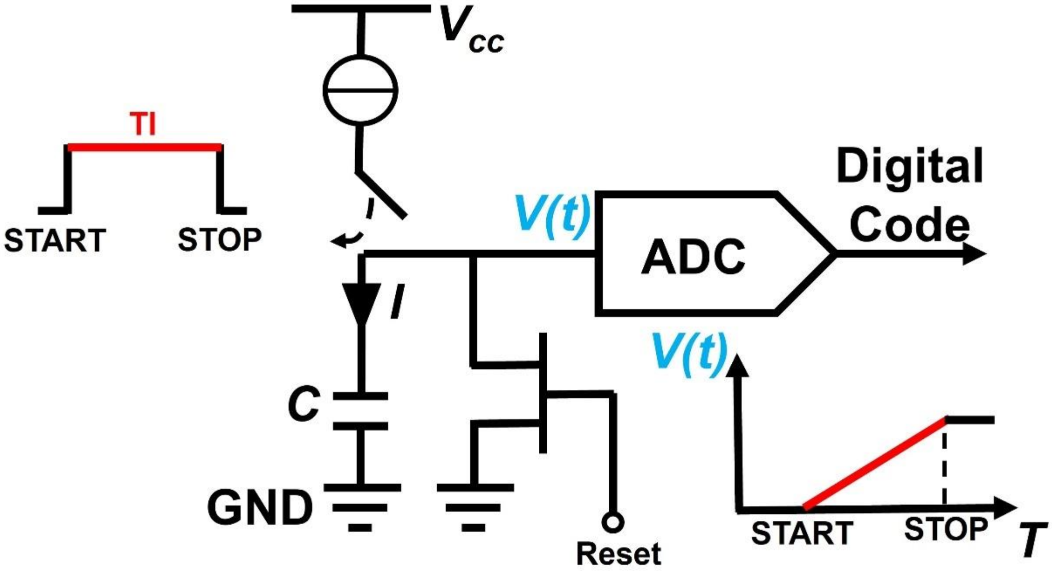


Figure 1.4 Architecture of double-conversion analog-TDCs.

Digital-TDCs (architectures and enhancing methods of digital-TDCs will be reviewed in Chapter 2) can be implemented on both application-specific integrated circuits (ASICs) and field programmable gate arrays (FPGAs). Generally, ASIC-TDCs exhibit better performance compared with FPGA-TDCs due to customised placing and routing strategies. However, FPGA-TDCs feature shorter development cycles and lower development costs. Hence, FPGA-TDCs are more appropriate when rapid prototype verification is required in industrial and scientific applications.

For commercial FPGAs, AMD Xilinx (hereafter referred to simply as Xilinx) [13] and Intel Altera (hereafter referred to simply as Altera) [14] are the leading two manufacturers, each offering several product series. For example, Xilinx has Kintex and Virtex series products, while Altera has Cyclone and Stratix series products [13], [14]. Despite each series having its own features, these products have similar architectures, containing configurable logic blocks (CLBs in Xilinx FPGAs and LBs in Altera FPGAs), programmable routing resources and I/O blocks. Among these components, CLBs/LBs are responsible for configurable logical and arithmetic functions, programmable routing resources are used to connect different CLBs/LBs, and I/O blocks are for chip-to-chip connections. In this thesis, all innovative designs, including architectures and enhancing methods, are implemented in Xilinx FPGAs. Hence, Xilinx FPGAs are briefly introduced below.

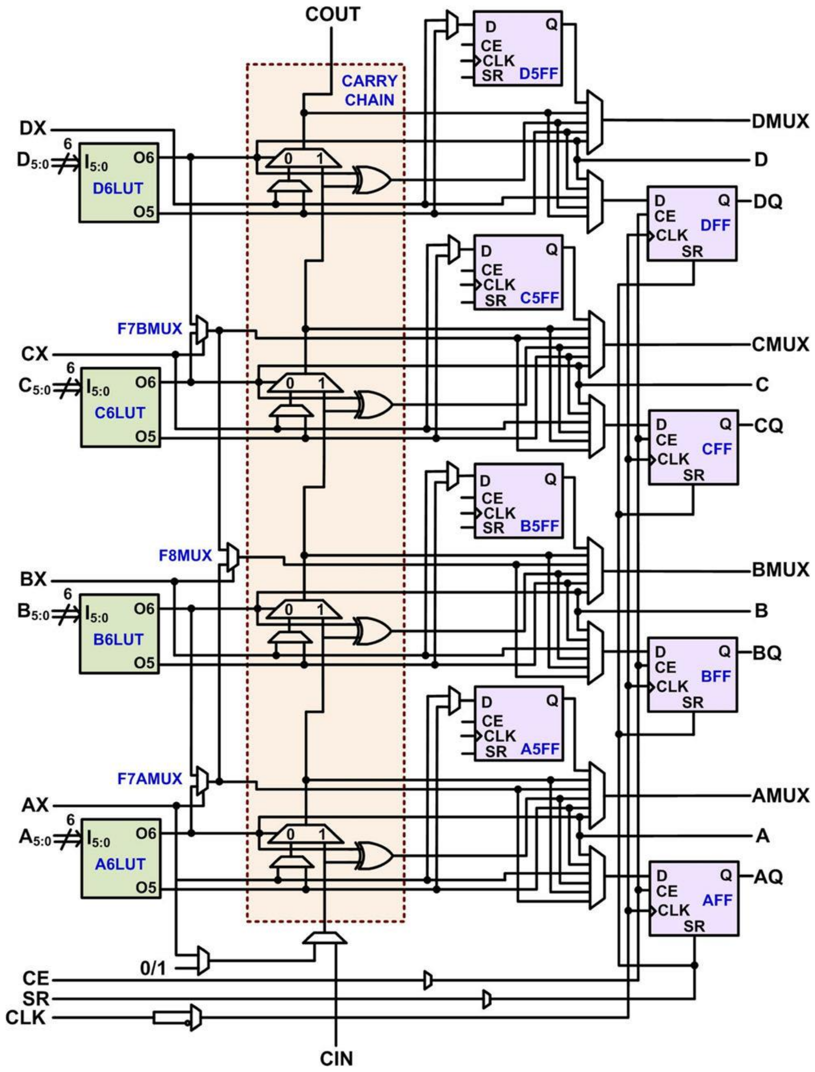


Figure 1.5 Diagram of the 7-series FPGA's slice.

CLBs are fundamental components in Xilinx FPGAs, serving as highly flexible and reconfigurable resources for implementing a wide range of digital logical and arithmetic functions. For Xilinx 7-series FPGAs such as Kintex-7 and Virtex-7 FPGAs, each CLB consists of two slices, with each slice containing four look-up tables (LUTs), a 4-tap carry chain and eight D-type flip-flops (DFFs) [15], as shown in Fig 1.5. The slices in the UltraScale and UltraScale+ series FPGAs, such as Kintex-UltraScale FPGAs, are structured similarly to the slices in 7-series FPGAs. However, benefiting from more advanced manufacturing technologies, two independent slices within a CLB in 7-series FPGAs are merged into one slice in the CLB of UltraScale and UltraScale+ series

FPGAs, causing each slice to contain eight LUTs, an 8-tap carry chain and sixteen DFFs [16]. In the Xilinx 7-series and more advanced FPGAs, each LUT has six inputs and two outputs, as shown in Fig. 1.6 [17], [18]. All of these ports operate independently, allowing the LUT to be utilised in various configurations. For example, the 6-input LUT can operate as a single LUT with six inputs or be divided into two separate LUTs with five or fewer inputs. In the configuration where two separate LUTs are used with five or four inputs, at least one input port must be shared. In contrast, LUTs do not share any input ports when configured as two separate LUTs with three or two inputs. Unlike LUTs, which are mainly used for logical functions such as AND and OR, the carry chain is mainly used for arithmetic operations such as addition and subtraction. As shown in Fig. 1.5, the carry chain consists of multiplexers (MUXs) and XOR gates. In 7-series FPGAs, the carry chain in each slice has four taps (or eight taps in UltraScale and more advanced FPGAs), and vertically neighbouring carry chains can be connected together through the dedicated route for more than 4-bit arithmetic operations (or more than 8-bit arithmetic operations in UltraScale and more advanced FPGAs).

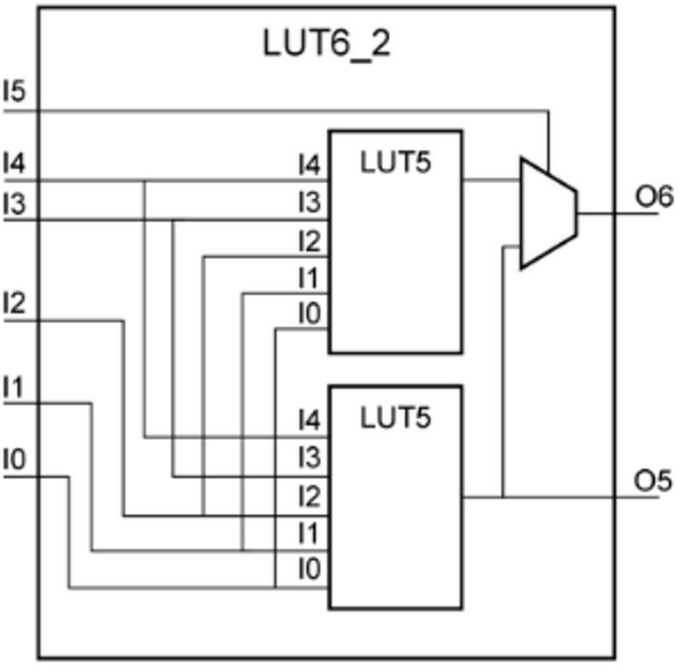


Figure 1.6 Diagram of the 6-input and 2-output LUT [18].

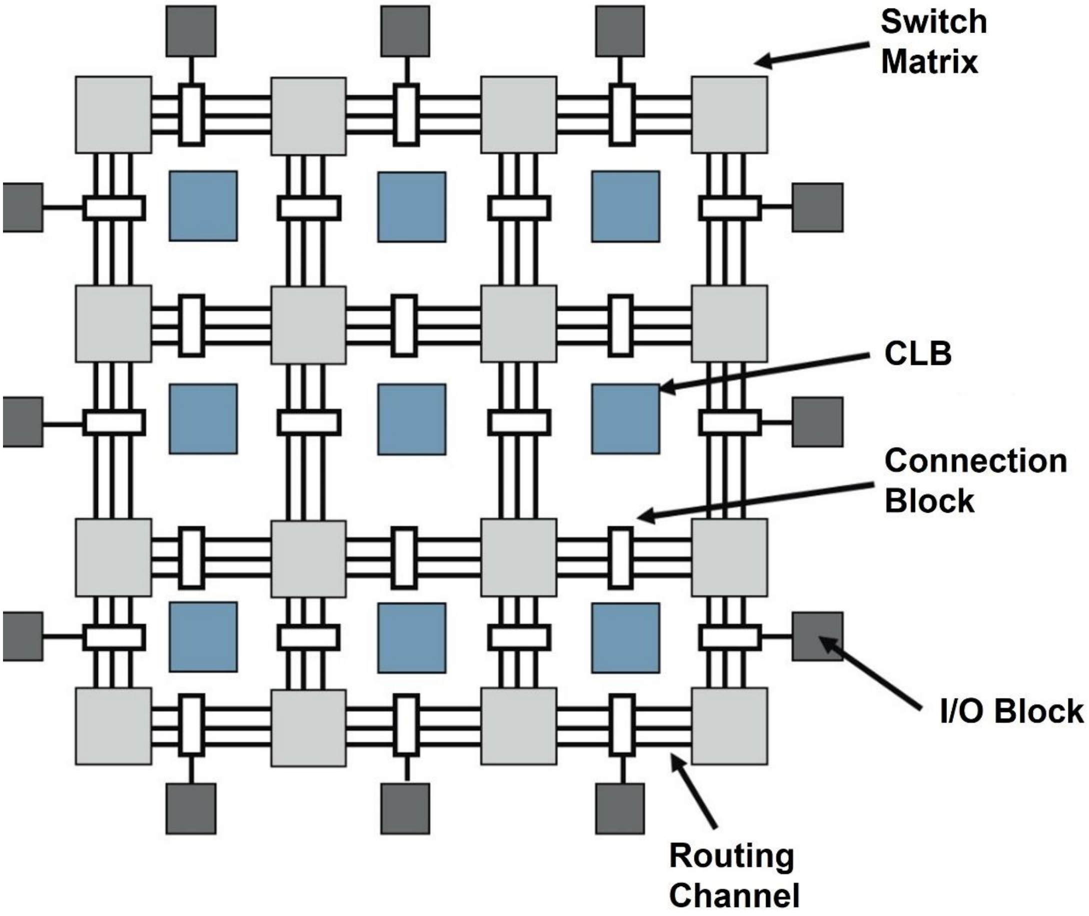


Figure 1.7 Island-style architecture [19].

Similar to CLBs, routing sources are also fundamental components in FPGAs. Routing in Xilinx FPGAs is divided into two types, including intra-slice routing and inter-slice routing [20]. Intra-slice routing has dedicated routing resources (the solid line shown in Fig. 1.5). However, inter-slice routing is implemented through switch matrices routing channels, and connection blocks, as shown in Fig. 1.7. The routing channels in FPGAs are wires with fixed tracks that link all the resources together. In contrast, both connection blocks and switch matrices are programmable. Connection blocks provide connections between neighbouring CLBs in both vertical and horizontal directions, while switch matrices establish connections between different routing channels. Since FPGAs have prefabricated routing resources, the electronic design automation (EDA) tool, such as Xilinx Vivado, must work within the framework of the architecture's resources, determining specific routing paths for connecting signals between CLBs and ensuring no excess connections are built [21]. Simultaneously, designer must also consider the routing congestions; otherwise, EDA tool may need to re-reroute the congested area, which is time-consuming.

In addition to CLBs and routing resources, FPGAs also have other components such as IDELAY, block random access memory (BRAM), ISERDESE, OSERDESE and so on [17], [18]. Each component serves a specific function. For example, IDELAY is used to insert a programmable delay into the input signal, BRAM is used as the on-chip memory, and ISERDESE and OSERDESE are serial-to-parallel and parallel-to-serial converters. These components are predefined with fixed functions, facilitating the implementation of designs.

1.1.2 TDC applications

Due to their high resolutions, TDCs play a pivotal role in TI-measurement applications such as light detection and ranging (LiDAR), fluorescence lifetime imaging microscopy (FLIM), and time-of-flight positron emission tomography (ToF-PET). They are also used in quantum random number generation (QRNG), high-energy physics, and slope-ADCs.

a. LiDAR

LiDAR is a remote sensing technology. As shown in Fig. 1.8a, a typical direct ToF LiDAR (dToF-LiDAR) system contains an emitter, a receiver, optical components, and a TDC. The “START” pulse marks the timestamp when the laser emitter begins illuminating the target with photons, while the “STOP” pulse indicates the timestamp when the receiver detects the photons reflected back from the target. Therefore, the TI between the “START” and the “STOP” pulses represents the ToF for photons to travel from the emitter, reach the target, and return to the receiver. In the dToF-LiDAR system, this TI is measured by a TDC to evaluate the distance to the target. LiDAR is widely used in robotics [22], [23], autonomous vehicles [24], [25], surveys [26], [27] and simultaneous localisation and mapping (SLAM) [28], [29]. In these applications, the measurement range of LiDAR varies from a few centimetres to several hundred meters (66.6-ps TI in dToF LiDAR corresponds to 1-centimeter distance). Hence, the TDC for dToF-LiDAR requires a wide measurement range [30].

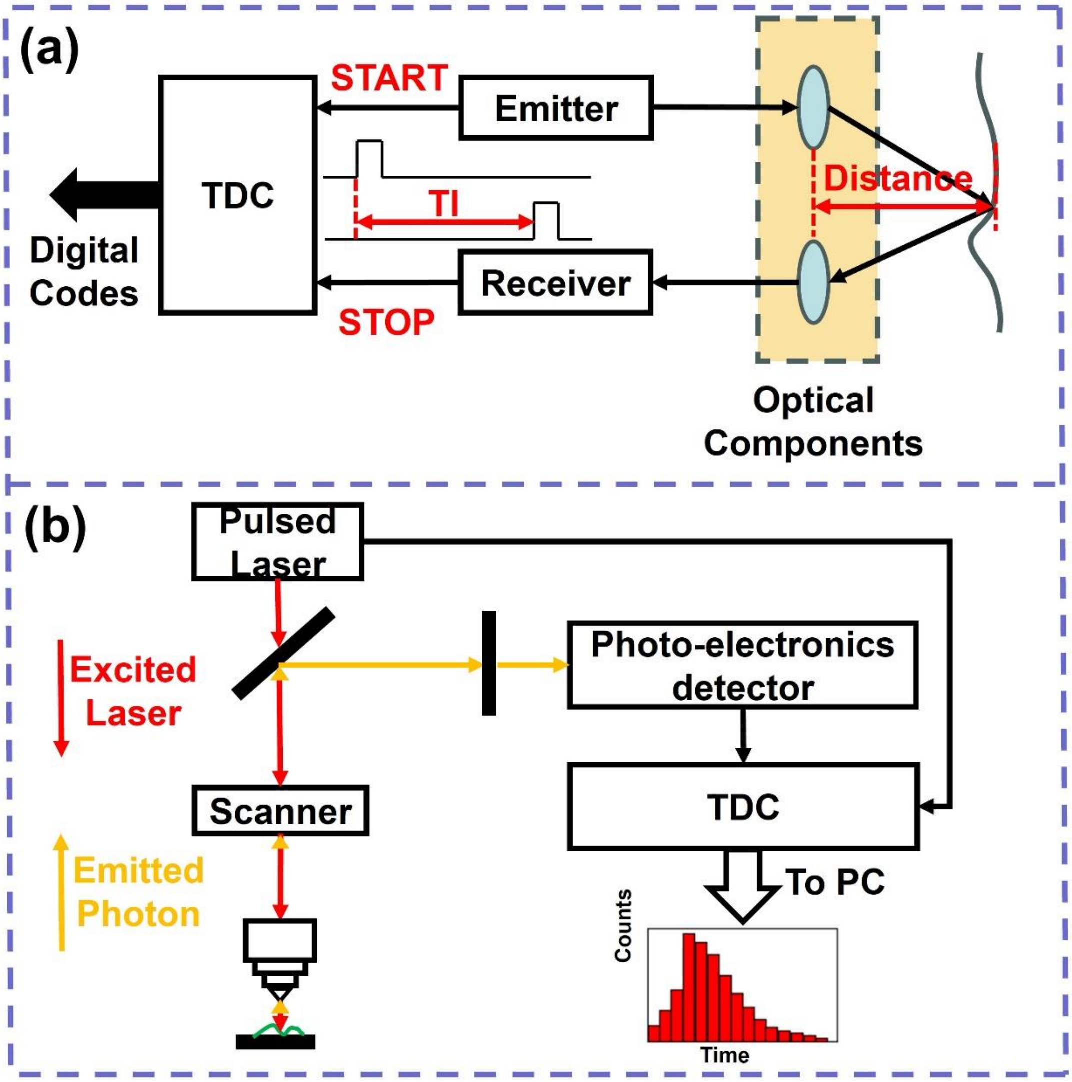


Figure 1.8 Diagrams of (a) a dToF LiDAR system and (b) a time-based FLIM system.

b. FLIM

Fluorescence is an optical phenomenon that occurs when a substance, termed fluorophores, absorbs photons with high energy and then re-emits photons at longer wavelengths. Fluorescence has many properties, including intensity, polarisation, lifetime and absorption/emission spectra. All of these properties can be applicable to specimen analysis. However, among them, the fluorescence lifetime, which is defined as the TI for the fluorophore to reduce from the maximum fluorescence intensity to 1/e (approximately 36.8%) of the maximum, is the most used metric. This is because the fluorescence lifetime is a relatively stable metric and cannot be influenced by the intensity of excitation light, fluorophore concentration and photobleaching [31]. Benefiting from the stability, FLIM is a powerful microscopy technique to monitor protein interaction, tension and folding [32]. Besides, FLIM can be used to monitor the microenvironment of living cells, such as pH [33], ion concentration (Ca2+, Cl−, K, Na...) [34], viscosity [35], [36], temperature [37], [38], and oxygen levels [39], [40]. As shown in Fig. 1.8b, a TDC is used to measure the decay time of fluorophores in a time-based FLIM system. In this system, a repetitive pulse laser is used to excite the fluorophores in the sample, causing them to emit light. The emitted photons are then captured by sensors, and over time, the number of captured photons is proportional to the fluorescence intensity. The decay of fluorescence intensity is recorded, and the fluorescence lifetime is determined by fitting the decay curve [41], [42]. The resolution of TDCs for FLIM varies from 50 ps to 120 ps [4].

c. ToF-PET

PET is an advanced medical technique that reveals the metabolic and physiological activity of tissues in the human body. In PET, a radionuclide-labelled substance is injected into the bloodstream. The radionuclide then emits positrons when traveling through body tissues. After that, positrons interact with electrons of neighbouring atoms, causing the annihilation of both particles (positrons and electrons) and emitting two gamma rays in opposite directions. A ring-shaped detector shown in Fig. 1.9 detects these gamma rays and records their line of response (LoR) and energy levels. This information is crucial for diagnosing and monitoring various diseases, including cancer, neurological disorders, and heart diseases [43]. Notably, the arrival time for the two gamma rays to reach the detector differs. Without the arrival time information, conventional PET (highlighted in red in Fig. 1.9) is limited by the sensitivity and signal-to-noise ratio (SNR), leading to degraded imaging quality [44]. To address these issues, ToF-PET (highlighted in green in Fig. 1.9) uses TDCs to measure the arrival time difference of two photons along the LoR. TDCs for the current commercial ToF-PETs (manufactured by Philips and GE) have time resolutions of several hundred picoseconds [45], [46], [47]. For ToF-PETs used in scientific research, TDCs have better time resolutions of around 50 ps [4].

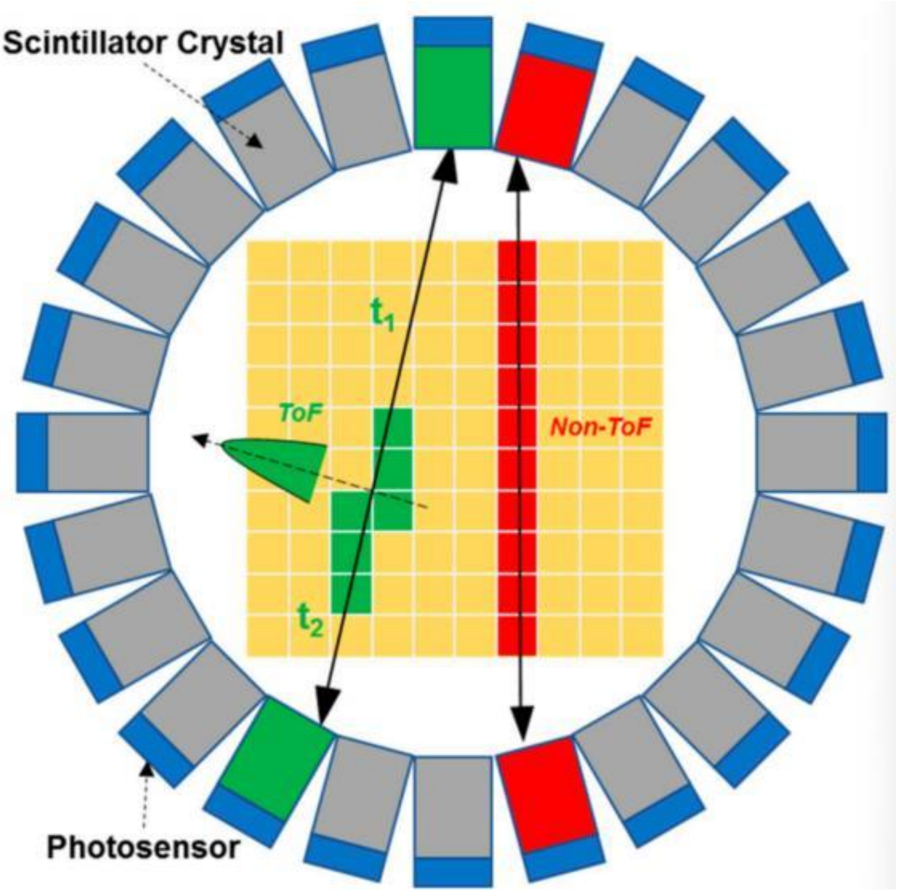


Figure 1.9 PET with (green) and without (red) ToF technique [44].

d. QRNG

High-quality random numbers are necessary for applications such as cryptography and secure communications [48], [49], [50]. However, conventional random number generators relying on random number generation algorithms can only generate pseudo-random numbers and hence predictable [50]. These pseudo-random numbers cause systems to have the potential to be attacked. Hence, QRNG, which extracts randomness from unpredictable quantum phenomena, attracts more and more attention from researchers due to the trustable randomness of quantum mechanics [50]. TDCs are crucial in time-of-arrival (ToA)-QRNG (also known as the optical QRNG) and are responsible for recording the timestamps of photons arriving at the detector. Benefitting from TDC's high precision and low dead time, TDC-based QRNG can generate true-random numbers with high efficiency and speed. For example, Khanmohammadi *et al.* proposed a 1Mb/s-generation-rate QRNG, successfully passing the "ENT", "STS", and "DIEHARD" randomness tests [51]. Similarly, Tontini *et al.* achieved a random bit generation rate of 7.3Mb/s, and the designed system passed the "NIST" randomness test [52].

e. High-energy physics

High-resolution TI measurement is also necessary in some high-energy physics experiments. Hence, TDCs are used in this experimental equipment [53], [54], [55]. For example, for the decay spectroscopy (DESPEC) detecting system "FATIMA", a TDC with a 24.6 ps resolution was designed to measure lifetimes of excited nuclear states through the method of delayed coincidence electronic fast timing [56]. A TDC with a 25 ps resolution was designed for the readout system of the Beijing Spectrometer (BES III) to ensure accurate particle identification [57]. And a low-power TDC was designed as a part of the readout ASIC, called the endcap timing readout chip (ETROC), to read out low-gain avalanche detectors (LGADs) for the compact muon solenoid (CMS) endcap timing layer (ETL) of high-luminosity large hadron collider (LHC) upgrade [58].

f. Slope-ADC

In addition to ToF and ToA measurements, TDCs can also be used in slope-ADCs, achieving high sampling rates in FPGAs. The FPGA-implemented slope-ADC (FPGA-ADC) was first proposed by Wu *et al.* in 2007, achieving a sample rate of 22.5MS/s with four external components (including resistors and capacitors) [59]. The concept of this design is shown in Fig. 1.10. The reference slope (highlighted in orange in Fig. 1.10) is synchronised with the system clock. A comparator inside FPGAs then compares the analog input (highlighted in purple) with the slope, outputting the square wave highlighted in green in Fig. 1.10. Finally, the TIs (t_r and t_f in Fig. 1. 10) are measured, and corresponding digital codes are output by the FPGA-TDC as the measurements for the input analog signal. The architecture of the FPGA-TDC has been further improved since it was proposed. For example, Homulle *et al.* achieved a sampling rate of 400MS/s per channel with just one external resistor [60]. Leuenberger *et al.* achieved a sampling rate of 1.2GS/s per channel, even without any other external components [61]. The fine resolutions and high sampling rates of implemented TDCs contribute to improving the resolutions and sampling rates of FPGA-TDCs. Besides, FPGA-TDCs can achieve a higher sampling rate using the multi-phase-interleaved method [60].

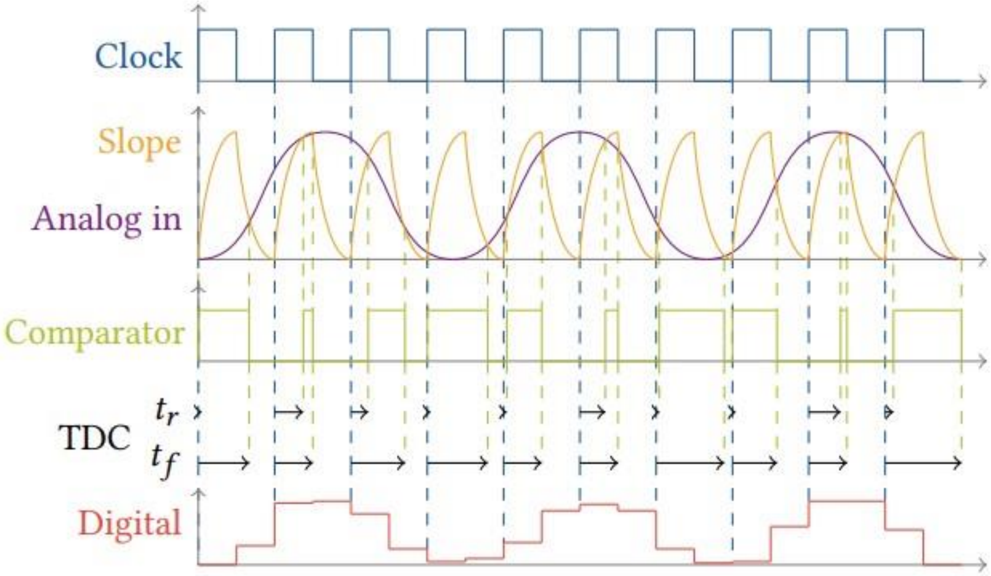


Figure 1.10 Principle of FPGA-ADCs [61].

1.2 Research aim

Although TDCs have recently garnered increasing focus from industry and research, the available commercial TDCs are still rare. Only a few manufacturers, such as Texas Instruments, Analog Devices, Renesas and Swabian Instruments, offer dedicated TDC chips and systems [62], [63], [64], [65]. However, these products are limited by resolutions, conversion speed, the number of channels and high prices, restricting their applications in rapid prototype verification. Hence, in this thesis, four low-cost FPGA-TDCs with outstanding performance and flexibility are proposed to meet different requirements.

The main goals of this PhD project are: 1) conducting a literature review of recently proposed TDCs' architectures and enhancing methods, 2) proposing new FPGA-TDCs' architectures and enhancing methods, targeting better resolutions, linearity and hardware utilisation efficiency, and 3) implementing linearity-enhancing methods in FPGAs to simplify online calibration and configuration of FPGA-TDCs for users who are unfamiliar with TDCs.

1.3 List of contributions

Contributions of the proposed FPGA-TDCs are summarised below:

1) Multi-channel Automatic Calibration TDCs in a Xilinx Zynq-7000 System on Chip (SoC) Device

- Proposed a single-step BRAM-based calibration method.
- Proposed an automatic calibration architecture and implemented it in a Xilinx Zynq-7000 SoC.

2) Resolution-configurable and low hardware utilisation TDCs in 16 nm, 20 nm and 28 nm FPGAs

- Proposed a sampling matrix architecture, significantly improving the Gray Code Oscillator (GCO)-TDC's resolution.
- Proposed a virtual bin calibration method (VBCM) for online resolution configuration and automatic calibration.
- Implemented the VBCM using the hardware description language (HDL). Through multiplexing critical components, this core is hardware-efficient.

3) Ultra-high resolution 4-edge Wave Union (WU) TDCs with a bidirectional encoder

- Manually routed for the WU launcher to precisely control the output pulse width.
- Proposed a bidirectional encoder for 4-edge real-time hardware encoding.
- Implemented the 4-edge WU TDC in a 16 nm FPGA device.

4) A Two-Stage Interpolation TDC Implemented in multiple FPGA devices

- Proposed a new two-stage interpolation TDC architecture and introduced the measurement concept of this architecture.
- Used LUT-based instead of carry-based ring oscillators (ROs) to construct Vernier Gray code oscillator TDC (VGCO-TDC), reducing carry element utilisation.
- Reduced the length of the delay line and relevant hardware utilisation due to the tapped-delay line (TDL)-TDC only covering the resolution of the VGCO-TDC.

1.4 Outline of the thesis

The summary of the remaining chapters is shown below:

Chapter 2: A literature review of TDCs

Chapter 2 first introduces the critical parameters for TDCs. Following this, several digital-TDC architectures (which can be applied to both ASICs and FPGAs) for fine-time measurement are reviewed. This chapter then presents the specific designs for ASIC-TDCs and FPGA-TDCs, respectively. The comparison finally illustrates the different features of ASIC-TDCs and FPGA-TDCs, as well as the potential improvements of FPGA-TDCs.

Chapter 3: Automatic calibration TDCs implemented in a Zynq SoC[1]

Chapter 3 introduces the proposed automatic calibration TDCs implemented in a Xilinx Zynq-7000 SoC. This design uses the programmable logic (PL) of the SoC to construct a 16-channel TDC. Besides, the processing system (PS) of the SoC is used to implement the BRAM-based calibration method, enhancing TDCs' linearity.

[1] This design was published in ***Multichannel Time-to-Digital Converters With Automatic Calibration in Xilinx Zynq-7000 FPGA Devices***.

Chapter 4: Low hardware utilisation and resolution-configurable GCO-TDCs implemented in multiple FPGAs[1]

Although the automatic calibration method for TDCs is proposed and implemented in Chapter 3, the TDC's resolution is fixed, and the automatic calibration method has a dependency on PS in the SoC, limiting its application in general FPGAs. Hence, the VBCM method is proposed and implemented in general FPGA devices for FPGA-TDCs' configurable resolutions and bin-width calibration. Meanwhile, the sampling matrix is also proposed to enhance the GCO-TDC's resolution with low hardware utilisation.

Chapter 5: Ultra-high resolution 4-edge WU TDCs with a bidirectional encoder[2]

Chapter 5 introduces a manually routed WU launcher to precisely control the width of the output pulse series. Simultaneously, a bidirectional encoder is proposed and used for 4-edge real-time encoding. These innovations, combined with the sub-TDL and dual-sampling methods, allow the proposed TDC to achieve a 0.46 ps ultra-high resolution in a 16-nm Xilinx Zynq UltraScale+ MPSoC.

Chapter 6: Two-stage interpolation TDCs implemented in multiple FPGA devices[3]

The design in Chapter 4 achieves low hardware utilisation with acceptable resolutions, and the design in Chapter 5 has an ultra-high resolution but with significant hardware utilisation. Therefore, a trade-off between resolution and hardware utilisation exists in these two designs. In Chapter 6, the two-stage interpolation is proposed, aiming to achieve high resolution and low hardware utilisation simultaneously.

Chapter 7: Conclusion

This chapter summarises the main contributions of my Ph.D. study. A future research plan is also included in this chapter.

[1] This design was published in ***Low-Hardware Consumption, Resolution-Configurable Gray Code Oscillator Time-to-Digital Converters Implemented in 16 nm, 20 nm, and 28 nm FPGAs.***

[2] This design was published in ***High-resolution time-to-digital converters (TDCs) with a bidirectional encoder***.

[3] This design was published in ***A Two-Stage Interpolation Time-to-Digital Converter Implemented in 20 and 28 nm FGPAs***.

Chapter 2 A literature review of TDCs

This chapter first introduces parameters used to characterise TDC's performance, facilitating a fair comparison across different designs. It then introduces mainstream TDC architectures and enhancing methods. Finally, FPGA-TDCs are summarised and compared with ASIC-TDCs in this chapter, revealing the potential improvements of FPGA-TDCs.

2.1 Performance parameters

For TDCs, critical parameters for characterising TDCs include resolution, linearity, precision, accuracy, and dead time. The subsequent contents will present detailed explanations and formulas for these parameters, offering an overall understanding of how they influence TI measurement.

2.1.1 Resolution

Resolution, also referred to as the least significant bit (LSB), bin width, and quantization step, is the primary parameter of TDCs [8]. It is the minimal TI that can be distinguished by a TDC [66]. Similar to analog-to-digital conversion, the process of time-to-digital conversion also involves mapping and quantifying a timestamp into a digital code, as shown in Fig. 2.1. Ideally, the width of every quantization step is identical (highlighted in red in Fig. 2.1). However, quantization steps are actually different (highlighted in blue in Fig. 2.1). Therefore, the resolution is usually evaluated by the average bin width. For a TDC with N time bins, the resolution is calculated as:

$$LSB = \frac{T}{N}, \tag{2.1}$$

where T is the measurement range that a fine counter covers.

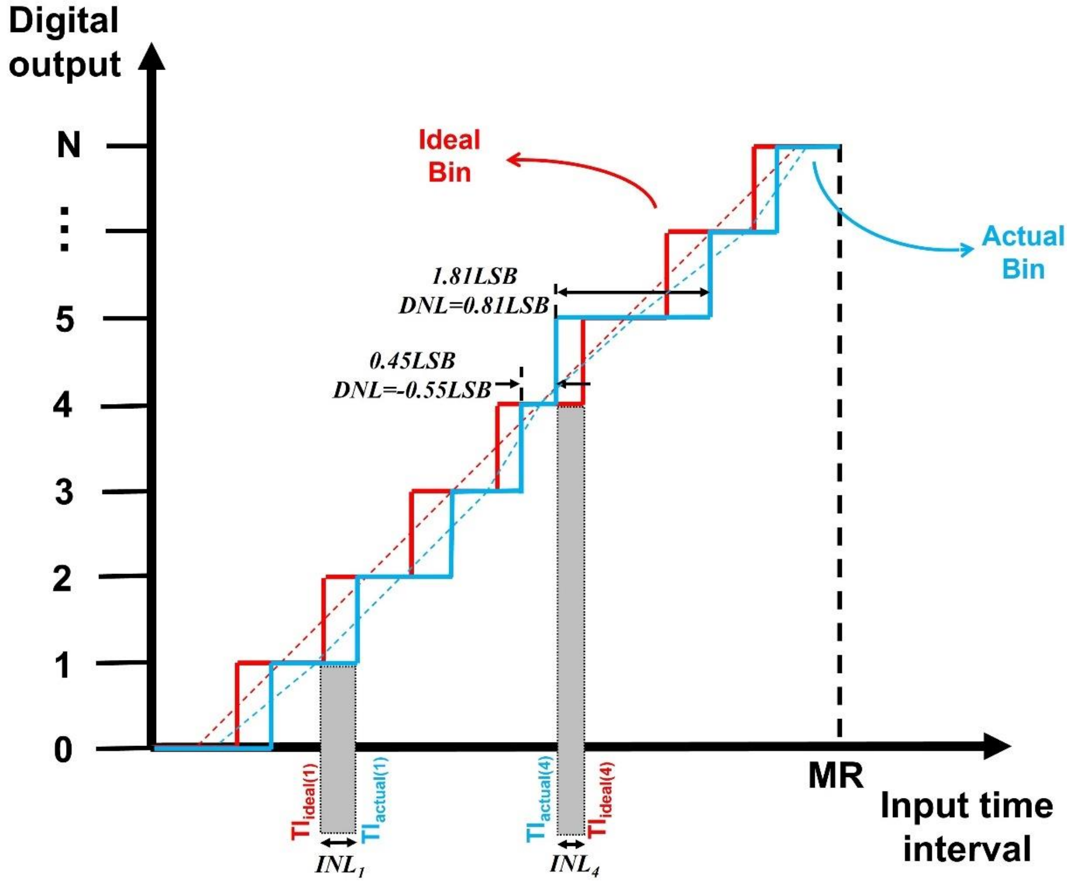


Figure 2.1 TDC conversion function (DNL: differential nonlinearity; INL: integral nonlinearity).

2.1.2 Linearity

The variance between the ideal bin width and the actual bin width is characterised by linearity, including differential nonlinearity (DNL) and integral nonlinearity (INL). DNL reflects the deviation of an actual quantization step from the ideal value, while INL represents the measurement error accumulated from DNL. They are respectively calculated as:

$$DNL[i] = \frac{(W[i]-LSB)}{LSB}, \tag{2.2}$$

and

$$INL[i] = \sum_{n=0}^{i} DNL[i], \tag{2.3}$$

where *W[i]* is the actual bin width of the *i-th* time bin and is usually evaluated by static code density tests (SCDTs) [67].

Hits unrelated to the TDC's sampling clock or random hits are input into a TDC to perform SCDTs [68]. Ideally, all bin widths are uniform. Hence, the possibility of a hit falling to every time bin is equal. However, bin widths are different. Therefore, the number of hits collected at each time bin differs and is proportional to its actual bin width. With SCDTs, the actual bin width is calculated as:

$$W[i] = \frac{Num_{col.}[i]}{Num_{tot.}} \times T, \tag{2.4}$$

where $Num_{col.}[i]$ is the number of hits collected at *i-th* time bin and $Num_{tot.}$ is the total number of hits for the SCDTs.

2.1.3 Precision

Precision, sometimes also referred to as standard deviation precision (σ) or root-mean-square precision (RMS precision), is the parameter that represents the deviation of each measurement (represented by yellow x in Fig. 2.2) from the mean measurement value (represented by green cross in Fig. 2.2). It can be evaluated by the TI test, in which the TDC measures a fixed TI repetitively. Precision is calculated as:

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n-1}, \tag{2.5}$$

where x_i is the *i*-th measurement, and μ is the average value for *n* measurements when the TI is constant.

Many factors, including environmental noise like temperature drifts and electronic noise like jitters, deteriorate precision [19]. A bin-by-bin calibration method can improve the TDC precision [68]. It is calculated as:

$$t_k = \frac{W[k]}{2} + \sum_{j=0}^{k-1} W[j], \tag{2.6}$$

where t_k is the calibrated timestamp corresponding to the centre of the k-th time bin.

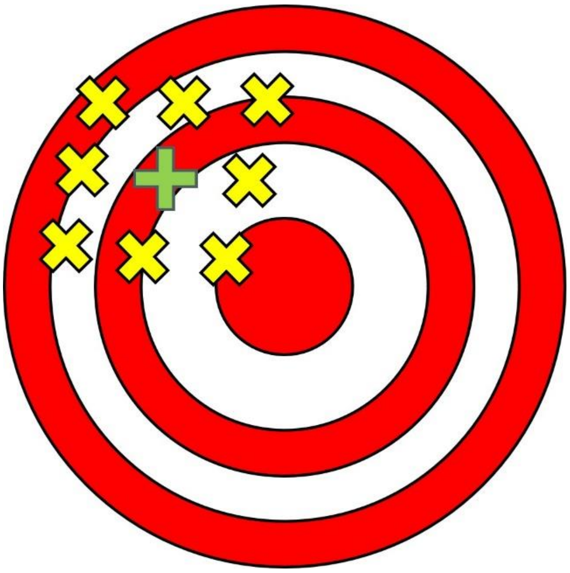


Figure 2.2 Concept of precision.

2.1.4 Accuracy

Accuracy in measurements represents to the correctness degree of a measurement value to the truth. For TI measurements, a TDC measures a fixed TI repetitively and uses the mean value as the final output. Hence, as shown in Fig. 2.3, the TDCs' accuracy is defined as the absolute difference between the mean value of multiple measurements and the true reference [19].

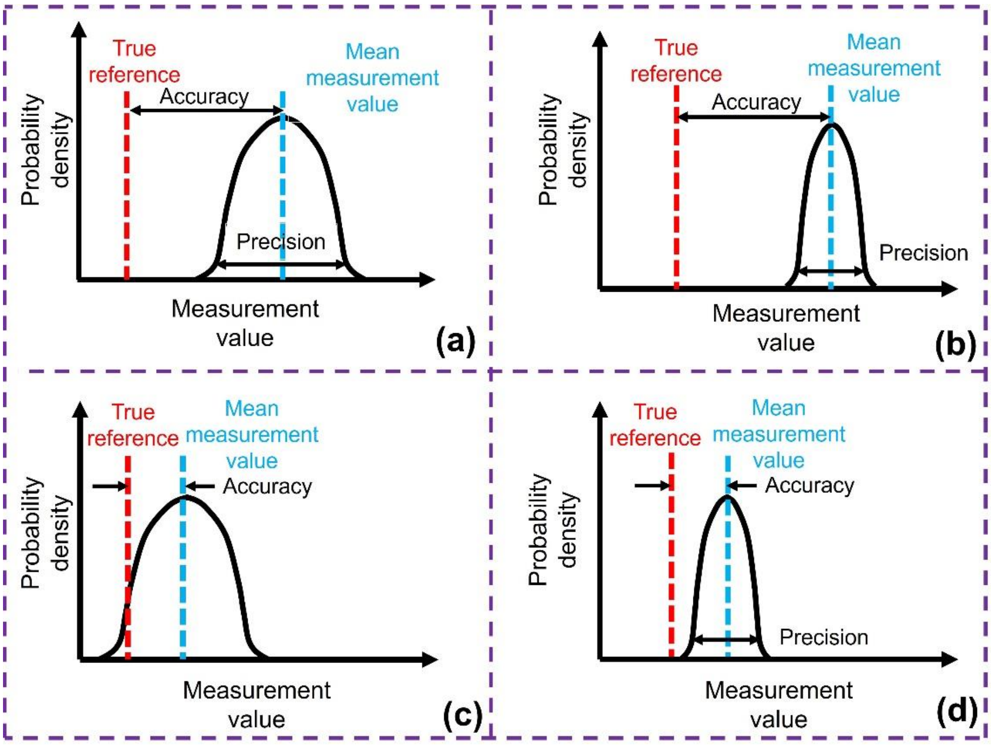


Figure 2.3 Measurement scenarios of a) low accuracy and low precision, b) low accuracy and high precision, c) high accuracy and low precision, and d) high accuracy and high precision.

Besides, Fig. 2.3 shows different measurement scenarios with different accuracy and precision. For measurements with high precision, as shown in Fig. 2.3b and Fig. 2.3d, the measurement results are more concentrated around the mean value than those shown in Fig. 2.3a and Fig. 2.3c. Hence, scenarios shown in Fig 2.3b and Fig 2.3d require fewer measurements to acquire a trustable result. Moreover, measurements in Fig. 2.3c and Fig. 2.3d have better accuracy, corresponding to the mean measurement value closer to the true reference. The precision can be improved by the bin-by-bin calibration method, as introduced in Section 2.1.3. However, the inaccuracy is usually treated as the offset during measurement and is calibrated by the offset cancelling.

2.1.5 Dead time

Dead time is the time required for a TDC to complete the current conversion and prepare for the subsequent measurement. It characterises TDC's capability to measure high repeat-rate events and is reciprocal to the sampling rate [31]. Low dead time (or high conversion rate) TDCs are important for time-resolved measurement systems.

Otherwise, pulses containing valid information from detectors like single photon avalanche diodes (SPADs) may be missed when the last conversion is conducting [4]. Besides, one TDC serves several SPAD pixels in most designs [69], [70], [71]. Hence, the low dead time in these scenarios can reduce the number of required TDC channels, as more SPAD pixels can share one TDC [19].

2.2 Digital fine-time measurement architectures

Compared with analog-TDCs, digital-TDCs are now mainstream in industrial and scientific applications due to their low power consumption, compact size, and high tolerance to interferences from temperature variations and electromagnetic noise. Besides, benefiting from advanced semiconductor manufacturing technologies, the resolution of FPGA-TDCs and ASIC-TDCs are both improved from hundreds of picoseconds [72], [73], [74] to several picoseconds [75], [76], [77], [78], [79], which is competitive compared with analog-TDCs. Therefore, this section will introduce the mainstream digital high-resolution TI-measurement architectures that can be implemented in both ASICs and FPGAs.

2.2.1 Tapped delay line architecture

TDL-TDC was first proposed in 1982 [80]. Now, the TDL is one of the most used architectures in reported designs because it can be easily implemented in both FPGAs and ASICs [81], [82], [83], [84]. Fig. 2.4 shows the basic architecture of a TDL-TDC. This architecture connects several delay cells into a line to propagate the "START" signal. Moreover, outputs from each delay cell in the delay line are sampled by corresponding DFFs driven by the "STOP" signal. Typically, the "START" signal contains a logic-level transition (for example, the "START" signal shown in Fig. 2.4 contains a rising edge), and the "STOP" signal is the TDC's system clock. The outputs of all delay cells are sampled simultaneously when the clock's sampling edge arrives. Then, a thermometer code (such as "11110000…00 " shown in Fig. 2.4) is output and converted into a binary code by the following encoder, representing the propagation distance of the logic-level transition along the delay line. According to the binary code, the TI between the input logic-level transition and the clock's sampling edge can be calculated as:

$$TI = b \times \tau, \tag{2.7}$$

where b is the output binary code and τ is the propagation delay of the delay cell. TDL-TDC has low dead time. However, the intrinsic propagation delay of delay cells restricts the resolution of a plain TDL-TDC. Moreover, TDL-TDC also consumes significant resources to construct the delay line and encode outputs from TDL.

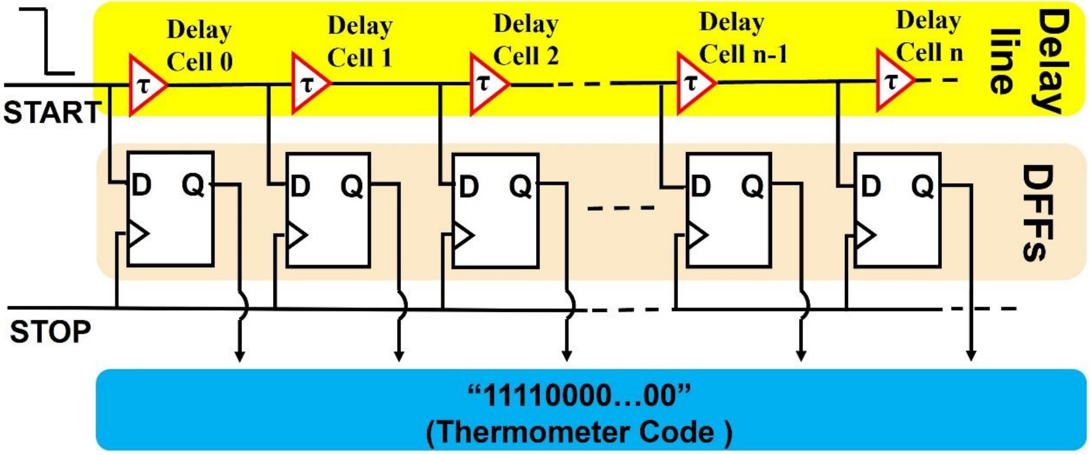


Figure 2.4 Architecture of TDL-TDC.

2.2.2 Ring oscillator architecture

For TDL-TDCs, a large number of delay cells is required to construct a delay line, leading to significant hardware utilisation of sampling DFFs and the encoder. To reduce hardware utilisation, the ring oscillator (RO) architecture [85], [86], [87], [88], [89] shown in Fig. 2.5 was proposed. Fig. 2.5a and Fig. 2.5b show RO-TDCs consisting of inverters and delay cells, respectively. The RO is free-running during measurement, and the signal always toggles along the inverters or the delay line. The corresponding outputs from the sampling logic module and the loop counter are recorded, respectively, when the “Start” and “Stop” signals assert. The TI between “Start” and “Stop” is then calculated as:

$$TI = (SL_{stop} - SL_{start}) \times \tau + (Counter_{stop} - Counter_{start}) \times 2n \times \tau, \qquad (2.8)$$

where SL_{stop} and SL_{start} are outputs from the sampling logic module, $Counter_{stop}$ and $Counter_{start}$ are outputs from the loop counter, and n and τ are respectively the number and propagation delay of inverters or delay cells.

The RO-TDC is preferred for large-scale TDC arrays and can be integrated into various circuits due to its low hardware utilisation. However, the free-running RO leads to high power consumption in both work and standby scenarios [90]. Therefore, based on the RO-TDC, the Gated RO(GRO)-TDC [91], [92], [93] was proposed in ASICs to stop the RO and reduce the power consumption in the standby scenario. However, to my knowledge, GRO-TDCs have not been implemented in FPGAs until now.

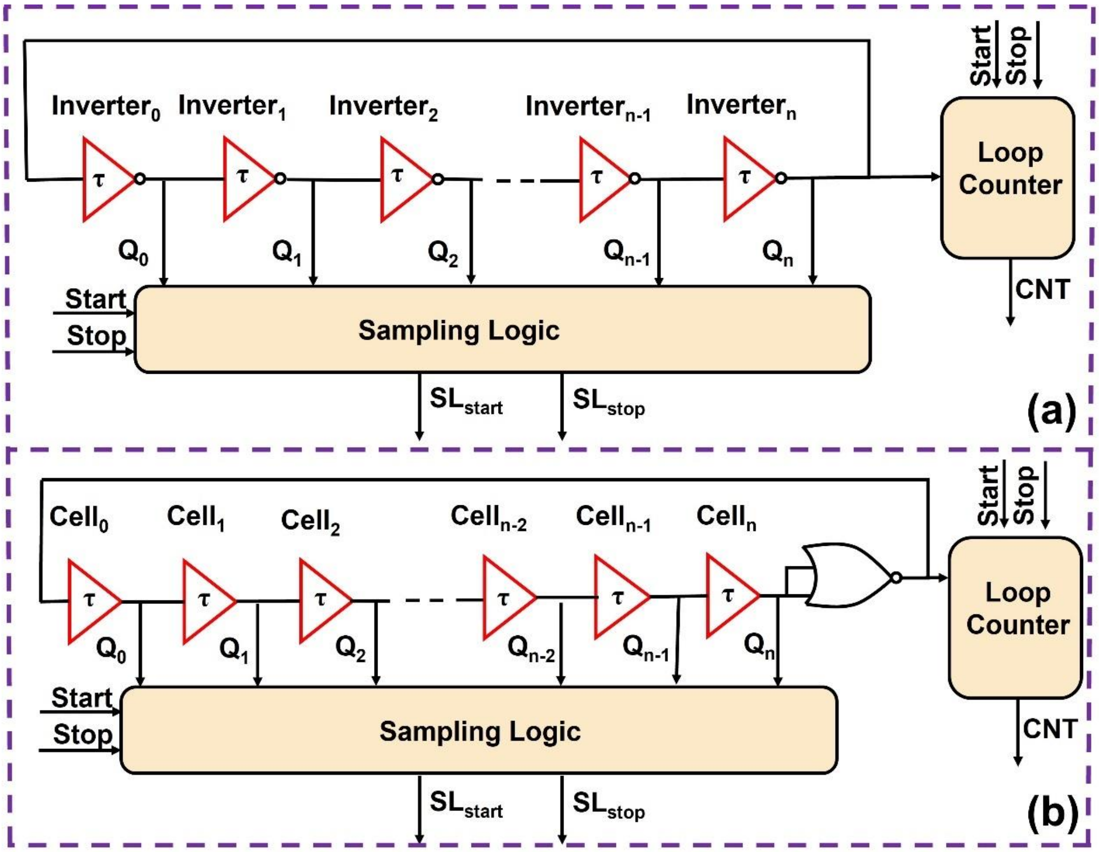


Figure 2.5 Architectures of ROs consisting of (a) inverters and (b) delay cells.

2.2.3 Pulse-shrink architecture

The intrinsic propagation delay of delay cells limits the resolutions of TDL-TDCs and RO-TDCs. Hence, the pulse-shrink architecture shown in Fig. 2.6a was proposed, aiming to break the process-related limitation and achieve a sub-gate-delay resolution [12]. In this method, the TI is represented as a pulse width and is input into the delay line. Then, the pulse is gradually shrunken along the delay line until it disappears due to the different rising (t_r) and falling (t_f) transition times. The resolution of this architecture is defined as $t_r - t_f$. And the conversion result is output as a thermometer code through flip-flops. The one-to-zero transition indicates the position where the pulse has disappeared, and the TI is calculated as:

$$TI = \left(t_r - t_f\right) \times n, \tag{2.9}$$

where *n* is the number of delay cells the pulse propagates along until disappearing.

To pursue a high resolution, the TDC in Fig. 2.6a requires a large number of delay cells, leading to significant area occupation. Hence, the loop pulse-shrink TDC (a pulse-shrink TDC with a loop delay line and a counter) shown in Fig. 2.6b was proposed. In Ref. [94] and [95], loop pulse-shrink TDCs can achieve 6 ps and 20 ps resolutions with 0.13 *um* and 0.8 *um* CMOS manufacturing process, respectively. Moreover, Li *et al.* designed a 2.38 ps resolution loop pulse-shrink TDC in a 16 nm Zynq UltraScale+ FPGA [96]. However, loop pulse-shrink TDCs still suffer from a long dead time due to the process of step-by-step pulse shrinking.

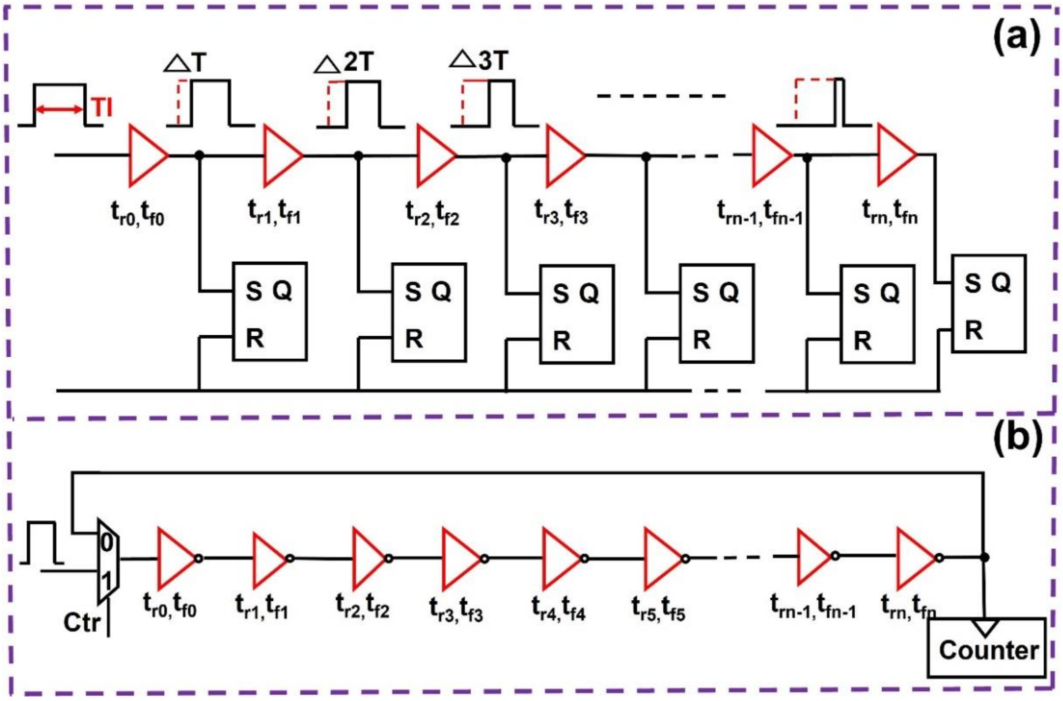


Figure 2.6 Architectures of (a) basic pulse-shrink and (b) loop pulse-shrink TDCs.

2.2.4 Vernier RO architecture

The cornerstone of the Vernier RO (VRO)-TDC shown in Fig. 2.7a is a pair of ROs, including a slow RO and a fast RO. The VRO-TDC also contains DFFs, counters, and a coincidence circuit (CC). The measurement concept of this method is shown in Fig. 2.7b. The slow and fast ROs subsequently launch after being independently triggered by the “START” and “STOP” signals. After several cycles, the rising edges of the slow and fast ROs’ outputs align. Simultaneously, the counters driven by the two ROs count and output respective results. In this architecture, the resolution and the measured TI are calculated as:

$$LSB = T_1 - T_2, \tag{2.10}$$

and

$$TI = n_1 \times T_1 - n_2 \times T_2, \tag{2.11}$$

where *T_1* and *T_2* are respectively the oscillation periods of the slow and fast ROs, and *n_1* and *n_2* are the outputs from the slow and fast counters when the fast RO catches up with the slow RO.

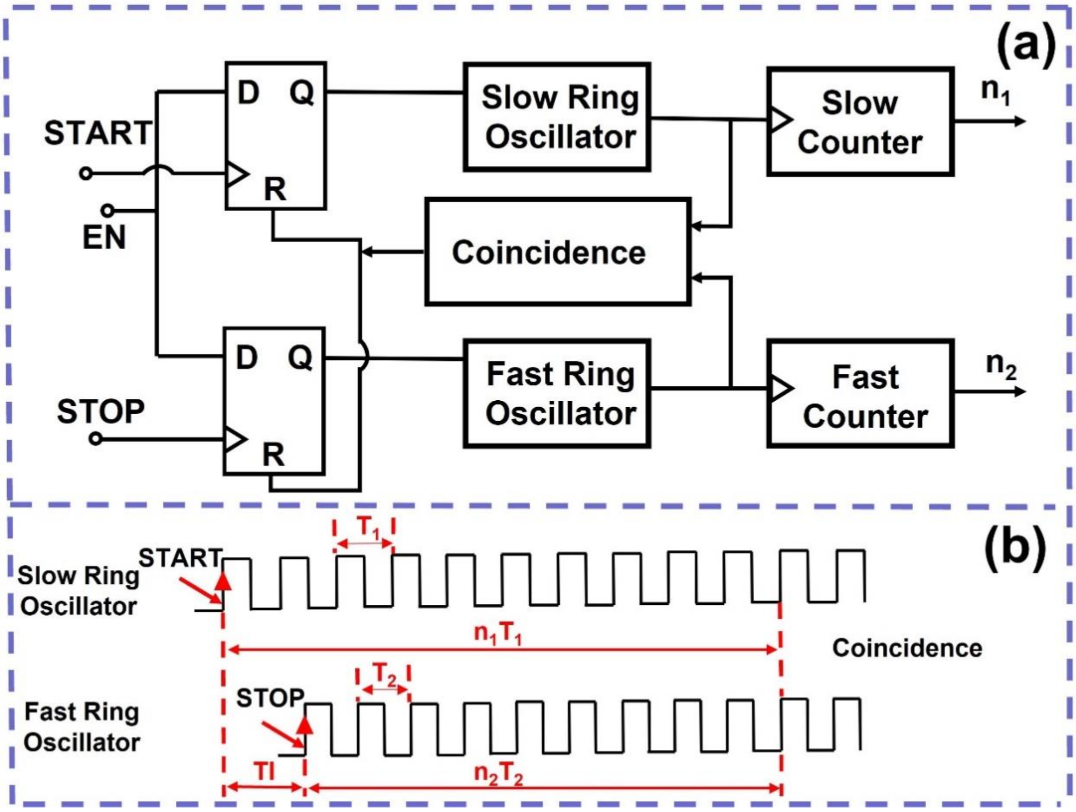


Figure 2.7 (a) Architecture and (b) concept of VRO-TDC.

The VRO-TDC can offer a satisfactory resolution with high linearity and low hardware utilisation[97], [98], [99], [100]. However, more oscillations are required for the same TI if a higher resolution is pursued, leading to a long dead time and significant accumulated jitters. Hence, the designer should make a trade-off between the resolution and other parameters when designing.

2.3 ASIC-TDC designs

AISCs have features of fully customised designs, including customised cells, placement and routing. Hence, ASIC-TDCs have more flexibility than FPGA-TDCs. This section introduces architectures that can only be implemented in ASICs.

2.3.1 Vernier TDL architecture

The Vernier TDL (VTDL) architecture, also referred to as the differential TDL architecture and the 2-D TDL architecture, modifies the TDL architecture and propagates both the "START" and "STOP" signals with different propagation delays [101], [102], [103]. Fig. 2.8 shows the VTDL-TDC's architecture. The resolution and measured TI can be respectively calculated as:

$$\tau = \tau_1 - \tau_2, \tag{2.12}$$

and

$$TI = n \times \tau, \tag{2.13}$$

where τ_1 and τ_2 are propagation delays of the delay cells for the "START" and "STOP" signals, respectively, and n is the number of DFFs sampling "1"s (high logic-level).

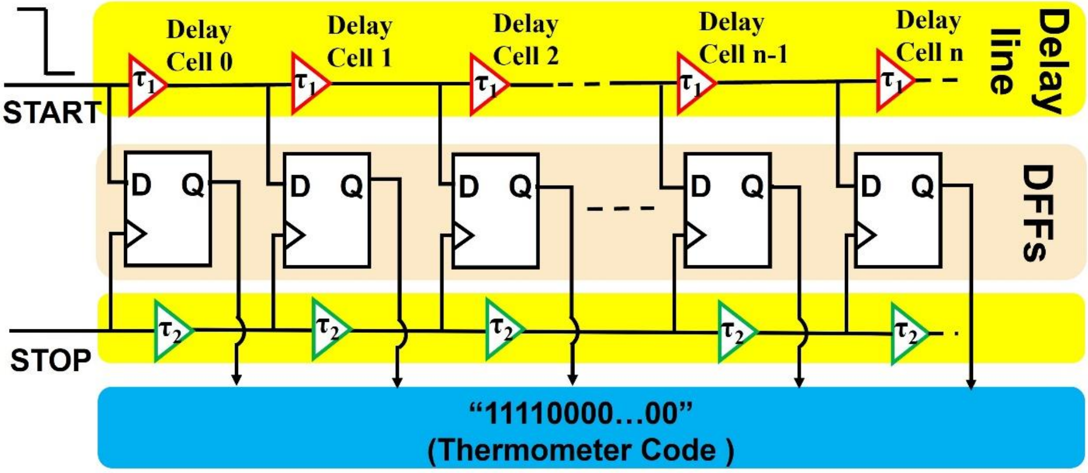


Figure 2.8 Architecture of VTDL-TDC.

It is worth noting that there are many available cells with different propagation delays in ASICs. However, the choice of delay cells in FPGAs is limited, making the architecture shown in Fig. 2.8 impossible to implement in FPGAs [66]. Besides, for VTDL-TDCs implemented in ASICs, the cell's propagation delay for the "STOP" signal should be less than that for the "START" signal. Otherwise, the "STOP" cannot overtake the "START", leading all DFFs to output "1". The VTDL architecture requires a longer delay line than the plain TDL architecture shown in Fig. 2.4. Hence, more delay cells and sampling DFFs are required for the same TI when pursuing a higher resolution.

2.3.2 Multi-path GRO architecture

Like the plain TDL-TDC, the delay cell's intrinsic propagation delay limits the GRO-TDC's resolution. To break this limitation, the multi-path GRO (MPGRO)-TDC shown in Fig. 2.9a was proposed in Ref. [104]. This architecture employs the multi-input inverters (shown in Fig. 2.9b) to tap into different stages of the RO, decreasing the effective delay per stage. As a result, the TDC's resolution is enhanced [4]. For example, in Ref. [91], the MPGRO-TDC manufactured by 0.13 μm process achieved a 6 ps resolution.

However, the multiple output transitions from different inverters may occur at the same time due to the inputs of each stage being connected to earlier stages than in a typical RO. (For example, the input of Z1 shown in Fig. 2.9b includes Z35, Z37, Z39, Z43 and Z47. But the input is only Z47 in a typical RO.) Hence, the output of the MPGRO-TDC should be segmentally encoded to ensure only one transition occurs at a time [91].

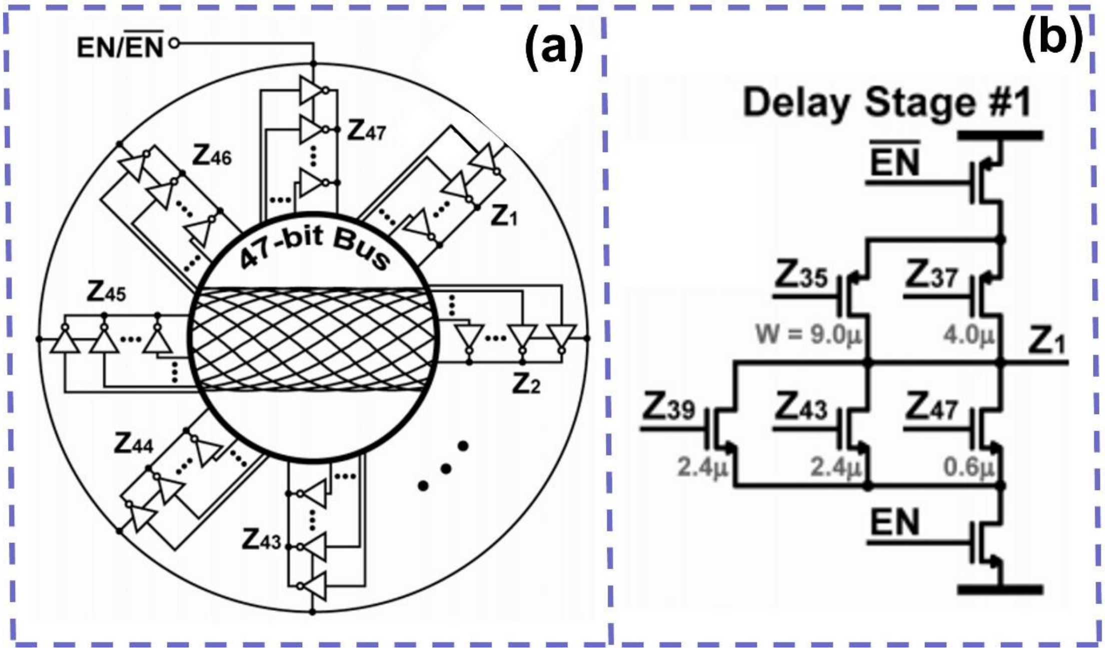


Figure 2.9 (a) MPGRO and (b) multi-input inverter [104].

2.4 FPGA-TDC designs

Compared with ASICs, FPGAs feature predefined configurable logic resources, including carry elements, LUTs, switch matrices, digital signal processing (DSP) hard macros, etc. Some of these logic elements can serve as delay cells when constructing TDCs. For example, FPGA TDL-TDCs usually use carry elements to propagate the input signal. Besides, due to the feature of the predefined logic, only limited methods can be used for FPGA-TDCs' enhancements and de-bubble (see section 2.4.4 below for details). This section will focus on architectures used only in FPGAs and cover methods of performance enhancements and de-bubble for FPGA-TDCs.

2.4.1 FPGA-specific TDC architectures

In addition to carry elements, other predefined logic resources in FPGAs can also be used as delay cells for FPGA-TDCs. Architectures for these FPGA-TDCs are introduced in this subsection.

a. DSP TDC

In FPGAs, conventional TDL-TDCs typically employ carry elements as delay cells due to their abundance and ease of implementation. However, since the carry element is configurable, its fanout and ON-resistance are much higher than those of the 48-bit dedicated calculation logic "DSP48", leading to a higher propagation delay [6]. Hence, in Ref. [105], [106] and [107], DSP48s in FPGAs are used as delay cells to form a delay line for TDL-TDCs (DSP-based TDL-TDCs, also known as DSP-TDCs). The architecture of DSP48 is shown in Fig. 2.10. It consists of a pre-adder, an arithmetic logic unit (ALU) and a multiplier. Both the ALU (shown in Fig. 2.10a) and the pre-adder (shown in Fig. 2.10b) can be utilised to form a delay line. Moreover, in the Kintex-7 FPGA, the ALU-based TDL-TDC has a resolution of 4.23 ps, which is better than that of the pre-adder-based TDL-TDC (LSB = 8.12 ps) and the carry-based TDL-TDC (LSB =10.7 ps) [106]. However, the linearity of ALU-based and pre-adder-based TDL-TDCs deteriorates compared with carry-based TDL-TDCs [106].

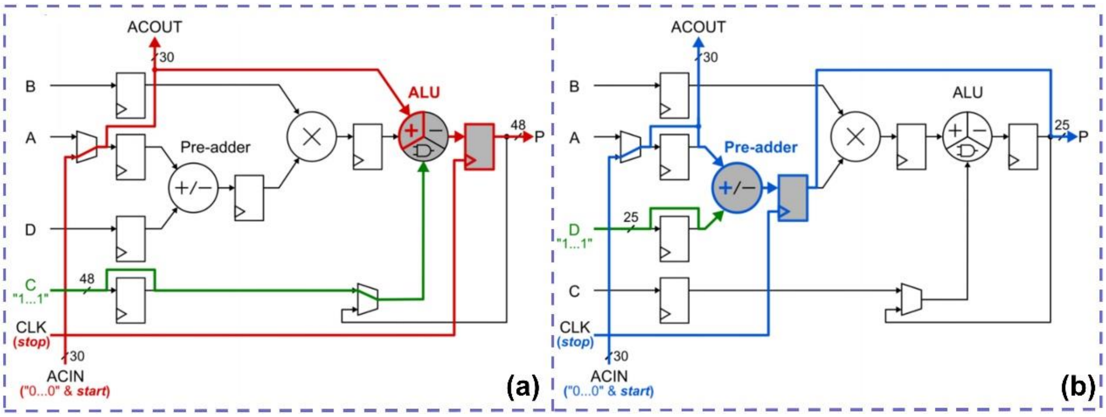


Figure 2.10 A simplified DSP48 configuration to form a delay line with the use of (a) ALU and (b) a pre-adder [106].

b. Large-scale parallel TDC

Routing delays are significant challenges for achieving timing closure. However, routing delays are subtly utilised as "delay cells" in the large-scale parallel routing (LSPR)-TDC [108], as shown in Fig. 2.11. In this design, the measured TI is first converted into a pulse width. Then, the pulse simultaneously propagates along 1024 paths with different routing delays. All the propagated pulses serve as the "enable" signal for counters driven by the same clock, and the interpolation is achieved by averaging the output values from all counters. Moreover, a 32-bit coarse counter is also used to extend the measurement range. In Ref. [108], this design was implemented in 40-nm Virtex-6, 28-nm Kintex-7 and 20-nm Kintex Ultrascale FPGAs, respectively, achieving resolutions of 3.95 ps, 1.29 ps and 5.5 ps. Although this architecture offers high resolutions, it is unsuitable for multi-channel designs due to the significant consumption of routing resources per channel [108], potentially leading to routing congestion issues in multi-channel designs.

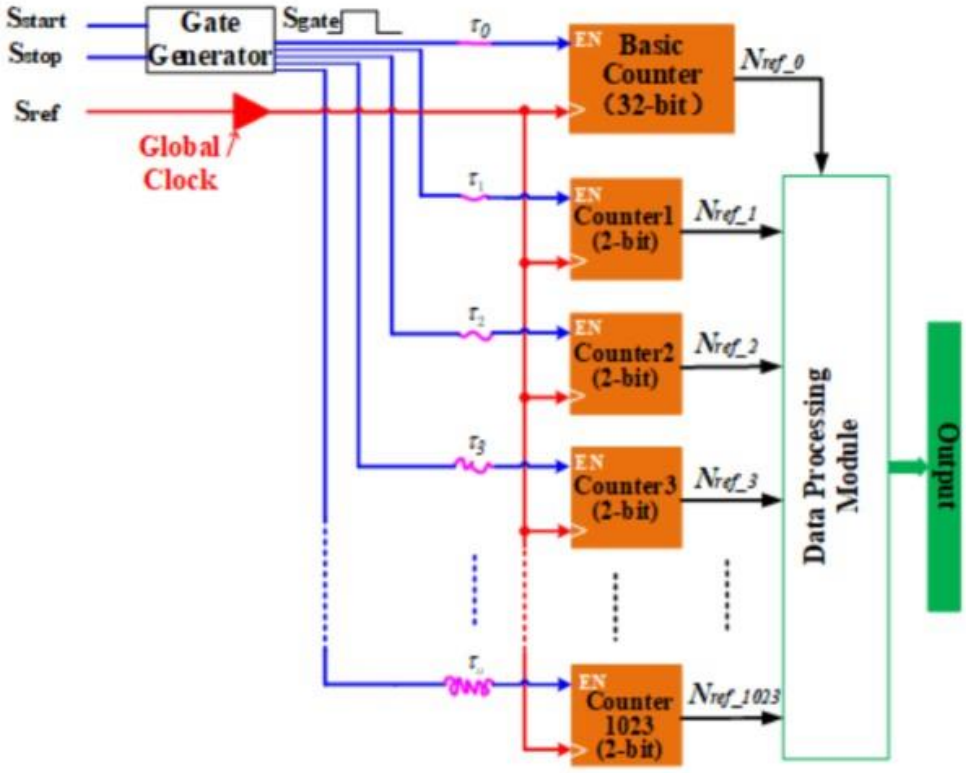


Figure 2.11 Diagram of the LSPR-TDC [108].

c. GCO-TDC

Combinational loops such as in Fig. 2.5 are generally avoided in standard digital circuit designs, especially for the multi-bit case. The reason is that propagation delays along various feedback paths differ, potentially resulting in the “race and competition” phenomenon [109]. However, in contrast to the conventional binary code, only one bit toggles between two successive states in the Gray code. Hence, the GCO shown in Fig. 2.12a is an exception among multi-bit counters that can be implemented in combinational logic due to its immunity to the “race and competition” phenomenon.

Based on the GCO, the GCO-TDC shown in Fig. 2.12b was first proposed by Wu and Xu [109]. The GCO-TDC consists of two components, including a LUT-based GCO and sampling DFFs. The GCO operates independently of a clock signal and runs freely once the single “OKOP” is asserted. Then, the outputs of the GCO are sampled by DFFs when the rising edge of the clock arrives. Finally, the sampled output is converted into a binary code, corresponding to the TI between the input signal “OKOP” asserting and the subsequent rising edge of the clock. Moreover, a “FIN” signal in Fig. 2.12b is also designed to prevent the oscillator from running indefinitely which causes unnecessary power consumption. Low logic resource consumption is the primary advantage of GCO-TDCs. However, this architecture suffers from a limited resolution. Hence, the double-sampling method was proposed for a better resolution [110]. Besides, a careful manual routing strategy was proposed to enhance the GCO-TDC’s linearity [111].

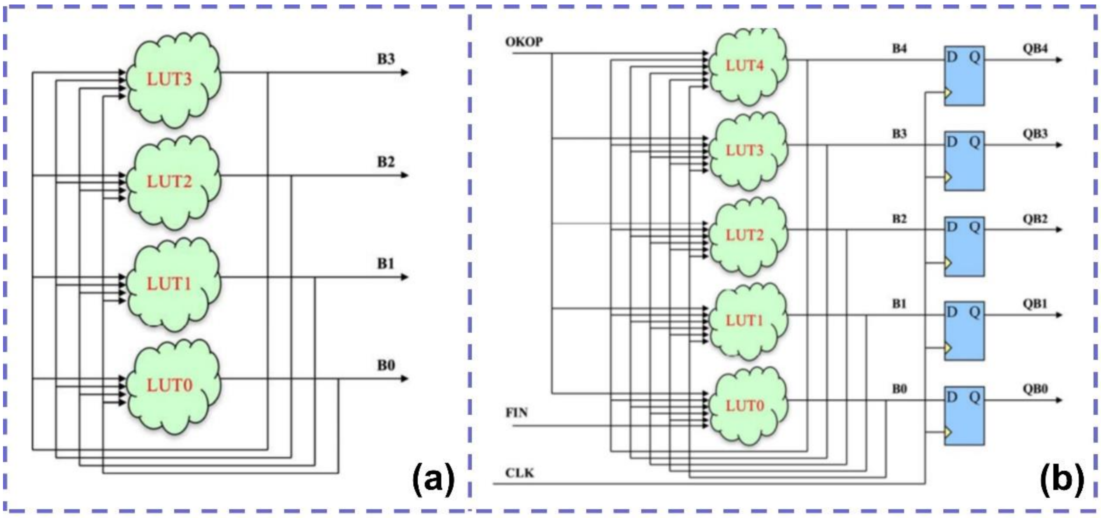


Figure 2.12 Diagrams of (a) GCO and (b) GCO-TDC [109].

2.4.2 Resolution enhancement methods

The resolutions of TDL-TDCs are limited by the inherent propagation delays, which are determined by the FPGAs' manufacturing process. This subsection introduces methods that improve the resolutions of TDL-TDCs beyond the limitations of manufacturing technologies.

a. Multi-chain merging method

Multi-chain merging is the most straightforward method to break the manufacturing-process limitation of resolutions [112], [113], [114]. Fig. 2.13 shows the diagram of the multi-chain merging method. For a single TDL, N delay cells are used to cover a sampling clock period T, and the resolution of a single-TDL-based TDC is calculated according to Eq. (2.1). However, for n-TDLs parallelly sampling, $n \times N$ cells are used to cover the sampling period, and the equivalent resolution is then calculated as:

$$LSB = \frac{T}{n \times N}. \tag{2.14}$$

In Ref. [112], the average bin width of a plain TDL-TDC implemented in a Virtex-6 FPGA is 24 ps. The average bin-width is then improved to 1.5 ps with 16-chain parallel sampling. Moreover, the design presented in Ref. [113] has a resolution of 1.15 ps with 20-chain parallel sampling in a Virtex-6 FPGA, achieving around a 21-fold enhancement from a resolution of 24.04 ps.

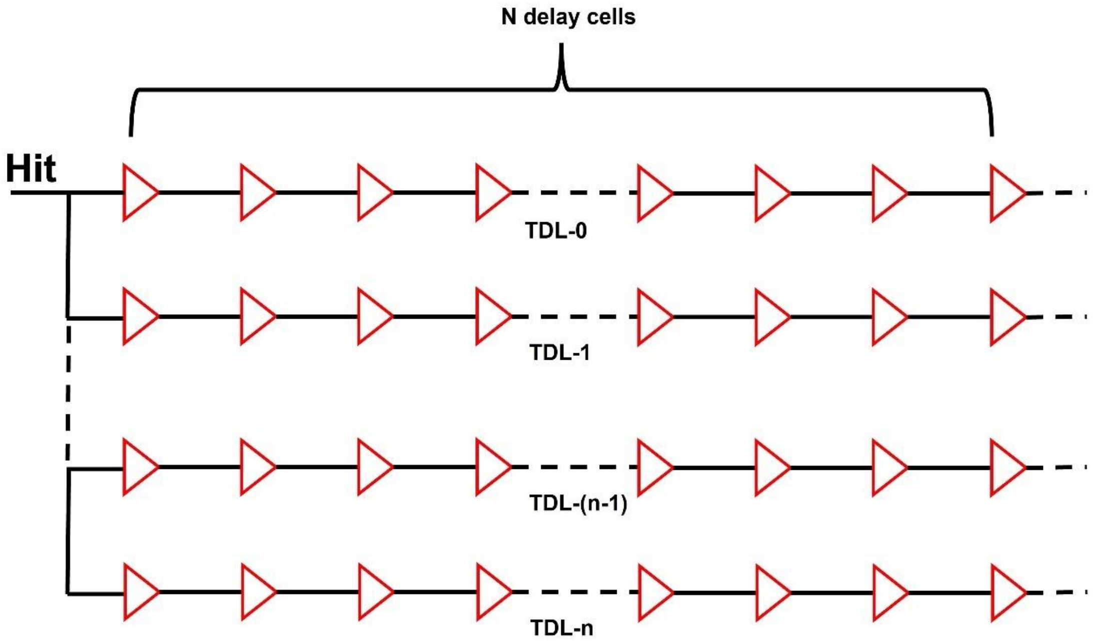


Figure 2.13 Block diagram of multi-chain merging.

b. Wave union method

The multi-chain merging method is efficient in enhancing the resolution. However, the hardware utilisation increases significantly due to the implementation of parallel TDLs. Therefore, the WU method was proposed to enhance resolutions with higher hardware utilisation efficiency [115]. The WU method, like the multi-chain merging method, employs multiple measurements for the same TI to enhance the resolution. However, in the WU method, multiple measurements are achieved by inputting a series of pulses (which include both rising and falling edges) into the TDL instead of using parallel TDLs. Therefore, the WU method has lower hardware utilisation. The concepts of WU methods, including WU-A and WU-B, are shown in Fig. 2.14a and Fig. 2.14b, respectively. As the figures show, the primary difference between WU-A and WU-B is the length of the pulse series. WU-A utilises a no-feed-back wave launcher to generate a pulse series with finite length. However, WU-B generates an infinite-length pulse series by using a feed-back wave launcher [115]. WU-B improves the resolution more than WU-A due to containing more logic transitions. However, it has a longer dead time [115].

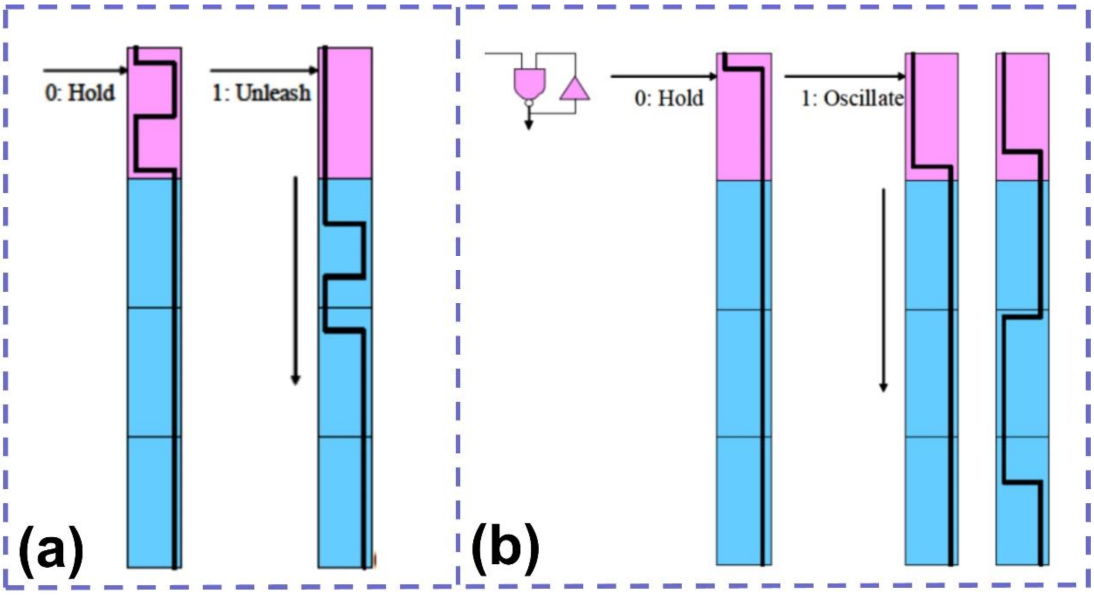


Figure 2.14 Concepts of (a)WU-A and (b)WU-B [115].

The WU method has been applied in several designs since it was proposed [79], [82], [116], [117]. For example, a design involving both the WU-A and WU-B was implemented in a Kintex-7 FPPGA, achieving a 0.4 ps resolution and an RMS precision of less than 5.2 ps [79]. Besides, a TDC combining WU-A and double sampling methods was presented in Ref. [82], achieving a 1.23 ps resolution in a Kintex-UltraScale FPGA.

The advantages and disadvantages of the aforementioned resolution enhancement methods are summarised in Table 2-1. The multi-chain merging method is easy to implement in FPGAs. However, it has significant hardware utilisation. In contrast, the WU method is hardware-efficient. However, it suffers from complex encoding.

Table 2-1 Comparisons between the resolution enhancement methods of TDL-TDC

Method	Pros	Cons
Multi-chain merging	● Easy implementation	● High hardware utilisation
WU	● Low hardware utilisation	● Complex encoding process

2.4.3 Linearity enhancement methods

Nonlinearity always exists in FPGA-TDCs due to imperfect manufacturing and clock skews. Four methods to enhance linearity are introduced in this subsection.

a. Tuned-TDL method

CARRY4s (in Xilinx 6-series and 7-series FPGAs) or CARRY8s (in Xilinx UltraScale and more advanced FGPAs) are cascaded as delay cells to form TDLs in TDL-TDCs [15], [16]. Fig. 2.15a and 2.15b show the block diagrams of CARRY4s and CARRY8s. As the figures show, each CARRY element has one direct output "CO" and one XOR-gated output "O". For CARRY4s, either "CO" or "O" can be selected as the output of a CARRY element. However, for CARRY8s, "CO" and "O" can be output simultaneously. Conventional TDL-TDCs select the same output type (such as all select "CO") as the output pattern. However, in Ref. [118], Won and Lee proposed that modifying output patterns can enhance the linearity of TDL-TDCs. They found that the pattern "CO-O-CO-O" delivers the best linearity in Kintex-7, Virtex-6 and Spartan-6 FGPAs. Since this method requires no extra hardware resources, it is suitable for multi-channel designs. For example, Chen and Li implemented 96-channel TDCs with the tuned-TDL method in Xilinx Kintex-UltraScale and Virtex-7 FPGAs [83].

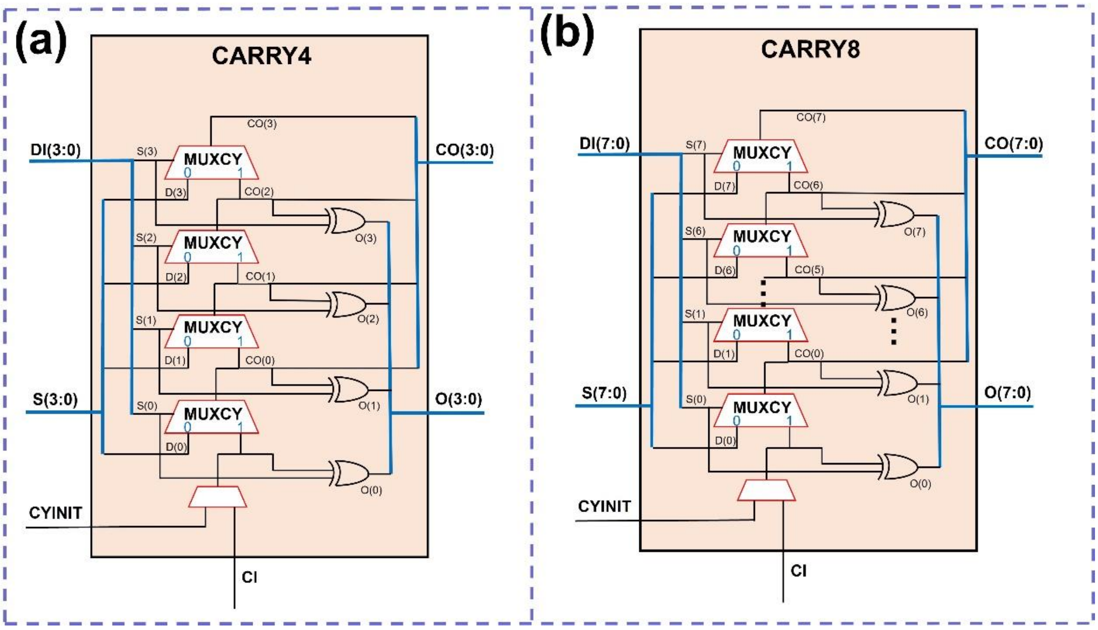


Figure 2.15 Block diagrams of (a) CARRY4 and (b) CARRY8.

b. Bin-width calibration method

In addition to the tuned-TDL method, the bin-width calibration method can also improve linearity. Dutton *et al.* first used the bin-width calibration method in a ToF ranging system for data post-process [119]. Then, Chen and Li modified this method, achieving online calibration in FPGAs [83], [120]. As shown in Fig. 2.16, the bin widths of actual bins vary significantly, with some bins spanning multiple ideal bins highlighted in yellow in Fig. 2.16. Hence, when a hit falls into an actual time bin, this hit is first remapped to the corresponding ideal bins in the bin-width calibration method. For example, a hit falling into actual bin N is remapped to ideal bin N-1 and ideal bin N, as shown in Fig. 2.16. Then, weighted accumulation is conducted according to the actual bin's bin width located at the corresponding ideal bins. TDCs using this method achieve a 0.13 LSB $\mathrm{DNL}_{\text{pk-pk}}$ (with a 10.54 ps LSB) in a Virtex-7 FPGA and a 0.23 LSB $\mathrm{DNL}_{\text{pk-pk}}$ (with a 5.02 ps LSB) in a Kintex UltraScale FPGA [83]. However, this method requires manual pre-calculation channel-by-channel [83].

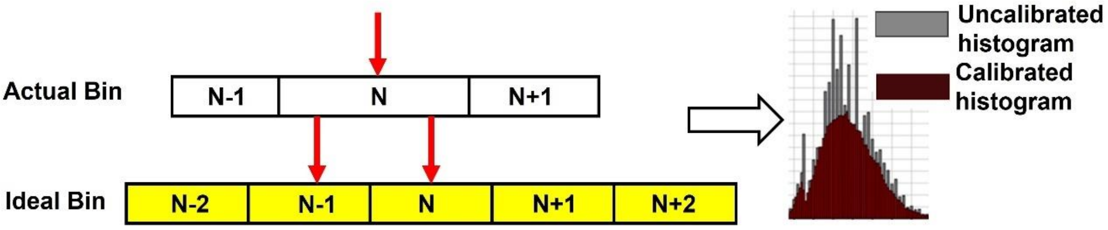


Figure 2.16 Concept of the bin-width calibration method.

c. Bin decimation method

The bin decimation method, also referred to as the binning method or the down-sampling method, was first proposed by Wang and Liu [121]. Fig. 2.17 shows the concept of this method. By merging adjacent time bins, this method builds larger bins and enhances the uniformity of bin widths without extra hardware utilisation. However, the improvement in linearity is achieved at the expense of the reduced resolution, which limits its application in high-resolution TDCs. For example, in Ref. [30], Xie *et al.* merged up to 20 actual bins into one bin in a Kintex-UltraScale FPGA, achieving a 0.275 LSB $\mathrm{DNL}_{\text{pk-pk}}$ with the resolution deteriorating from 5 ps to 100 ps. Similarly, by merging adjacent time bins and bin-width calibration in a Kintex-7 FPGA, Liu *et al.* achieved a 0.046 LSB $\mathrm{DNL}_{\text{pk-pk}}$ with LSB = 41.67 ps [122].

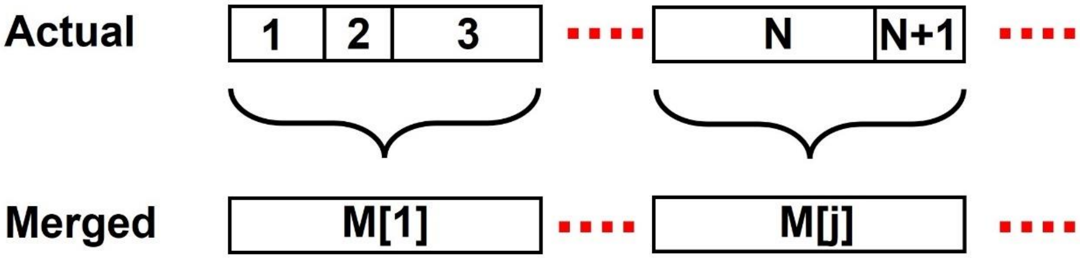


Figure 2.17 Concept of the bin decimation method.

d. Multi-phase sampling method

For TDL-TDCs, the TDL's length must cover the entire sampling clock period; otherwise, the TDC may miss valid inputs [123]. Advances in CMOS manufacturing technology have significantly improved the propagation delay of carry elements, leading to a much longer TDL required to cover the sampling period. However, this increase in length can lead to challenges, such as ultra-wide bins, especially when the

TDL crosses boundaries between clock regions. The clock skew between adjacent clock regions causes this phenomenon. To address this issue, the multi-phase sampling method was used to reduce the TDL's length and avoid clock regions' crossing [123]. For example, Won *et al.* reported a dual-phase sampling TDC in Ref. [123], and Chen and Li reported a tri-phase sampling design in Ref. [120]. However, the multi-phase design may induce extra errors, increasing the design complexity. Besides, due to the limited clock routing resources, the designer should carefully select the number of phases according to the implemented device.

Table 2-2 summarises the pros and cons of the aforementioned linearity enhancement methods. The tuned-TDL and bin decimation methods have the advantage of requiring no extra hardware utilisation. However, the bin decimation method suffers from deteriorated resolutions, and the tuned-TDL method can only enhance the linearity slightly. The bin-width calibration method can improve the linearity significantly. However, this method requires manual pre-calculation channel-by-channel, which is time-consuming. Without pre-calculation, the multi-phase sampling method aims to tackle the ultra-wide bins caused by the clock-region crossing. However, this method increases design complexity and consumes extra clock routing resources.

Table 2-2 Comparisons of linearity enhancement methods

Method	Pros	Cons
Tuned-TDL	● Without extra hardware utilisation ● Without pre-calculation	● Slight linearity enhancement
Bin-width calibration	● Significant linearity enhancement	● Requiring manual pre-calculation
Bin decimation	● Significant linearity enhancement ● Without extra hardware utilisation	● Reduced resolution ● Require raw bin-width information
Multi-phase sampling	● Without pre-calculation	● Increasing design complexity ● Consuming extra clock routing resources

2.4.4 De-bubble methods

TDLs may produce bubbles [unexpected “0” (low-logic level) among “1”s (high-logic level) or “1” among “0”s] in their output due to carry elements’ uneven propagation delays and clock skews. And bubbles can cause encoding errors when converting thermometer codes into binary codes. Therefore, bubbles should be removed before encoding, or a bubble-immune encoder should be designed. This subsection introduces methods to tackle bubbles.

a. Sub-TDL method

Bubbles are a common problem in FPGA-based TDL-TDCs. Since carry elements have faster propagation delays in high-end FPGAs, TDL-TDCs implemented in high-end FPGAs are more vulnerable to clock skews and have more severe bubbles. Therefore, the term “maximum bubble depth (MBD)” was proposed to characterise the maximal interval between the trustable logic transition and the unexpected logic transition (for instance, in pseudo thermometer code “11100111000”, the MBD is five) [124]. To address bubble problems, the sub-TDL method [120] (also referred to as the decomposition method [124]) shown in Fig. 2.18 was proposed. The bin widths of the sub-TDL and the plain-TDL are highlighted in blue and yellow in Fig. 2.18, respectively. As shown in Fig. 2.18, the sub-TDL extends TIs between taps by down-sampling TDL’s taps to obtain a bubble-free output. Outputs of every sub-TDL are then summed together to maintain the TDL-TDC’s resolution. It is worth noting that the interval between taps in the sub-TDL should be longer than the MBD; otherwise, bubbles still exist in the sub-TDL’s output. For example, in Fig. 2.18, the MBD is three, and the interval between neighbouring taps in sub-TDL is four. The sub-TDL method suppresses bubbles with acceptable extra logic (an adder for summing). Hence, this method has high hardware utilisation efficiency and is appropriate for multi-channel designs. For example, Song *et al.* designed a 256-channel TDC with the sub-TDL method in a Xilinx Kintex-7 FPGA [124].

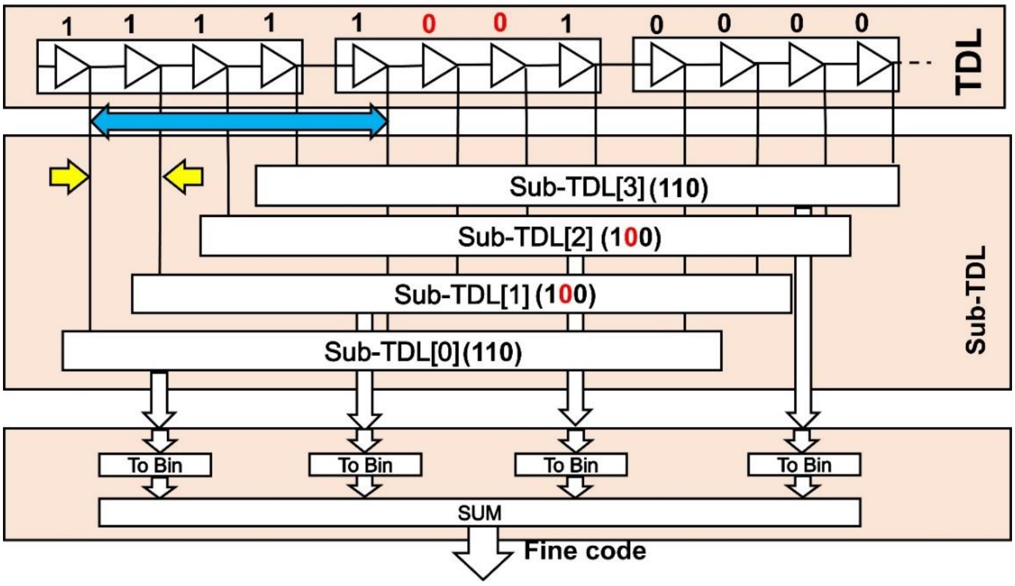


Figure 2.18 Block diagram of the sub-TDL method.

b. Bin realignment method

The bin realignment method addresses the bubble issue by switching the positions of bubbles and the next tap's output until all bubbles are moved (for example, this method converts "1110011000…" into "1111100000…" by switching bin positions) [125]. The concept of this method is straightforward. However, it requires iterations to cover all bubble scenarios for a single channel, and these realignments need to be applied to each channel respectively. Hence, the bin realignment method is time-consuming.

c. Ones-counter method

Unlike the sub-TDL and bin realignment methods, which aim to remove bubbles before converting the thermometer code into a binary code, the ones-counter method counts the number of bins with the valid state (such as logic "1" or logic "0") and outputs binary codes directly [126]. The block diagram of the ones-counter encoder is shown in Fig. 2.19. It uses three 6-input LUTs as a component to count the number of bins with the valid state for a 6-bit segment from the TDL. Outputs from all LUT-based components are then summed together as the final result. The ones-counter encoder is immune to bubbles and offers competitive hardware utilisation efficiency and latency compared with other encoders for TDL-TDCs. However, it cannot be applied to WU-

TDCs because the propagation distance of the input pulse series in WU-TDCs is not proportional to the number of bins with the valid state.

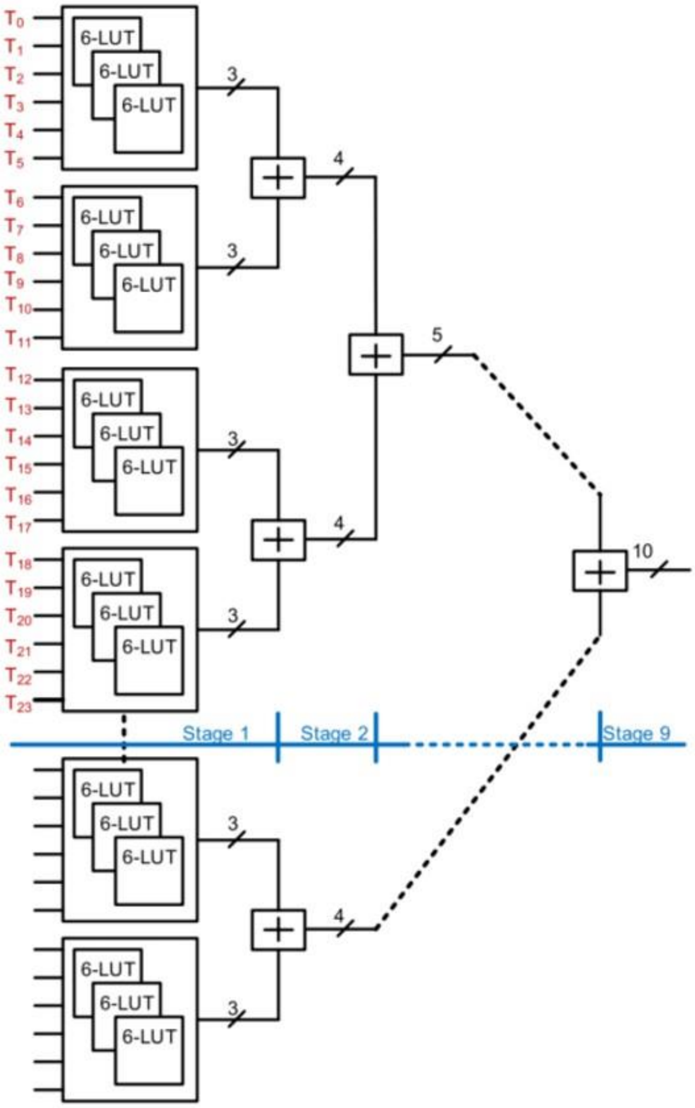


Figure 2.19 Block diagram of the ones-counter method [126].

Table 2-3 summarises the features of different de-bubble methods. Both sub-TDL and bin realignment methods are hardware-efficient. However, the sub-TDL method requires prior information on the MBD, and the bin realignment method necessitates iterations to cover all bubble scenarios. In contrast, the ones-counter does not require prior information or iterations. However, it cannot be applied to WU-TDCs.

Table 2-3 Comparisons of de-bubble methods

Method	Pors	Cons
Sub-TDL	● Low hardware utilisation	● Requiring the MBD information
Bin realignment	● Without extra hardware utilisation	● Requiring iterations
Ones-counter	● No prior information or iterations required	● Inappropriate for WU-TDCs

2.5 Comparison

Parameters and architectures for ASIC-TDCs and FPGA-TDCs are introduced in this chapter. The TDL, RO, pulse-shrink and VRO architectures can be implemented in both ASICs and FPGAs. However, only ASICs can implement VTDL and MPGRO architectures, benefiting from the customised cells. Besides, ASIC-TDCs deliver better performance than FPGA-TDCs due to careful placement and routing. However, FPGA-TDCs are mainstream in rapid prototype verification because of short development cycles and low development costs.

The recently reported FPGA-TDCs and features of FPGA-TDC architectures are summarised in Tables 2-4 and 2-5, respectively. Table 2-4 shows that the TDL-TDC is the mainstream of FPGA-TDCs due to its easy implementation. Moreover, other TDCs, such as the RO-TDC, pulse-shirk TDC, VRO-TDC, GCO-TDC, DSP-based TDL-TDC and LSPR-TDC, are well-developed.

Table 2-4 Recently reported FPGA-TDCs

Ref- Year	Methods	Device- Process	LSB (ps)	RMS Pre. (ps)	DNL (LSB)	INL (LSB)	DFF	LUT	BRAM[1]	DSP	CLB/Slice
TDL-TDC											
[112]-15	Multi-chain TDL	Virtex-6 (40 nm)	1.7	4.2	-	-	-	-	-	-	-
[127]-16	WU-A, Multi-chain TDL	Spartan-6 (45 nm)	0.9	<6	[-1, 6.25][2]	[-26.2, 11.5]	-	-	-	-	-
[113]-17	Multi-chain TDL, Offset-cancel	Virtex-7 (28 nm)	1.15	3.5	[-0.98, 3.5]	[-5.9,3.1]	N/S	19666[3]	43	127	N/S
[83]-19	Sub-TDL, Tuned-TDL Bin-width calibration	Virtex-7 (28 nm)	10.54	14.59[4]	[-0.05, 0.08]	[-0.09, 0.11]	1916	1145	1.5	-	712[5]
		Kintex- UltrScale (20 nm)	5.02	7.8[4]	[-0.12, 0.11]	[-0.18, 0.46]	1195	703	1.5	-	-
[82]-22	Sub-TDL, Double sampling, WU-A	Kintex- UltrScale (20 nm)	1.23	3.67[4]	[-0.84, 7.93]	[-6.36,24.70]	3463	2460	7.5	-	405[6]
[128]-22	Mixed-binning	Kintex- UltrScale (20 nm)	51.28	15.89[4,7]	0.036[8]	0.055[8]	1124	663	2.5	-	185[6]
			83.33	21.67[4,7]	0.030[8]	0.029[8]					
			105.26	36.32[4,7]	0.018[8]	0.017[8]					
[79]-23	Multisampling, WU-B	Kintex-7 (28 nm)	0.4	<5.2	[-0.97,5.95]	[-8.02,219.30]	2998	6304	43	-	2184[5]

Ref-Year	Methods	Device-Process	LSB (ps)	RMS Pre. (ps)	DNL (LSB)	INL (LSB)	DFF	LUT	BRAM[1]	DSP	CLB/Slice
Other TDCs											
[129]-17	LSPR	Virtex-5 (65 nm)	6.8	6.8	0.74[9]	1.57[9]	1410	666	2	-	1265[5]
[89]-21	RO, Delay tuning	Zynq-7 (28 nm)	-	92.7[4]	-	-	53	29	-	-	-
[110]-21	GCO, Double sampling, Bin-by-bin Cali,	Zynq-UltraScale+ (16 nm)	69	54.99	[-0.95,0.81]	[-1.01:0.49]	5	19	-	-	-
[96]-22	Loop pulse shrink	Zynq-UltraScale+ (16 nm)	2.38	<5.66	[-0.23,0.22]	[-0.24,0.27]	-	-	-	-	<10[5]
[130]-22	DSP-based TDL, 5 Edge WU-A	Artix-7 (28 nm)	-	11.49[4]	-	-	-	-	-	10%[10]	-
[131]-23	VRO, Delay-adjustment, Multi-step	Stratix-III (65 nm)	<10	13.2	[-1,1]	[-1,1]	318	1064	-	-	-

[1] 36K-BRAM; [2] The LUT utilisation with a 250 MHz operating clock; [3] Approximate value from figures presented in literature; [4] Single-shot precision = RMS precision/sqrt(2); [5] The value of the used Slice; [6] The value of the used CLB; [7] Averaged valid precision; [8] The averaged peak-to-peak DNL and INL results of the multi-channel hybrid TDC; [9] The peak-to-peak DNL and INL; [10] The percentage of used DSPs in the target device.

Table 2-5 Comparisons of different FPGA-TDC architectures

Architecture	Pros	Cons
TDL-TDC	● Easy implementation	● Significant hardware utilisation
LSPR-TDC	● Better resolution[1]	● Significant routing resources utilisation
RO-TDC	● Low hardware utilisation	● High power consumption
GCO-TDC	● Low hardware utilisation	● Worse resolution[2]
Loop-pulse-shrink-TDC	● Better resolution[1] ● High linearity ● Low hardware utilisation	● Long dead time
DSP-TDC	● Better resolution[1]	● Limited DSP resources available in FPGAs
VRO-TDC	● Better resolution[1] ● Low hardware utilisation	● Long deadtime ● Require manual adjustment for ROs

[1] Offer a better resolution than TDL-TDC when implemented in the same FPGA; [2] Offer a worse resolution than TDL-TDC when implemented in the same FPGA.

Compared with the TDL-TDC in Ref. [83], the LSPR-TDC in Ref. [129] achieves a similar resolution despite being implemented in a less advanced platform, indicating that this architecture can deliver a better resolution than TDL-TDCs when implemented in the same device. However, the LSPR-TDC consumes routing resources significantly, potentially leading to routing congestion in the multi-channel design. In contrast, the RO-TDC in [89] and the GCO-TDC in [110] have low hardware utilisation. However, these two designs offer limited resolutions. The RO-TDC also suffer from significant power consumption due to the unstopped ROs [89]. The loop-shrink-TDC in Ref. [96] and VRO-TDC in Ref. [131] can achieve low hardware utilisation and high resolution simultaneously. However, they both have a long dead time, and the VRO-TDC also requires manual adjustments for ROs' periods, increasing the design complexity.

Although several architectures were proposed for different requirements, compared with ASIC-TDCs, challenges for fine-time measurement, including nonlinearity (such as the designs in Ref. [127] and [113]) and limited resolutions (such as the design in Ref. [110]), still exist when designing FPGA-TDCs. Hence, the following chapter will propose methods and architectures to address these challenges. Besides, the multi-channel design is an increasing trend for modern FGPA-TDCs. Therefore, the hardware utilisation of the proposed designs is also a concern in this thesis.

Chapter 3 Automatic calibration TDCs implemented in a Zynq SoC

3.1 Motivation

TI measurements are widely used in numerous commercial applications, including LiDAR for autonomous driving [132], [133], 3-D reconstruction [134], [135], surveying [136], PET [137], and FLIM [41], [42]. Besides, prototyping is necessary before large-scale production. Therefore, FPGA-TDCs, characterising short development cycles and low development costs, are suitable for pre-production prototype verification for the aforementioned applications.

FPGA-based TDL-TDCs can achieve picosecond resolutions due to advanced CMOS manufacturing technologies. However, FPGA-TDCs have significant nonlinearity compared with ASIC-TDCs, due to the uneven propagation delays of delay cells. Hence, several methods were proposed to enhance the linearity of FPGA-based TDL-TDCs. For example, the tuned-TDL [118] method utilises modifying the output pattern of carry elements for better linearity. Despite these efforts, the achieved linearity is still uncompetitive compared with that of ASIC-TDCs. Hence, BRAM-based bin-width calibration methods, such as the mixed-calibration [83] and mixed-binning methods [128], were proposed. These two methods utilise BRAMs to store calibration factors to remap and calibrate bin widths. FPGA-TDCs with these methods can achieve competitive linearity. However, calibration factors stored in BRAMs require manual pre-calculation channel-by-channel based on SCDTs, which is time-consuming, particularly for multi-channel TDCs. Hence, a suitable calibration method for FPGA-TDCs is required to facilitate multi-channel prototyping applications.

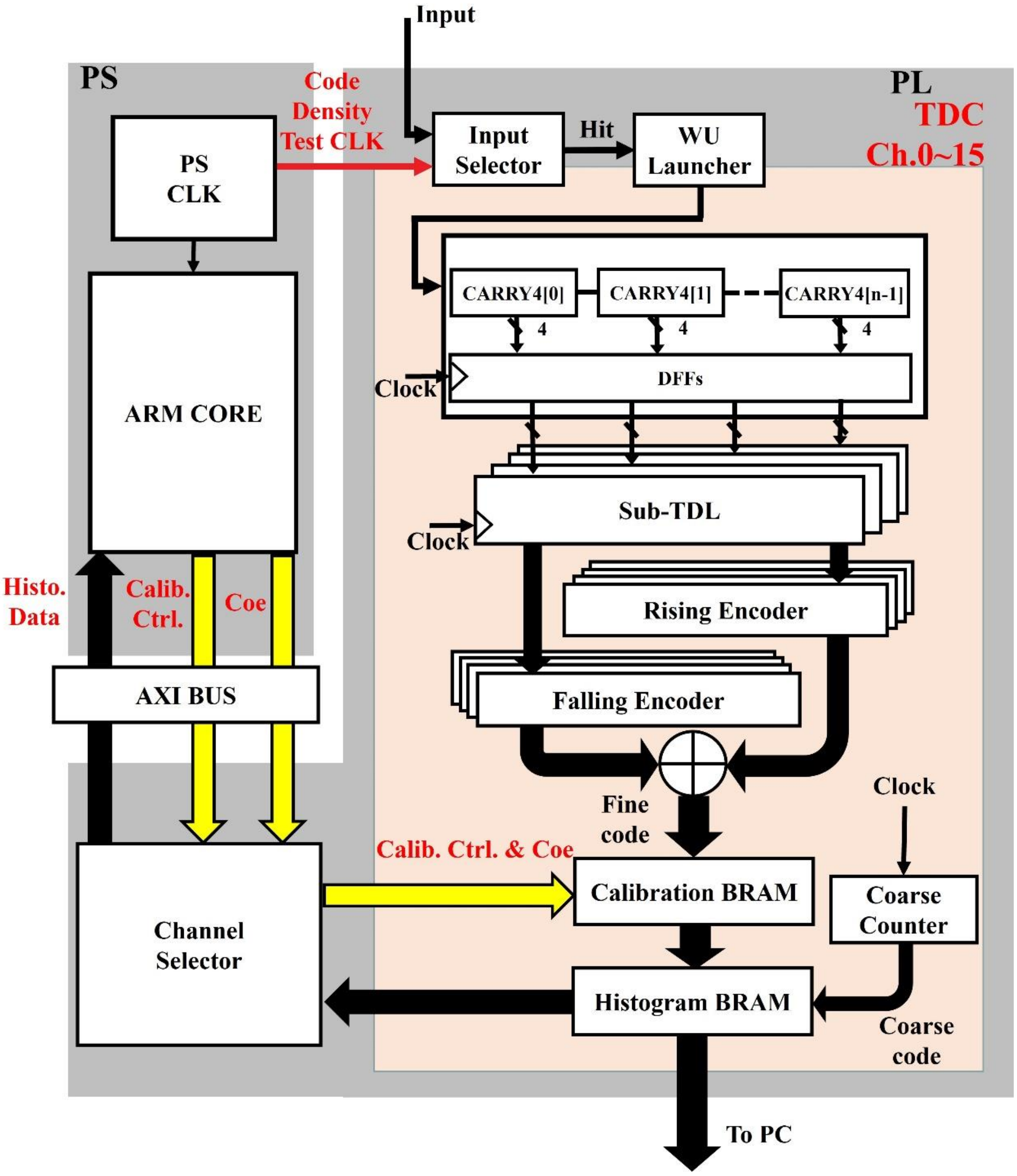


Figure 3.1 Block diagram of the proposed TDC system.

This chapter proposes an automatic calibration (Auto. Cali.) FPGA-TDC and a weighted histogram calibration method. The designed TDC is implemented in a Zynq-7000 SoC and utilises the ARM processor to calibrate all TDC channels without manual intervention.

3.2 Architecture and design

As shown in Fig. 3.1, the TDL, cascaded by CARRY4s, is the cornerstone of the proposed TDC. And the system, implemented in a Zynq-7000 SoC (XC7Z020, ZedBoard development board [138]), includes two components: the programmable logic (PL, which is equivalent to an Artix-7 FPGA) and the processing system (PS, a dual-core ARM Cortex-9) [139]. The TDCs shown in Fig. 3.1 (including TDLs, encoders, calibration modules, and histogram modules) are implemented by PL, while PS is dedicated to calculating remapping addresses and bin-width calibration factors. The outputs from each sub-TDL are first converted into one-hot codes by positioning "0-1" and "1-0" patterns. Then, these one-hot codes are converted into corresponding binary codes for the final result. Moreover, the channel selector is designed to select and transfer data between the TDC channels and the ARM core, facilitating multi-channel applications. And the advanced eXtensible interface (AXI) serves as the data bus to communicate between PL and PS [140].

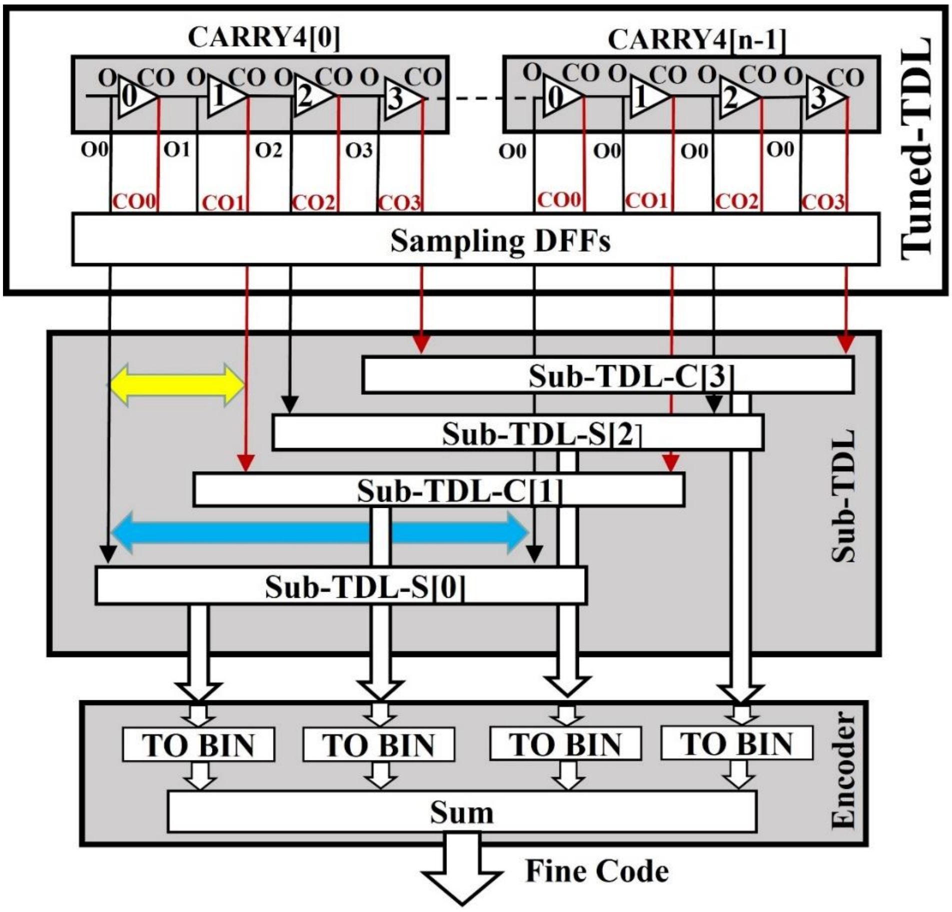


Figure 3.2 Block diagram of the tuned-TDL and sub-TDL.

This design also uses the WU method [115], the sub-TDL method [83], and the tuned-TDL method [118] to further enhance the TDC's performance.

3.2.1 CARRY4 and tuned-TDL architecture

As shown in Fig. 3.2, CARRY4s are connected to form the TDL, and each CARRY4 consists of four cascaded carry elements. Each carry element has two outputs, including a direct output (named CO in Fig. 3.2) and an XOR-gated output (named O in Fig. 3.2). Ref. [118] proposed that modifying TDL's output patterns can enhance carry-chain-based TDL-TDC's linearity. Similar experiments were also conducted in the implemented device to find the best output pattern. Table 3-1 summarises the test results, indicating that the pattern "O-CO-O-CO" performs the best linearity.

Table 3-1 Output-pattern comparisons

Pattern (LSB = 9.83 LSB)	DNL (LSB) [min, max], peak-to-peak	INL (LSB) [min, max], peak-to-peak
CO-CO-CO-CO	[-0.99, 4.32], 5.32	[-5.44, 4.17], 9.61
O-O-O-O	[-0.91, 3.26], 4.17	[-5.32, 4.71], 10.03
CO-O-CO-CO	[-0.98, 3.25], 4.23	[-5.29, 5.19], 10.48
CO-O-CO-O	[-0.95, 5.07], 6.02	[-7.40, 5.28], 12.68
O-CO-O-O	[-0.97, 3.26], 4.24	[-7.18, 5.27], 12.45
O-CO-O-CO	[-0.89, 2.94], 3.84	[-5.73, 4.96], 10.69

3.2.2 WU method and sub-TDL architecture

As shown in Fig 3.3, the WU launcher is mainly constructed by a LUT, and it respectively generates a rising and falling transition when the TDC's input port receives a hit signal. Then, these two transitions ("0-1" from low logic level to high logic level and "1-0" from high logic level to low logic level) propagate along the TDL until the sampling clock toggles from "0" to "1". Simultaneously, the outputs from every TDL tap are sampled by DFFs. With two transitions, the TDC conducts two measurements for the same TI within a single sampling clock period. Therefore, as shown in Fig. 3.3, the resolution of the WU-TDC is calculated as [82]:

$$LSB_{WU} = \frac{T}{N_r + N_f} = \frac{LSB_r \times LSB_f}{LSB_r + LSB_f}, \tag{3.1}$$

where N_r and N_f represents the bin numbers of the plain TDC when inputting a rising and falling transition. Several studies have indicated that the WU method can effectively enhance the TDL-TDC's resolution [130], [141]. However, it also deteriorates bubble problems, causing encoding failures[115], [142], [143]. The bin realignment method [144] is effective in removing bubbles when there is only one transition propagating along the TDL. However, in the WU-TDC, this method fails to achieve the same effectiveness due to the difference in propagation speed between rising and falling transitions [82].

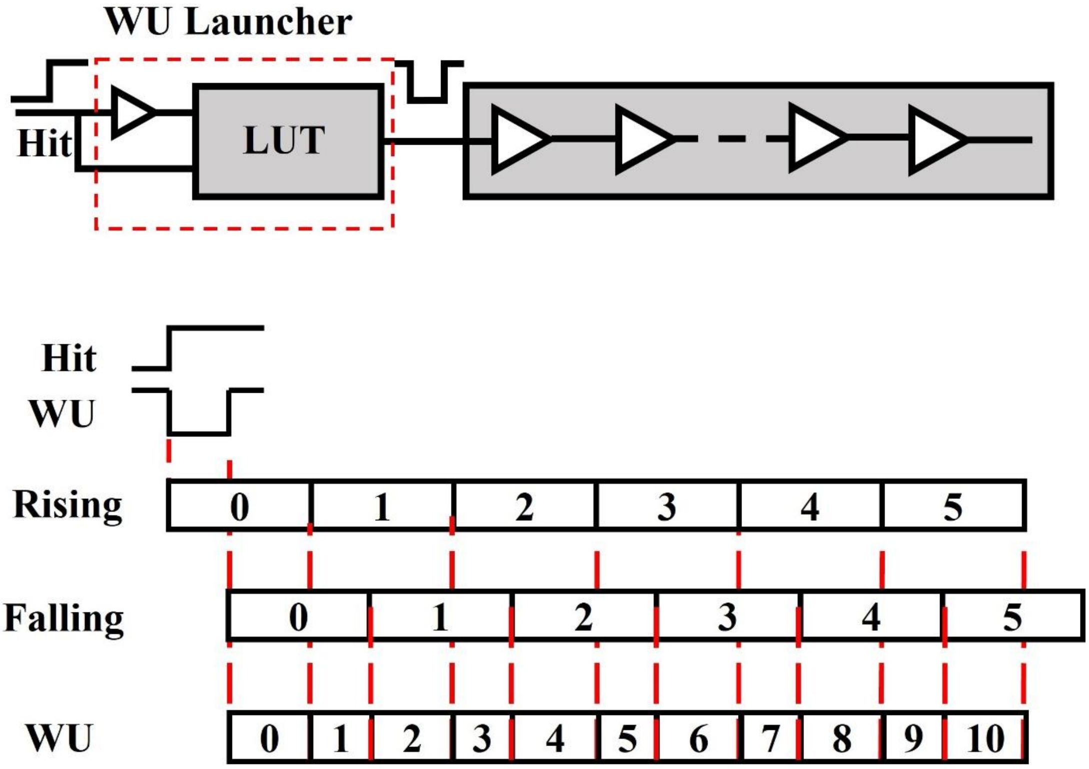


Figure 3.3 Concept of the WU method.

Therefore, the sub-TDL method [83] (also known as the decomposition method [124]) was employed in this design to remove bubbles. As shown in Fig. 3.2, this method extends time intervals between taps by down-sampling taps within the same slice as the CARRY4s to obtain a bubble-free output. In TDL-TDCs, uneven propagation delays and significant clock skews cause bubbles [124], [126]. Ref. [145] reports that the Kintex-7 FPGA has a maximum clock skew of approximately 19 ps within a clock region. And the typical propagation delay between two adjacent delay cells (highlighted in yellow in Fig. 3.2) is approximately 7-20 ps in 7-series FPGAs [125], [146]. Hence, as shown in Fig. 3.2, the TDL's output is split into four groups at a fixed interval to minimise the impacts of uneven propagation delays and clock skews, thereby removing bubbles. Besides, the sub-TDL method is side-effect-free since the TDC's resolution is maintained by interpolations with four sub-TDLs.

3.2.3 Weighted histogram calibration method

The tuned-TDL method enhances linearity without extra hardware utilisation. However, the TDC with the tuned-TDL still suffers from ultra-wide (> 2 LSB) and ultra-narrow bins (< 0.2 LSB). To address these issues, Chen and Li proposed the mixed-calibration method, achieving linearity competitive with ASIC-TDCs [83]. This method encompasses two steps: bin compensation and width calibration. Both steps require SCDTs to determine remapping addresses and bin-width calibration factors, respectively. In the bin compensation stage, the actual bin is remapped to corresponding ideal bins by compensation factors (BCF_m and BCF_c) shown in Fig. 3.4. Then, the bin widths of the remapped bins are further calibrated by bin-width calibration factors (WCF_m and WCF_c shown in Fig. 3.5) in the width calibration stage. Although the mixed calibration method is effective, it has limitations in addressing ultra-wide bins spanning more than 2 ideal bins. For example, missing codes are introduced (Bin [$M+1$] highlighted in red in Fig. 3.4) when coping with the ultra-wide bin (Bin [n] highlighted in blue in Fig. 3.4). Besides, the two-step calculations are unsuitable for Auto. Cali. function due to the communication latency between PL and PS. Hence, a single-step weighted histogram calibration method was proposed and implemented in the SoC.

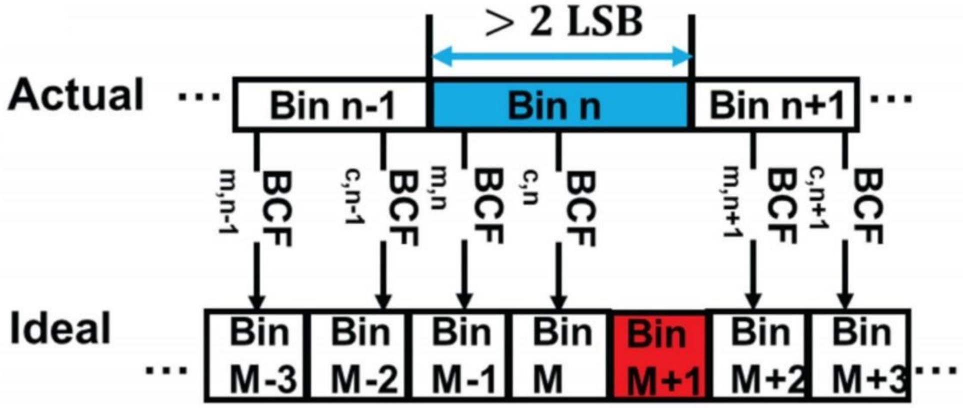


Figure 3.4 Bin compensations in the mixed calibration method.

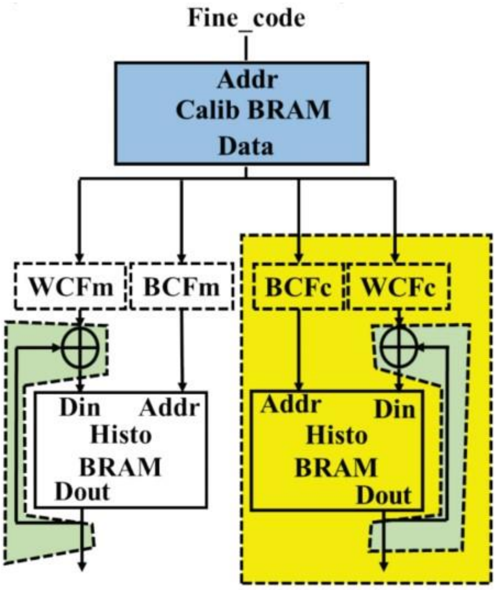


Figure 3.5 Hardware implementation of the mixed calibration method. [128]

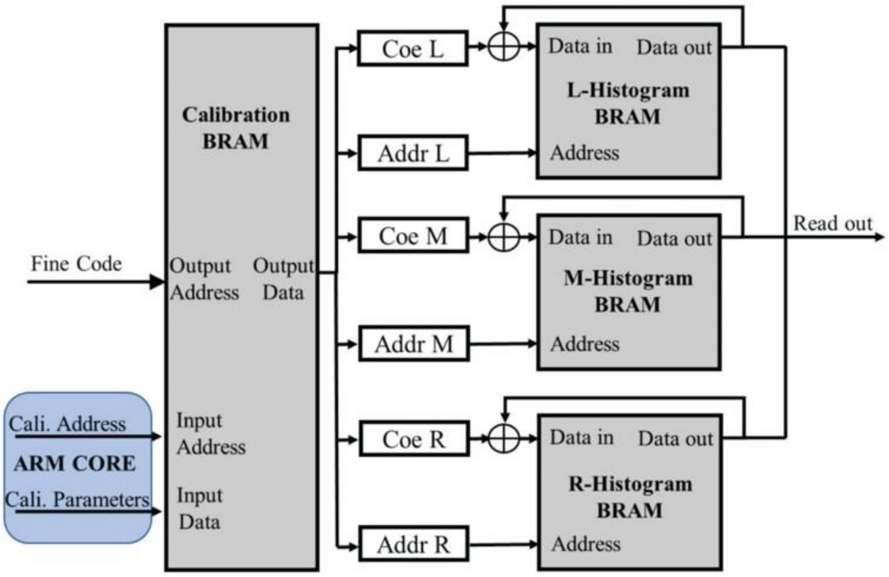


Figure 3.6 Hardware implementation of the weighted histogram calibration method.

Fig. 3.6 shows the hardware implementation of the proposed weighted histogram calibration method. Adders in Fig. 3.6 operate as accumulators to achieve the histogram function. The proposed method, similar to the mixed calibration method [83], consists of two primary steps: bin compensation and width calibration. However, unlike the mixed calibration method, the proposed method stores three pairs of remapping addresses (*Addr L*, *Addr M*, and *Addr R*) and bin-width calibration factors (*Coe L*, *Coe M*, and *Coe R*) in the calibration BRAM module, as shown in Fig. 3.6, to effectively address ultra-wide bins.

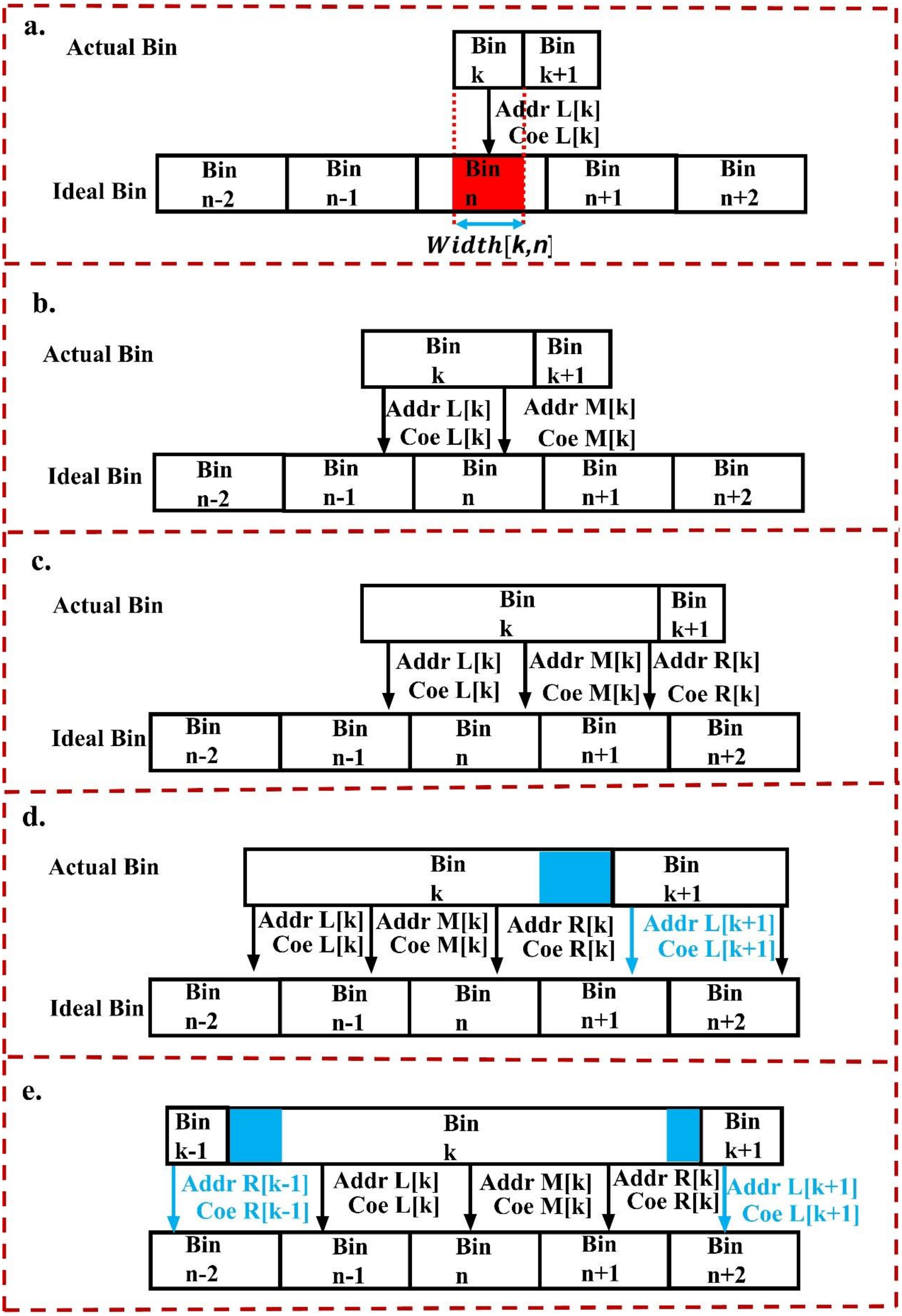
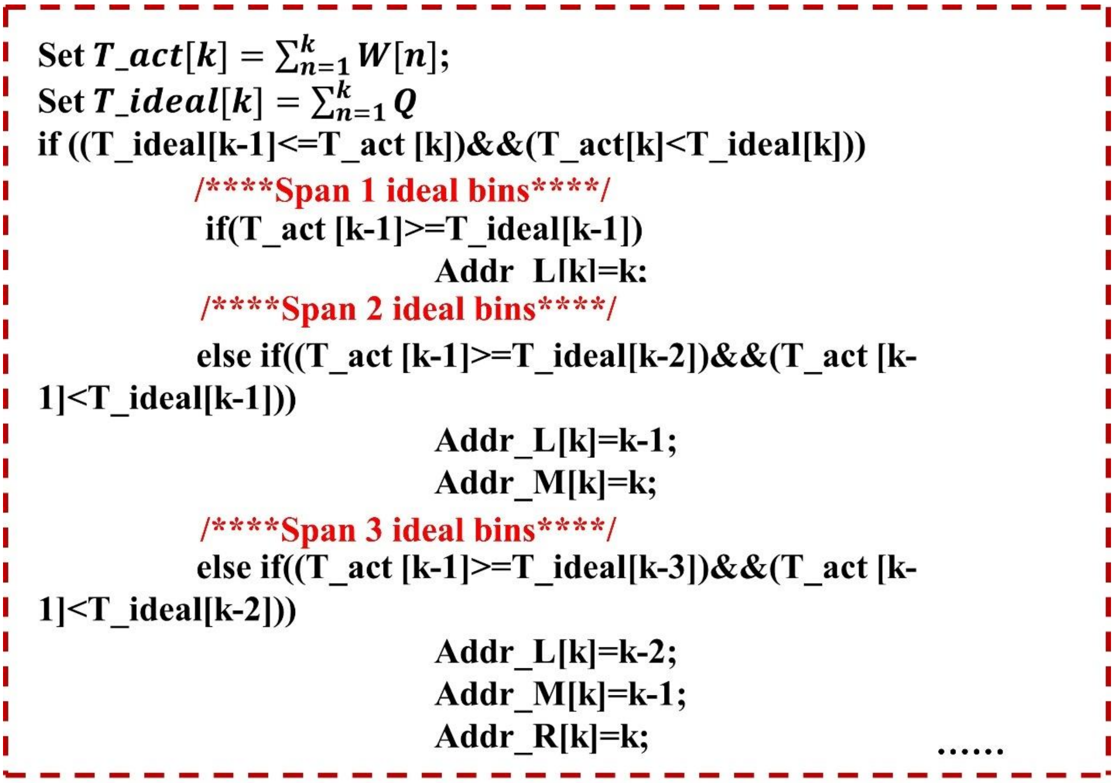


Figure 3.7 Bin compensations when actual bins span (a) 1 ideal bins, (b) 2 ideal bins, (c) 3 ideal bins, (d) 4 ideal bins, and (e) 5 ideal bins.



```
Set $T\_act[k] = \sum_{n=1}^{k} W[n]$;
Set $T\_ideal[k] = \sum_{n=1}^{k} Q$
if ((T_ideal[k-1]<=T_act [k])&&(T_act[k]<T_ideal[k]))
            /****Span 1 ideal bins****/
            if(T_act [k-1]>=T_ideal[k-1])
                        Addr  L[k]=k:
            /****Span 2 ideal bins****/
            else if((T_act [k-1]>=T_ideal[k-2])&&(T_act [k-1]<T_ideal[k-1]))
                        Addr_L[k]=k-1;
                        Addr_M[k]=k;
            /****Span 3 ideal bins****/
            else if((T_act [k-1]>=T_ideal[k-3])&&(T_act [k-1]<T_ideal[k-2]))
                        Addr_L[k]=k-2;
                        Addr_M[k]=k-1;
                        Addr_R[k]=k;                ……
```

Figure 3.8 Pseudocodes for compensation factor calculations.

Fig. 3.7a – 3.7e shows the following five potential cases for bin compensation:

Case A: Span 1 ideal bin.

Case B: Span 2 ideal bins.

Case C: Span 3 ideal bins.

Case D: Span 4 ideal bins.

Case E: Span 5 ideal bins.

And Fig. 3.8 shows the pseudocodes to calculate compensation factors. Ideally, the proposed method, utilising three pairs of factors, can effectively tackle the ultra-wide bin up to spanning 5 ideal bins, and Fig. 3.7e shows this case. In Fig. 3.7e, Bin [*k*] is an ultra-wide bin which spans 5 ideal bins. This bin is adjacent to two bins: Bin [*k-1*] and Bin [*k+1*]. Bin [*k*] can be only remapped to Bin [*n-1*], Bin [*n*] and Bin [*n+1*] with three pairs of factors. However, using the factors Addr R [*k-1*] and Addr L [*k+1*], the actual bins Bin [*k-1*] and Bin [*k+1*] can cover the ideal bins Bin [*n-2*] and Bin [*n+2*]. Therefore, the missing codes caused by ultra-wide bins (overlapping less than 6 ideal bins) can be effectively addressed by the proposed method, indicating this method is more efficient than the mixed calibration method.

Width-calibration factor calculations follow similar principles in Fig. 3.7. In Case A, only *Coe L* is used and can be calculated as:

$$Coe\ L[k] = \frac{Width[k,n]}{W[k]}, \tag{3.2}$$

where $Width[k, n]$ is a portion of Bin [k] in the actual TDL, which should be in Bin [n] in the ideal TDL (highlighted in red in Fig. 3.7a).

In Case B, Bin [k] is remapped to Bin [n-1] and Bin [n], as shown in Fig. 3.7b. Hence, *Coe L* and *Coe M* are used and calculated as:

$$Coe\ L[k] = \frac{Width[k,n-1]}{W[k]}, \tag{3.3}$$

and

$$Coe\ M[k] = \frac{Width[k,n]}{W[k]}. \tag{3.4}$$

Case C uses all width factors and they are calculated as:

$$Coe\ L[k] = \frac{Width[k,n-1]}{W[k]}, \tag{3.5}$$

$$Coe\ M[k] = \frac{Width[k,n]}{W[k]}, \tag{3.6}$$

and

$$Coe\ R\ [k] = \frac{Width[k,n+1]}{W[k]}. \tag{3.7}$$

However, in Case D, *Coe L* [*k+1*] is also used to calibrate the bin width covered by Bin [*k*] (highlighted in blue in Fig. 3.7d) and hence is calculated as:

$$Coe\ L\ [k+1] = \frac{Width[k+1,n+1]}{W[k+1]} + \frac{Width[k,n+1]}{W[k+1]}. \tag{3.8}$$

Similarly, *Coe R* [*k-1*] and *Coe L* [*k+1*] are used to calibrate bin widths covered by Bin [*k*] (highlighted in blue in Fig. 3.7e) in Case E, and they are respectively calculated as:

$$Coe\ L\ [k+1] = \frac{Width[k+1,n+2]}{W[k+1]} + \frac{Width[k,n+2]}{W[k+1]}, \tag{3.9}$$

and

$$Coe\ R\ [k-1] = \frac{Width[k-1,n-2]}{W[k-1]} + \frac{Width[k,n-2]}{W[k-1]}, \tag{3.10}$$

The remapping addresses and bin-width calibration factors can be calculated with a single round SCDTs in the proposed method, as shown in Fig. 3.9a. In contrast, the mixed calibration method shown in Fig 3.9b requires two rounds. Therefore, the proposed weighted histogram calibration method is more appropriate for Auto. Cali..

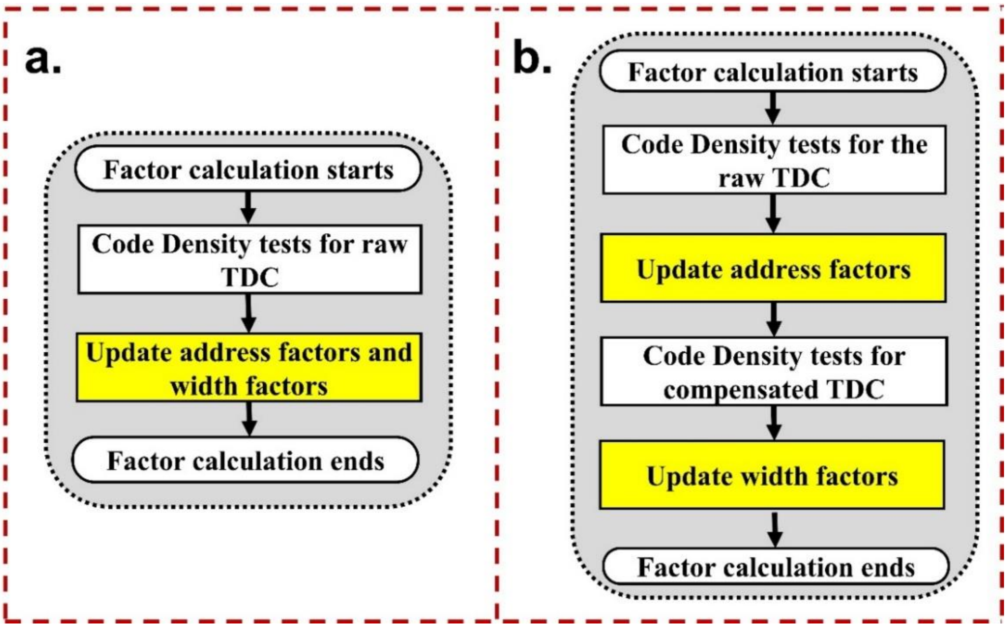

a.
Factor calculation starts
Code Density tests for raw TDC
Update address factors and width factors
Factor calculation ends
b.
Factor calculation starts
Code Density tests for the raw TDC
Update address factors
Code Density tests for compensated TDC
Update width factors
Factor calculation ends

Figure 3.9 Flow diagrams of (a) proposed weighted histogram calibration method and (b) mixed calibration method [83].

3.2.4 Automatic calibration

As shown in Fig. 3.1, TDCs are implemented in PL, and an ARM processer in PS calculates compensation and bin-width calibration factors. Fig. 3.10 shows the workflow of the proposed Auto. Cali. WU-TDC, and the whole workflow is split into two stages: the initial stage and the measurement stage. The initial stage requires SCDTs to quantify the bin width and then stores the results in the Histogram BRAM shown in Fig. 3.6. After that, the test results are read out by ARM processor through AXI data bus to calculate compensation and bin-width calibration factors. Finally, the ARM processor loads these factors to the calibration BRAM shown in Fig. 3.6 and completes the initial stage. In the measurement stage, compensation and bin-width calibration factors are indexed by raw output from the encoder and fetched from the calibration BRAM, then used as Histogram BRAMs' addresses and inputs of accumulators (shown in Fig. 3.6), respectively.

The proposed Auto. Cali. architecture frees the system from manual calibrations. Hence, it is suitable for multi-channel prototype verification, especially for engineers and researchers unfamiliar with TDCs. Besides, it also has the potential for commercial applications, such as being integrated into commercial LiDAR systems.

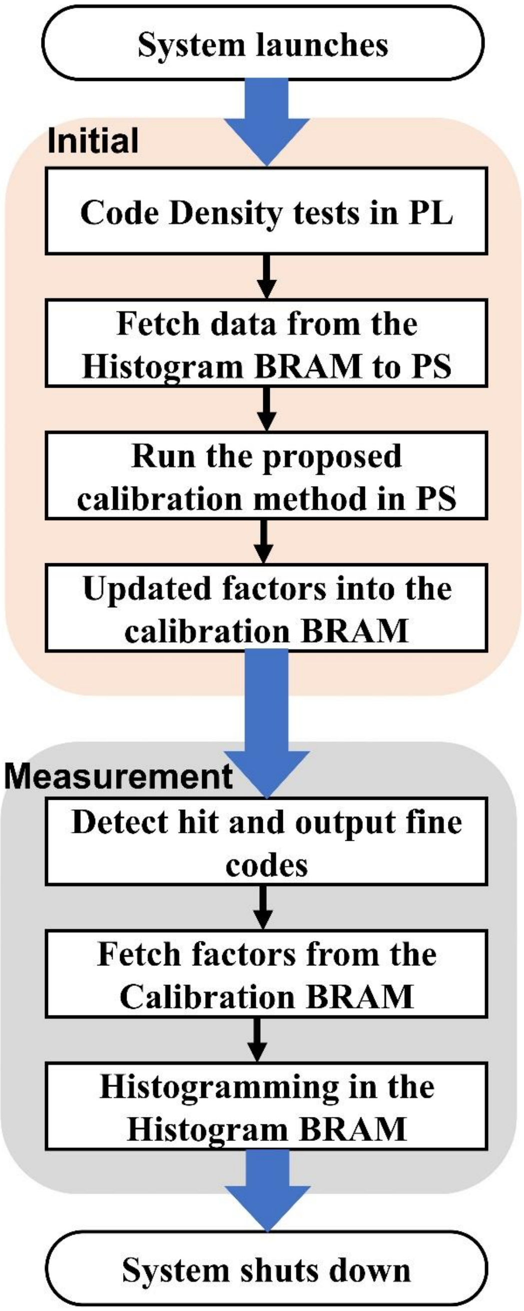

System launches
Initial
Code Density tests in PL
Fetch data from the Histogram BRAM to PS
Run the proposed calibration method in PS
Updated factors into the calibration BRAM
Measurement
Detect hit and output fine codes
Fetch factors from the Calibration BRAM
Histogramming in the Histogram BRAM
System shuts down

Figure 3.10 Workflow of the automatic calibration TDC.

3.3 Experimental results

The proposed TDC was implemented in a Zedboard [138], as shown in Fig. 3.11, and experiments were conducted to evaluate the TDC's performance. The system clock and the random input (uncorrelated to the system clock) for SCDTs are from two independent low-jitter crystal oscillators (Fox-767) in the ZedBoard [138]. The TDC's system clock operates at 300 MHz, derived from an on-board crystal oscillator running at 100 MHz. Moreover, both the voltage and the temperature were maintained throughout the experiments.

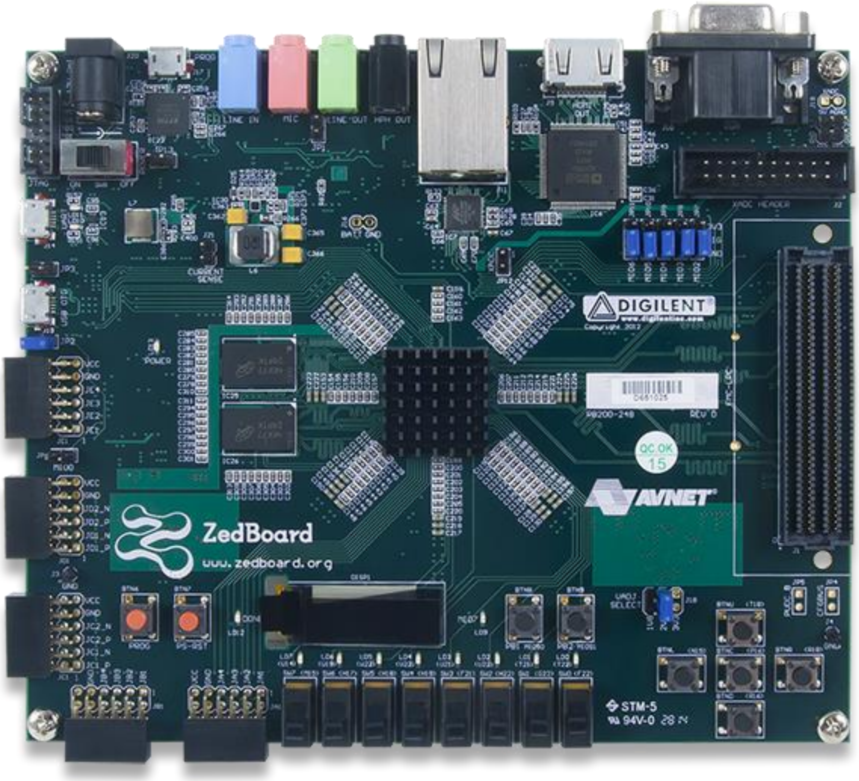


Figure 3.11 ZedBoard [138].

3.3.1 Linearity and bin-width distribution

DNL, INL and their standard deviations (σ_{DNL} and σ_{INL}) are used to evaluate the Auto. Cali. WU-TDC's linearity. Besides, the equivalent bin width ω_{eq} and its standard deviation σ_{eq} are also used to evaluate nonlinearity's overall impact on measurements [147]. They are respectively calculated as [147]:

$$\sigma_{eq}^2 = \sum_{i=1}^{N} \left(\frac{W[i]^2}{12} \times \frac{W[i]}{W_{total}} \right), \tag{3.11}$$

and

$$\omega_{eq} = \sigma_{eq}\sqrt{12} = \sqrt{\sum_{i=1}^{n} (\frac{W[i]^3}{W_{total}})}, \tag{3.12}$$

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

Table 3-2 Linearity comparisons between the uncalibrated TDC and calibrated TDC

	Tuned & Sub-WU	**AC-WU**
LSB (ps)	**9.83**	
DNL (LSB)	[-0.93,2.98]	**[-0.14,0.16]**
DNL_{pk-pk}(LSB)	3.91	**0.30**
σ_{DNL} (LSB)	0.86	**0.04**
INL (LSB)	[-6.52,5.53]	**[-0.25,0.42]**
INL_{pk-pk} (LSB)	12.05	**0.67**
σ_{INL} (LSB)	2.79	**0.13**
ω_{eq} (ps)	19.76	**9.85**
σ_{eq} (ps)	5.70	**2.84**

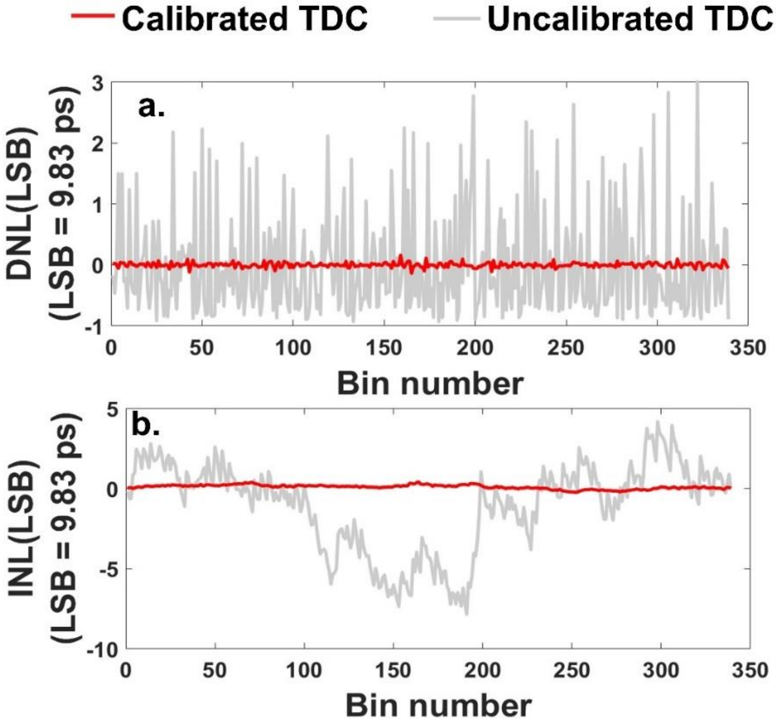


Figure 3.12 (a) DNL and (b) INL plots of the calibrated and uncalibrated TDCs.

Table 3-2 summarises the experimental results, and Fig. 3.12 shows the DNL and INL plots. The proposed TDC's DNL and INL have significant improvements compared with the uncalibrated TDC. After calibration, DNL_{pk-pk} and INL_{pk-pk} are enhanced by 13-fold (from 3.91 LSB to 0.30 LSB) and 18-fold (from 12.05 LSB to 0.67 LSB), respectively. σ_{DNL} is enhanced by 21-fold (from 0.86 LSB to 0.04 LSB), and σ_{INL} is also enhanced by 21-fold (from 2.79 LSB to 0.13 LSB). Moreover, ω_{eq} is enhanced from 19.76 ps to 9.85 ps, and σ_{eq} is enhanced from 5.70 ps to 2.84 ps. Fig. 3.13a and Fig. 3.13b also show bin-width distributions of calibrated and uncalibrated TDCs. As shown in Fig. 3.13b, the calibrated TDC's bin-width distribution is more concentrated than the uncalibrated TDC.

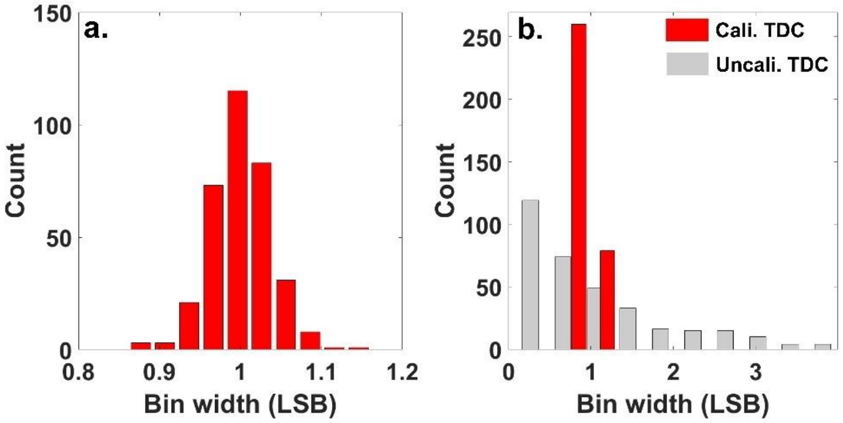


Figure 3.13 (a) Distributions of the calibrated bin widths, and (b) the comparison between calibrated and uncalibrated bin-width distributions when LSB = 9.83 ps.

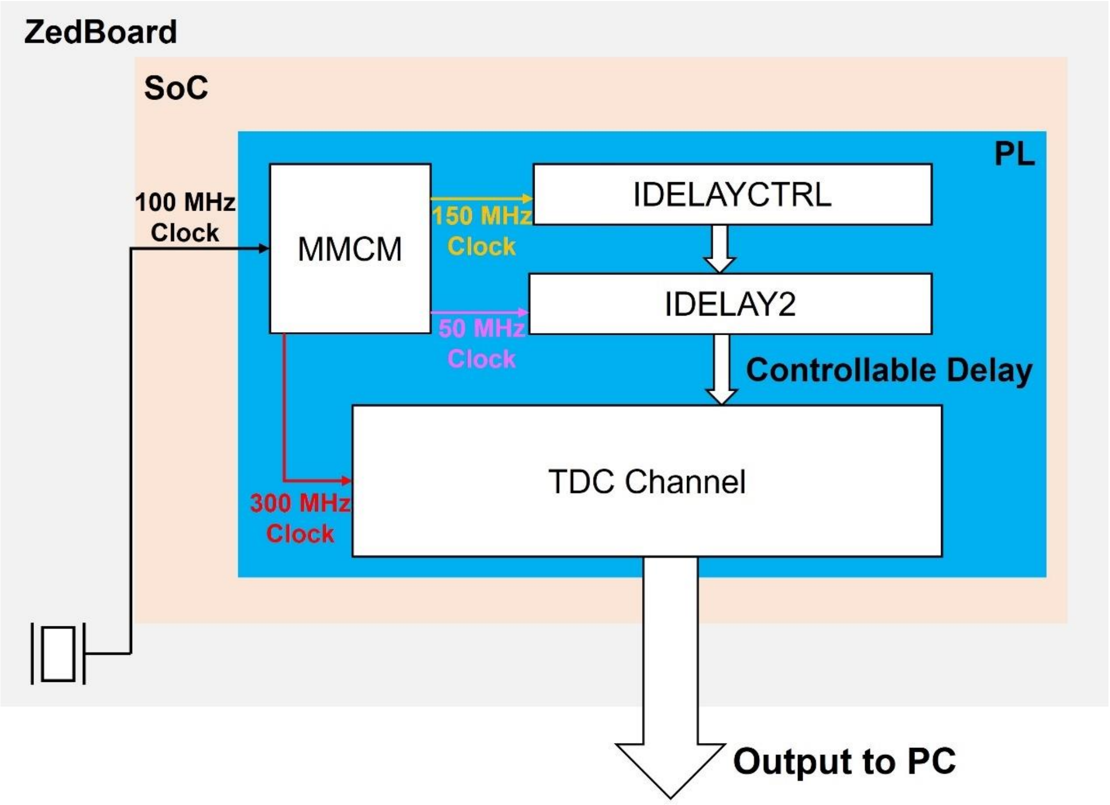


Figure 3.14 Test setup of the time interval test.

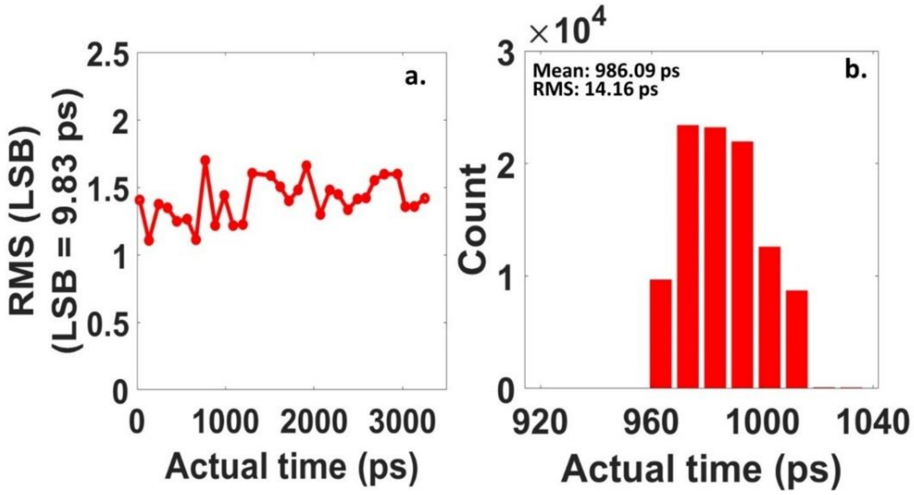


Figure 3.15 (a) TI measurement results and (b) TI histograms when TI = 980 ps.

3.3.2 Time interval tests

The standard deviation of repetitive fixed-TI measurements can evaluate the Auto. Cali. WU-TDC's precision. And the controllable delay between the sampling clock and the input hit was generated by FPGA-inside delay macros (IDELAY2 and IDELAYCTRL) [148] to minimise external jitters and measurement errors. The repetitive fixed-TI was measured using a single channel, as show in Fig. 3.14. Moreover, 30 measurements are conducted, as shown in Fig. 3.15a, to cover the entire sampling period, with each measurement capturing 100000 samples. The standard deviations of each measurement were calculated, and the mean value (13.86 ps) represents the precision of the proposed TDC. Moreover, Fig. 3.15b shows the measurement histogram when TI = 980 ps, achieving a precision of 14.16 ps.

3.3.3 Multichannel implementation and logic resources consumption

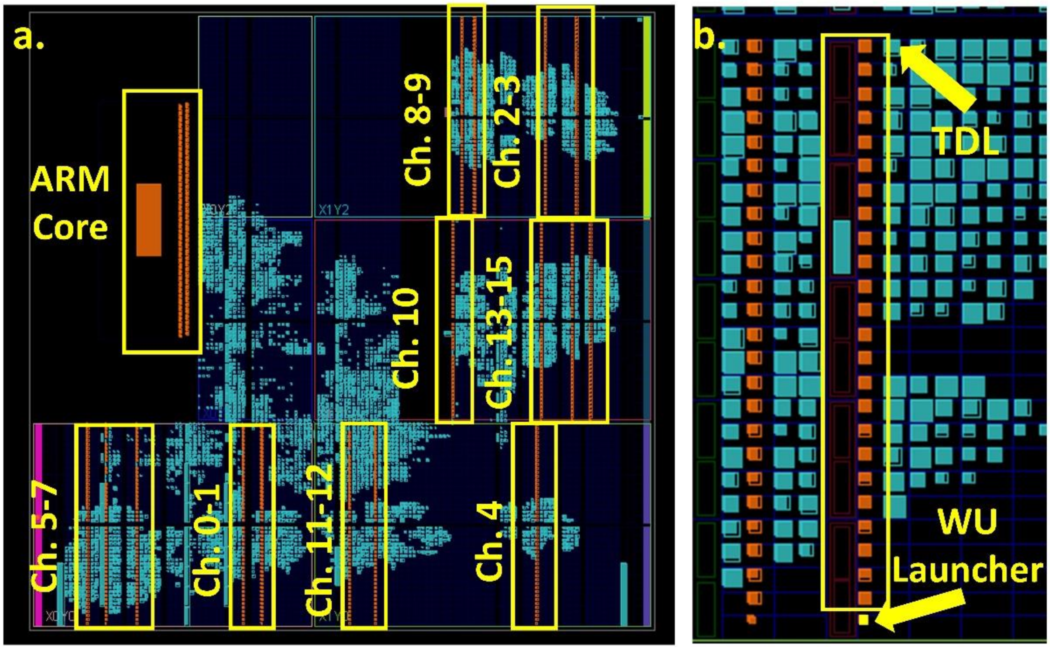


Figure 3.16 Implementation layouts of (a) entire TDC system and (b) a single channel.

Fig. 3.16a shows the implementation layout of the 16-channel TDC system implemented in a Zynq-7000 SoC. Each TDL, which contains 50 CARRY4s, is placed within a clock region to avoid substantial clock skews when crossing clock regions. Besides, as shown in Fig. 3.16b, each WU launcher is placed close to the corresponding TDL to reduce jitters induced by routing resources. Table 3-3 summarises the hardware utilisation of the proposed TDC. A single TDC channel consumes 764 LUTs, 1095 DFFs, and 2 BRAMs. Furthermore, the AXI4 bus also consumes 797 LUTs and 1278 DFFs for communication between PS and PL. The hardware utilisation indicates that the proposed Auto. Cali. WU-TDC is hardware-efficient and appropriate for multi-channel designs. In theory, up to 70 channels can be implemented in this device. However, considering routing congestion, a more realistic target is around 50 channels.

Table 3-3 Hardware utilisation of the proposed TDC

	CARRY4	LUTs	DFFs	BRAM
Available	13300	53200	106400	140
Single-channel	**50** (0.38%)	**764** (1.44%)	**1095** (1.02%)	**2** (1.42%)
16-channel	**800** (6.02%)	**9681** (18.19%)	**15141** (14.23%)	**32** (22.85%)
AXI Bus	**0**	**797** (1.49%)	**1278** (1.20%)	**0**

Code density tests were conducted for all 16 channels, with the linearity performance summarised in Table 3-4. The DNL$_{\text{pk-p}k}$ ranges from 0.21 LSB to 0.69 LSB with an average of 0.38 LSB, and the INL$_{\text{pk-p}k}$ ranges from 0.37 LSB to 0.90 LSB with an average of 0.63 LSB. The DNL$_{\text{pk-pk}}$ and INL$_{\text{pk-pk}}$ indicate good uniformity of the proposed 16-channel TDC.

Table 3-4 Linearity performance of the proposed 16-channel TDC (Unit: LSB)

Channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ave.
DNL_{pk-pk}	0.21	0.34	0.33	0.28	0.30	0.29	0.56	0.69	0.25	0.41	0.25	0.26	0.52	0.27	0.48	0.60	0.38
σ_{DNL}	0.03	0.05	0.05	0.04	0.04	0.04	0.10	0.07	0.04	0.06	0.03	0.04	0.07	0.03	0.06	0.07	0.05
INL_{pk-pk}	0.52	0.55	0.51	0.52	0.67	0.45	0.90	0.86	0.51	0.45	0.57	0.37	0.87	0.62	0.89	0.87	0.63
σ_{INL}	0.10	0.11	0.10	0.12	0.13	0.08	0.17	0.18	0.09	0.08	0.11	0.08	0.19	0.14	0.23	0.26	0.14

Table 3-5 Comparisons between published calibration methods and the proposed calibration method

Ref.-Year	Methods	Multiple steps	Auto/manual Calibration
[68]-10	Bin-by-bin calibration	Single-step	Manual
[120]-17	Bin-width calibration	Single-step	Manual
[83]-19	Mixed calibration	Two-step	Manual
[149]-21	Gain & error calibration	Single-step	Auto
[128]-22	Mixed binning	Single-step	Manual
This work	Weighted histogram calibration with an AC function	Single-step	Auto

Table 3-6 Comparisons of recently published FPGA-TDCs

Ref-Year	Methods	Device Process (nm)	LSB (ps)	ω_{eq} (ps)	Precision (ps)	DNL (LSB)	INL (LSB)	LUT	DFF	BRAM	AXI Bus
[118]-16	Tuned-TDL	28	10.6	N/S [1]	8.13	[-1.00,1.45]	[-1.23,4.30]	577	1641	0	-
		40	10.1	N/S [1]	9.82	[-1.00,1.18]	[-3.03,2.46]	577	1641	0	
		45	16.7	N/S [1]	12.75	[-1.00,1.22]	[-0.70,2.56]	261	787	0	
[120]-17	Tuned-TDL, Direct Histogram Bin-width Cali.	28	10.50	10.55	4.42	[-0.04, 0.04]	[-0.09, 0.04]	N/S [1]	N/S [1]	0	-
[150]-18	Two-Stage Delay Line Loop Shrinking	130	8.50	N/S [1]	42.40	[-0.22, 0.36]	[-0.62, 0.91]	N/S [1]	N/S [1]	0	-
[83]-19	Tuned-TDL, Sub-TDL, Mixed Calibration	28	10.54	10.55	14.59[2]	[-0.05, 0.08]	[-0.09, 0.11]	1145	1916	1.5	-
		20	5.02	5.03	7.80[2]	[-0.12, 0.11]	[-0.18, 0.46]	703	1195	1.5	-
[99]-20	Bidirectional, RO Vernier	65	24.50	N/S [1]	28.00	[-0.20, 0.25]	[0.03, 0.82]	172	986	0	-
[151]-20	PLL Delay Matrix with DDR	40	15.60	N/S [1]	15.60	[-0.18, 0.18] [3]	[-0.16, 0.14] [3]	9886 [4]	N/S [1]	0	-
[149]-21	Slide Scale, Gain & Error cal., Moving Ave.	28	4.88	N/S [1]	2.90~ 8.03	[-0.10, 0.15]	[-0.23, 0.28]	2962	4157	0	-
This Work	**WU-A, Tuned-TDL, Sub-TDL, Auto Cal.**	**28**	**9.83**	**9.85**	**13.86** [2]	**[-0.14, 0.16], 0.38** [5]	**[-0.25, 0.42], 0.63** [5]	**764**	**1095**	**2**	**1278 DFFs** **797 LUTs**

[1] N/S= not specified; [2] Single-shot precision = RMS precision/sqrt(2); [3] Rounding values from data presented in literature; [4] Combinational ALUTs in Altera FPGA; [5] Averaged peak-peak DNL or INL results of the multi-channel TDCs.

3.4 Comparison

Table 3-5 summarises the proposed and other published calibration methods. Moreover, the proposed and recently published FPGA-TDCs are summarised in Table 3-6.

Methods including the bin-by-bin calibration [68], bin-width calibration [120], mixed-calibration [83], and mixed binning methods [128] can enhance the linearity and precision of TDCs. However, these methods require manual calibration channel-by-channel, making them time-consuming for multi-channel applications and unfriendly for users unfamiliar with TDC design. Using signal processing methods, the gain and error calibration [149] has the capability of automatically correcting data. However, this method [149] is complex and consumes more hardware resources per channel than the proposed TDC (the comparison of hardware utilisation is shown in Table 3-6).

Unlike high-resolution (<5 ps) TDCs [82], [108], [113], [121], [145], [149], the Auto. Cali. WU-TDC is designed for a competitive resolution with high linearity in a low-cost SoC device. In comparison to previously published FPGA-TDCs with similar resolutions, the proposed TDC has better linearity and is easy to implement. The designed TDC exhibits similar linearity compared with the PLL delay matrix TDC in Ref. [151]. However, the PLL delay matrix TDC consumes 9886 LUTs, which is 9-fold more than this design. Compared with the two-stage delay line loop shrinking TDC [150] and the bidirectional RO Vernier TDC [99], the proposed TDC also has the advantage of easy implementation since there are no dedicated hardware resources to construct loop architectures in modern FPGAs.

3.5 Conclusion

This chapter proposes a multi-channel Auto. Cali. TDC with the weighted histogram method. Besides, the proposed TDC also combines the WU method [115], the tuned-TDL method [118] and the sub-TDL method [83] for a competitive resolution with high linearity in a Zynq-7000 SoC device. The proposed TDC achieves a 9.83 ps resolution, 0.38 LSB DNL$_{pk-pk}$, and 0.63 LSB INL$_{pk-pk}$. Moreover, a 16-channel TDC system was also implemented and tested, showing good uniformity between channels.

The advantages of this design include: 1) Channel-by-channel manual calibration is required by conventional calibration methods. However, the proposed Auto. Cali. architecture addresses this issue by using the ARM processor inside the Zynq-7000 SoC device. Hence, this design can calibrate TDC channels without manual intervention and is suitable for multi-channel rapid prototype applications. Moreover, it has the potential to be integrated into commercial systems due to its automatic calibration function. 2) Conventional histogram-based calibration methods require two steps to calculate remapping addresses and bin-width calibration factors. However, the proposed method simplifies this procedure and can calculate compensation and bin-width calibration factors in a single step by analysing relative positions between ideal and actual bins. Moreover, the proposed weighted histogram calibration method can effectively address ultra-wide-bin problems (up to spanning 5 ideal bins) with only one more BRAM compared with the mixed calibration method [83], which can only handle ultra-wide bins up to spanning 2 ideal bins.

Chapter 4 Low hardware utilisation and resolution-configurable GCO-TDCs implemented in multiple FPGAs

4.1 Motivation

In TDCs, interpolations utilise the propagation delays of delay cells and routing resources to achieve a higher resolution than clock-driven counters [108], [146], [152]. The TDL is the most used architecture for FPGA-TDCs because modern FPGAs have carry chains that facilitate the construction of TDLs. For example, Xilinx 6-series and 7-series FPGAs have CARRY4s, while UltraScale and UltraScale+ FPGAs have CARRY8s [18], [153]. The propagation delays of carry elements determine the resolutions of TDL-TDCs, and the resolution can be as fine as 10 ps or better [120], [142] in 7-series FPGAs and 5 ps or better in UltraScale FPGAs [82], [83]. Besides, implemented in more advanced FPGAs, a longer TDL is usually required to cover the entire sampling period due to the improved propagation delays of carry elements. For example, 200 carry elements (50 CARRY4s) are used to construct the TDL to cover a sampling period of around 1.4 ns in a Kintex-7 FPGA [125], and 592 carry elements are used to construct the TDL in a Kintex-UltraScale FPGA to cover a sampling period of 2 ns [128]. When implementing multi-channel designs, the consumption of carry elements becomes significant. For example, 31.26% of CARRY8s (75776 carry elements) are used for the 128-channel TDC in Ref. [128]. Besides, both multi-channel TDCs and signal-processing modules are integrated into FPGAs in highly integrated systems such as LiDAR systems in Ref. [132] and [154]. Hence, as basic units for arithmetic operations, carry elements should be used efficiently instead of mainly as delay cells.

Wu and Xu [109] proposed the GCO-TDC, which utilises LUTs rather than carry elements as delay cells. Besides, GCO-TDCs output Gray codes directly, avoiding complex encoding that consumes numerous logic resources in TDL-TDCs, as shown in

Fig. 4.1. Hence, the GCO-TDC has extremely high hardware utilisation efficiency, with one TDC channel consuming only eight LUTs and DFFs [109]. However, this design is limited by the resolution (256 ps) and has 1.25 LSB $\mathrm{DNL}_{\text{pk-pk}}$. To enhance linearity, Machado *et al.* proposed manual routing, improving $\mathrm{DNL}_{\text{pk-pk}}$ to 0.76 LSB [111]. However, the resolution of this design deteriorates to 380.9 ps [111]. For a better resolution, Araújo *et al.* proposed a double-sampling GCO-TDC [110], achieving a resolution of 69 ps. This resolution (69 ps) is acceptable since most time-resolved applications, including LiDAR, PET, and FLIM, require resolutions from 50 ps to 250 ps [4], [155]. However, the linearity of the TDC in Ref. [110] decreases (1.76 LSB $\mathrm{DNL}_{\text{pk-pk}}$), deteriorating the TDC's overall performance.

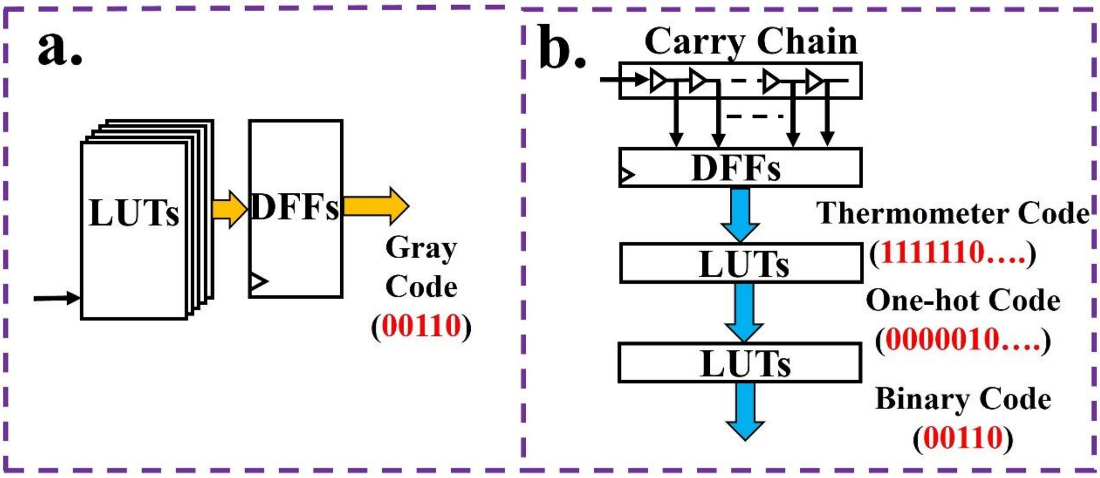


Figure 4.1 Encoding comparison between (a) GCO-TDC and (b) TDL-TDC.

Moreover, most previously published FPGA-TDCs [99], [120], [142] and the aforementioned GCO-TDCs [109], [110], [111] can only offer a fixed resolution, which is inconvenient if application requirements change. Therefore, TDCs in Ref. [128], [156] and [157] were designed with configurable resolutions for broader applications. However, these designs require manual configuration channel-by-channel when changing resolutions, which is time-consuming and challenging for users unfamiliar with FPGA-TDCs. TDCs in Ref. [149] and Chapter 3 can calibrate automatically, but their resolutions are fixed. Therefore, an automatically calibrated TDC with configurable resolutions is highly required for general applications.

This chapter proposes a multi-channel, automatically calibrated GCO-TDC with configurable resolutions for low hardware utilisation and high flexibility. In this design, the innovative sampling matrix is used to significantly improve the GCO-TDC's resolution. Moreover, the VBCM is proposed and implemented in FPGAs for automatic calibration and configurable resolutions.

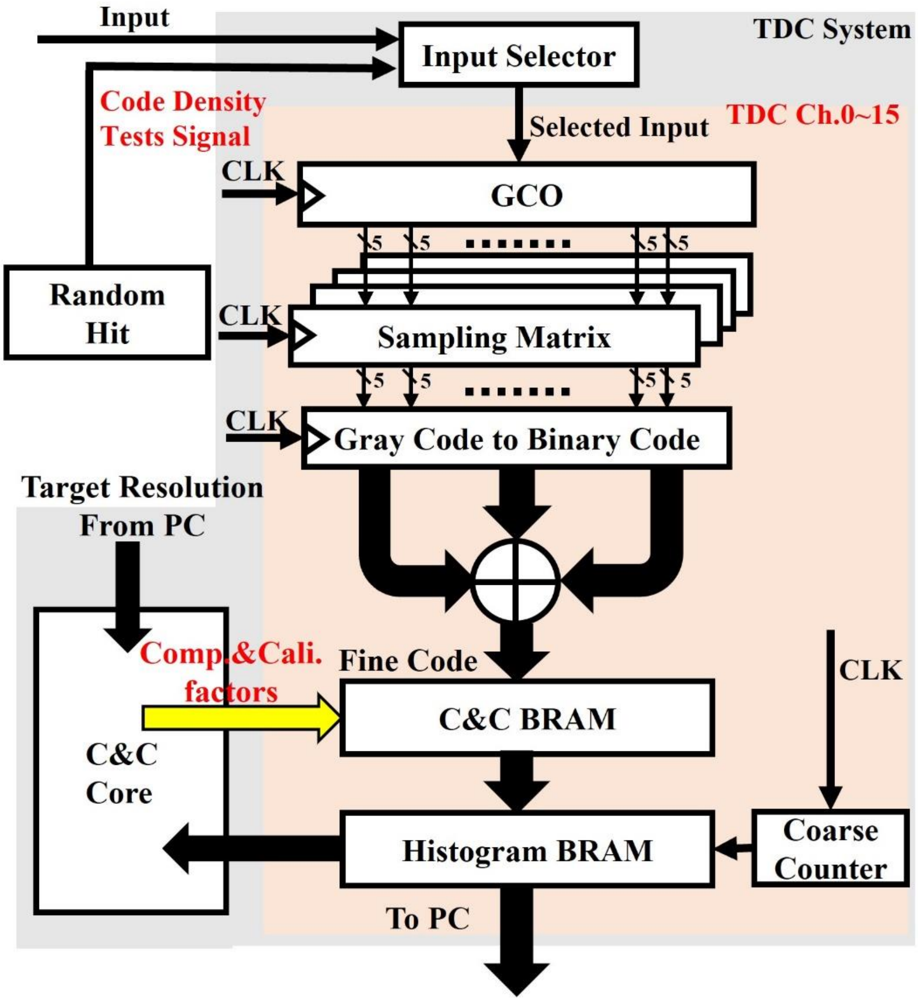


Figure 4.2 Block diagram of the proposed TDC system.

4.2 Architecture and design

Fig. 4.2 shows the proposed TDC's architecture. The GCO is the backbone of the TDC and is responsible for measuring TIs with a coarse counter. Moreover, the proposed TDC includes a sampling matrix, a Gray-code-to-binary-code converter, a histogram BRAM, a compensation and calibration BRAM (C&C BRAM), and a compensation and calibration core (C&C core). Notably, the C&C core is used to calculate compensation and calibration factors (comp.&cali. factors) and works only after accomplishing resolution configuration. The calculated comp.&cali. factors are then loaded into the C&C BRAM to achieve automatic calibration and configurable resolutions.

4.2.1 GCO-TDC

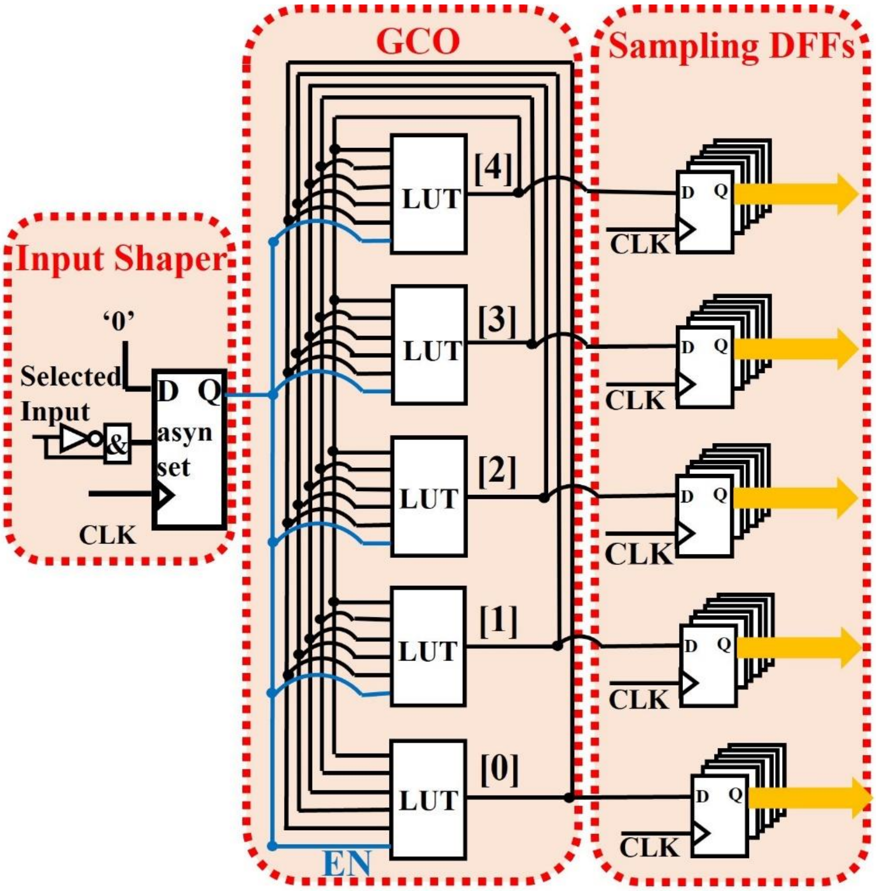


Figure 4.3 Block diagram of the GCO-TDC.

The GCO is shown in Fig. 4.3. Unlike the binary code, where multiple bits may simultaneously change between two successive states, only one bit changes in the Gray code, making the GCO immune to the "race and competition" phenomenon [109]. Therefore, the GCO can be implemented by LUTs in the proposed TDC. Besides, in 7-series and more advanced Xilinx FPGAs, each LUT has a maximum of six input ports [18], [153]. In this design, one of the input ports is connected to the "EN" signal (highlighted in blue in Fig. 4.3) to launch and reset the GCO, with the remaining five input ports responsible for getting feedback from the output ports of LUTs. By using

Vivado primitives to instantiate LUTs, five LUTs can output 5-bit Gray codes. Then, these Gray codes are sampled by the following DFFs to evaluate the measured TI. Due to the ps-level propagation delays of LUTs and routing resources, the GCO has the capability of measuring TI less than one sampling clock period. Moreover, as shown in Fig. 4.3, an "Input Shaper" is designed to launch and reset the GCO respectively after the input hit arrives and the DFFs sample, facilitating the GCO-TDC working with the coarse counter.

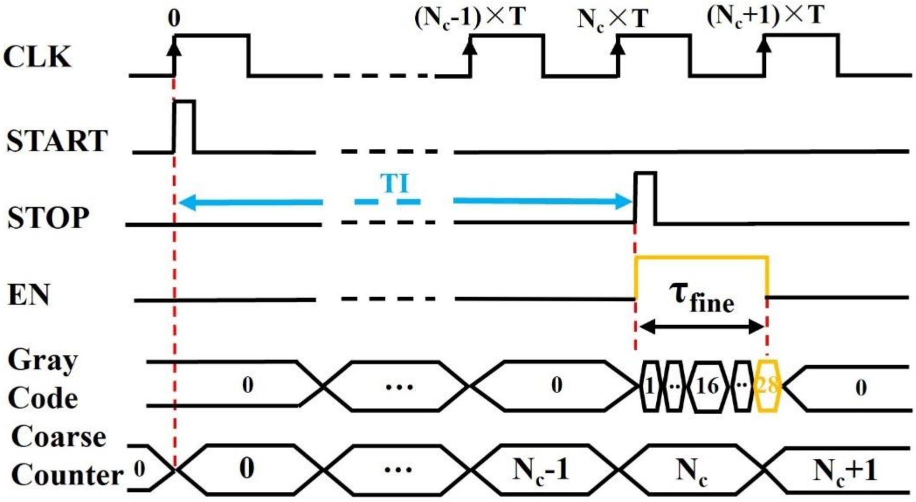


Figure 4.4 Timing diagram of the GCO-TDC.

Figure 4.4 presents the timing diagram of the proposed GCO-TDC. The "CLK" signal in Fig. 4.4 serves as the sampling clock of the proposed TDC, and the "START" and the "STOP" signals are synchronous and asynchronous with it, respectively. Once a rising edge of "STOP" is detected ("STOP" is the input signal for the "Input Shaper" shown in Fig. 4.3), the "EN" signal toggles to "1" and maintains this state until the subsequent rising edge of the "CLK" signal appears. Simultaneously, as shown in Fig. 4.4, the GCO launches and keeps running until its output is sampled by DFFs. With every rising edge of the "STOP" signal, the GCO and the coarse counter output a fine and a coarse code, respectively. Combining the fine and coarse codes, the TI highlighted in blue in Fig. 4.4 is calculated as:

$$TI = (Nc+1)\times T - \tau_{fine}, \tag{4.1}$$

where *Nc is* the coarse code from the coarse counter and τ_{fine} is the TI (highlighted in yellow in Fig. 4.4) corresponding to the fine code.

4.2.2 Sampling matrix

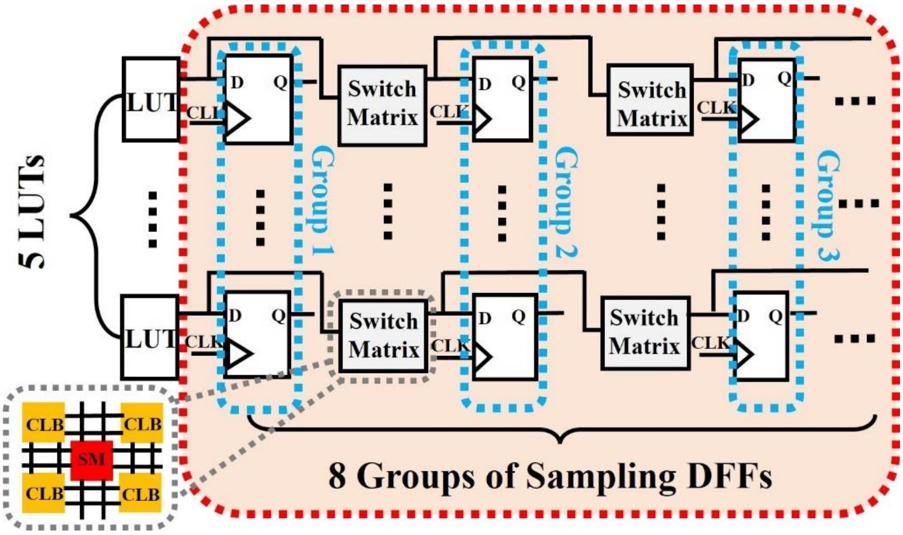


Figure 4.5 Block diagram of the sampling matrix.

Each LUT in the plain GCO-TDC is followed by a single DFF for sampling [109]. Although this architecture is efficient in hardware utilisation, its resolution still requires improvement. Therefore, this chapter proposes a sampling matrix shown in Fig. 4.5 to enhance the GCO-TDC's resolution.

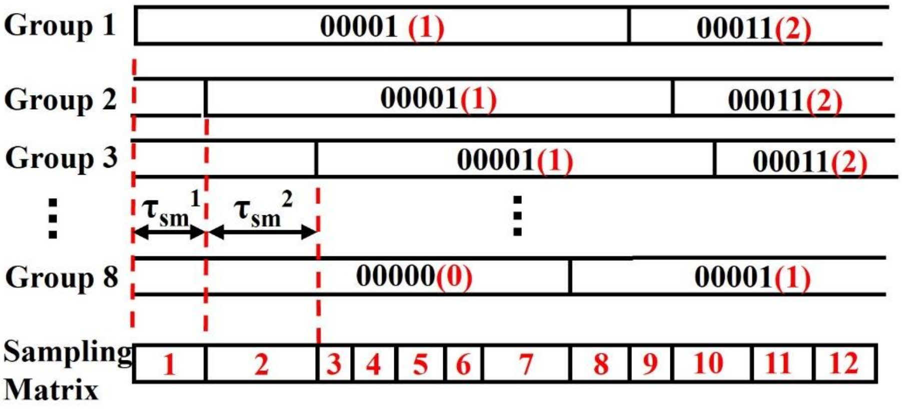


Figure 4.6 Concept of the sampling matrix.

In Fig.4.5, each LUT is sampled by eight DFFs, equivalent to measuring the same TI eight times in one sampling clock period. The concept of this method is shown in Fig. 4.6, where τ_{sm} represents the total propagation delay of the switch matrix and the routing resource between adjacent switch matrices. Moreover, the Vivado Tcl commands “set_property BEL” and “set_property BEL” are used to place LUTs and DFFs manually, and Tcl commands “set_property LOCK_PINS” and “set_property FIXED_ROUTE” are used to lock LUTs’ input pins and fix routing resources, respectively. The resolution of the plain GCO-TDC is defined as Eq. (2.1). However, for the GCO-TDC with a sampling matrix, plain time bins are further subdivided by τ_{sm}, as shown in Fig. 4.6. Therefore, with a sampling matrix, the resolution can be calculated as :

$$LSB_{sm} = \frac{T}{N} \times \frac{1}{M} = \frac{LSB}{M}, \tag{4.2}$$

where M is the number of DFF groups.

With different sampling factors (M), the resolutions and hardware utilisation of the GCO-TDC (without the C&C core) in all three FPGAs are tested and summarised in Table 4-1. According to the utilisation percentage, LUTs are dominant in hardware utilisation. Hence, the utilisation of LUTs is used as the metric to evaluate the hardware utilisation in Eq. (4.3). Besides, the normalized hardware utilisation efficiency of resolution improvement is also proposed as a figure of merit to find a balance between the resolution and hardware utilisation with an increasing M. It is defined as:

$$E_M = \frac{LSB_{M[i-1]} - LSB_{M[i]}}{LSB_{plain}} \ / \ \frac{LUT_{M[i]} - LUT_{M[i-1]}}{LUT_{plain}}, M = 2^i \tag{4.3}$$

Table 4-1 Comparisons of performance and hardware utilisation with different sampling factors (*M*)

M	***E***	***LSB (ps)***	***Used LUTs (%)*[1]**	***Used DFFS (%)*[2]**
		UltraScale+		
1	N/A[3]	158.03	117 (0.05%)	177 (0.04%)
2	2.56	79.01	139 (0.06%)	194 (0.04%)
4	0.96	40.23	170 (0.07%)	221 (0.05%)
8	**0.31**	**19.41**	**222 (0.10%)**	**268 (0.06%)**
16	0.08	10.51	295 (0.13%)	354 (0.08%)
		UltraScale		
1	N/A[3]	267.09	146 (0.06%)	233 (0.05%)
2	6.50	136.39	157 (0.06%)	238 (0.05%)
4	1.76	68.93	178 (0.07%)	248 (0.05%)
8	**0.44**	**35.42**	**220 (0.09%)**	**268 (0.06%)**
16	0.15	19.02	279 (0.12%)	380 (0.08%)
		Virtex-7		
1	N/A[3]	256.41	146 (0.03%)	233 (0.03%)
2	5.73	125.69	159 (0.04%)	238 (0.03%)
4	11.69	64.10	162 (0.04%)	248 (0.03%)
8	**0.43**	**32.54**	**204 (0.05%)**	**268 (0.03%)**
16	0.11	17.05	281 (0.06%)	380 (0.04%)

[1] Percentage of LUTs' utilisation; [2] Percentage of DFFs' utilisation; [3] Not available.

where LSB_{plain} and LUT_{plain} are, respectively, the resolution and LUTs' utilisation of the plain GCO-TDC (M = 1), and LSB_M and LUT_M represent the TDC's resolution and utilisation of LUTs with an M-order sampling matrix. In UltraScale and UltraScale+ FPGAs, E_M decreases when M increases. However, in the Virtex-7 FPGA, E_M has a different trend, reaching the peak value when $M = 4$, due to the different architecture of the implemented device. Overall, in all three FPGAs, E_M approximates 0 when $M = 16$, indicating the low hardware utilisation efficiency of resolution improvement with M increasing from 8 to 16. Therefore, $M = 8$ is selected to balance the performance and hardware utilisation in the proposed TDC.

4.2.3 Virtual bin calibration method

In Chapter 3, an ARM-based architecture was proposed for automatic calibration. However, this architecture offers a fixed resolution and is device-dependent. Hence, the VBCM is proposed in this chapter for automatic calibration and online resolution configuration in common FPGA devices.

Fig. 4.7 shows the workflow of the proposed VBCM. The C&C core calculates compensation factors ($Addr_l$, $Addr_m$, and $Addr_r$, details please see Fig, 4.13) and bin-width calibration factors [Coe_l, Coe_m, and Coe_r, details please see Eq. (4.7)] once the system is launched and the resolution is configured. These factors are then loaded into the C&C BRAM in the compensation and calibration stage (Comp.&cali. in Fig. 4.7). In the following measurement stage (Meas. in Fig. 4.7), indexed by a fine code, these factors are transferred from the C&C BRAM to the histogram BRAM. During this process, the bin-width calibration factors are seriatim added to the corresponding bins of the histogram BRAM, using the compensation factors as the remapping addresses, as shown in Fig. 4.8. Through this process, real-time calibration and resolution configuration are achieved.

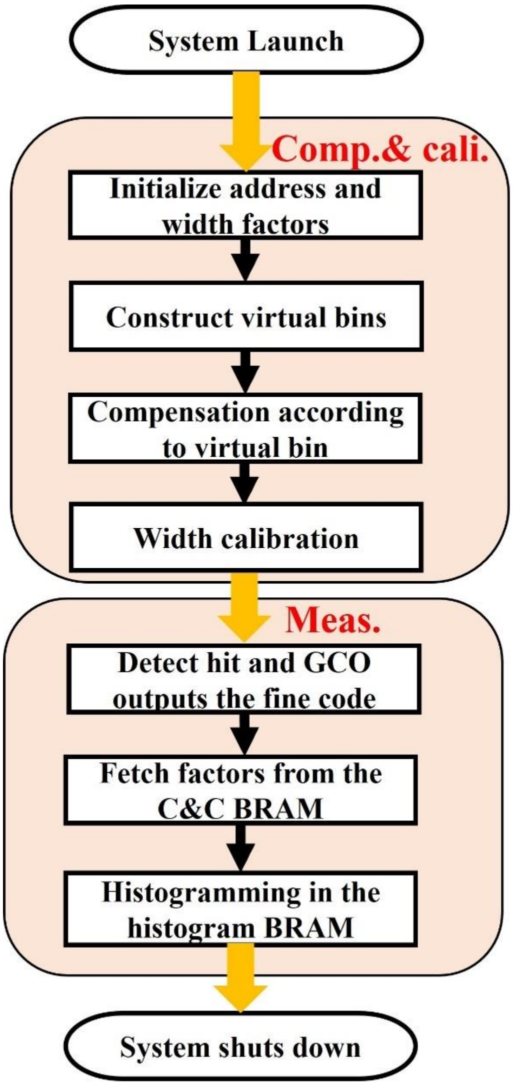

System Launch
Comp.& cali.
Initialize address and width factors
Construct virtual bins
Compensation according to virtual bin
Width calibration
Meas.
Detect hit and GCO outputs the fine code
Fetch factors from the C&C BRAM
Histogramming in the histogram BRAM
System shuts down

Figure 4.7 Workflow of the proposed VBCM.

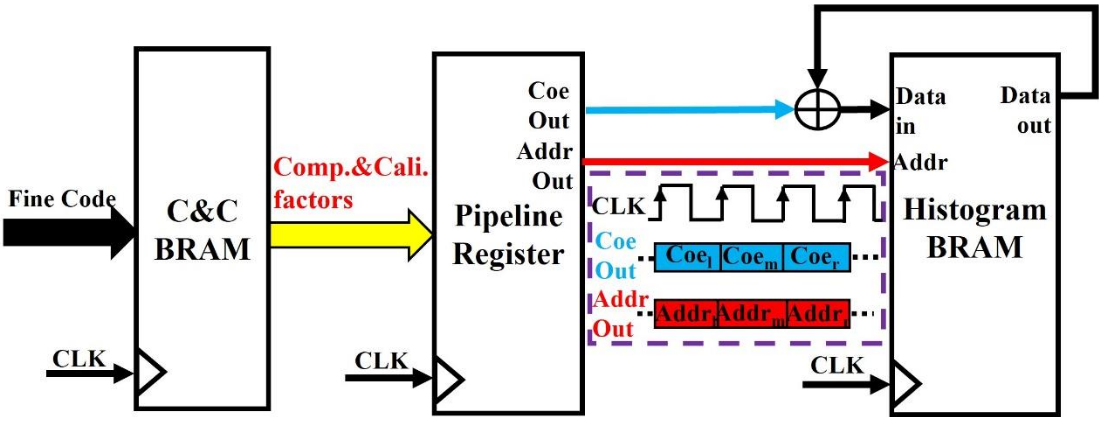


Figure 4.8 Hardware implementation of the real-time histogram with the VBCM.

The process of determining compensation and bin-width calibration factors includes two primary steps: 1) constructing virtual bins and 2) calculating compensation and calibration factors based on virtual bins. Both steps require SCDTs. Fig. 4.9 shows the workflow of constructing virtual bins. When the target resolution is R_{conf}, the number of hits collected at each virtual bin should be:

$$hit_{vir} = \frac{R_{conf}}{T} \times \tilde{N} = \frac{\tilde{N}}{n_{vir}}, \tag{4.4}$$

where n_{vir} [$n_{vir} \leq N \times M$ in Eq. (4.2)] represents the number of virtual bins in one sampling period *T*, and Ñ is the number of random hits for SCDTs. Therefore, the "timestamp" of the *m*-th virtual bin (number of hits collected until the *m*-th virtual bin) and the "timestamp" of the *k*-th raw bin can be defined, respectively, as:

$$T_{vir}[m] = hit_{vir} \times m, \ m \in [1, n_{vir}], \tag{4.5}$$

and

$$T_{raw}[k] = \textstyle\sum_1^k hit_{raw}[j], \ k \in [1, N \times M], \tag{4.6}$$

where $hit_{raw}[j]$ is the number of random hits collected at *j*-th raw bin.

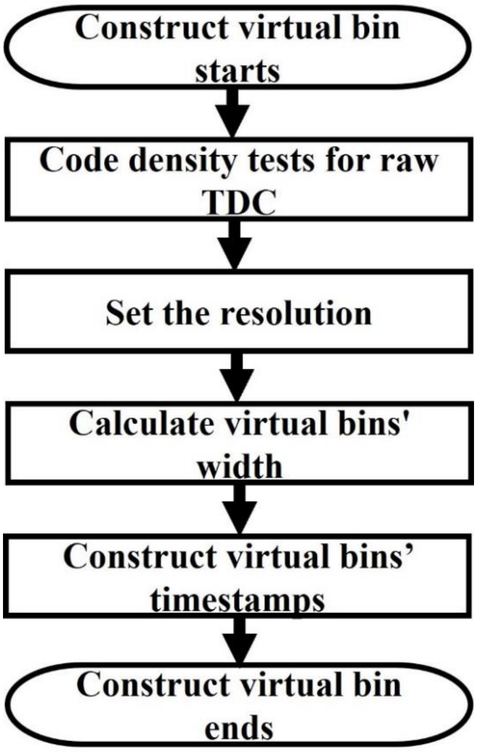

Construct virtual bin starts
Code density tests for raw TDC
Set the resolution
Calculate virtual bins' width
Construct virtual bins' timestamps
Construct virtual bin ends

Figure 4.9 Workflow of the virtual bin construction.

Then, similar to the mixed calibration [83] and weighted histogram calibration (proposed in Chapter 3) methods, the compensation factors for remapping can be calculated using T_{vir} and T_{raw}. However, both the mixed calibration and weighted histogram calibration methods suffer from a limited compensation range, leading to "missing bins" (highlighted in blue in Fig. 4.10 and Fig. 4.11). To address this limitation, this chapter proposes a novel missing-bin-free compensation strategy. The concept of this strategy, which uses three compensation factors ($Addr_l$, $Addr_m$, and $Addr_r$), is shown in Fig. 4.12. Mostly, ultra-narrow bins (<1 LSB, highlighted in green in Figs. 4.10, 4.11, and 4.12) and regular bins (1-2 LSB, highlighted in purple in Figs. 4.10, 4.11, and 4.12) are adjacent to ultra-wide bins (>2 LSB, highlighted in yellow in Figs. 4.10, 4.11, and 4.12). However, previous studies, including the mixed calibration method and the weighted histogram calibration method in Chapter 3, did not fully exploit all compensation factors of ultra-narrow bins. For example, only $BCF_{m,n+1}$ is used in Fig. 4.10, and only $Addr_{l,n+1}$ is used in Fig. 4.11. To leverage the "idle" compensation factors, in Fig. 4.12, both $Addr_{l,n+1}$ and $Addr_{m,n+1}$ are used to remap to virtual bins covered by Bin$_n$ (Bin$_{m+3}$ and Bin$_{m+4}$, highlighted in red in Fig. 4.12). Fig. 4.13 shows the pseudocodes of compensation factor calculations (CFCs). After updating compensation factors based on virtual bins, the SCDTs are conducted again for the compensated TDC to calculate width-calibration factors, as shown in Fig. 4.14. Then, with the number of hits collected at each bin, bin-width calibration factors can be calculated as:

$$Coe_{l,m,r}[k] = \frac{\tilde{N}}{n_{vir}} \times \frac{1}{hit_{com}[i]}, i = Addr_{l,m,r}[k], \tag{4.7}$$

where $hit_{com}[i]$ represents the number of hits collected at the i-th bin of the compensated TDC.

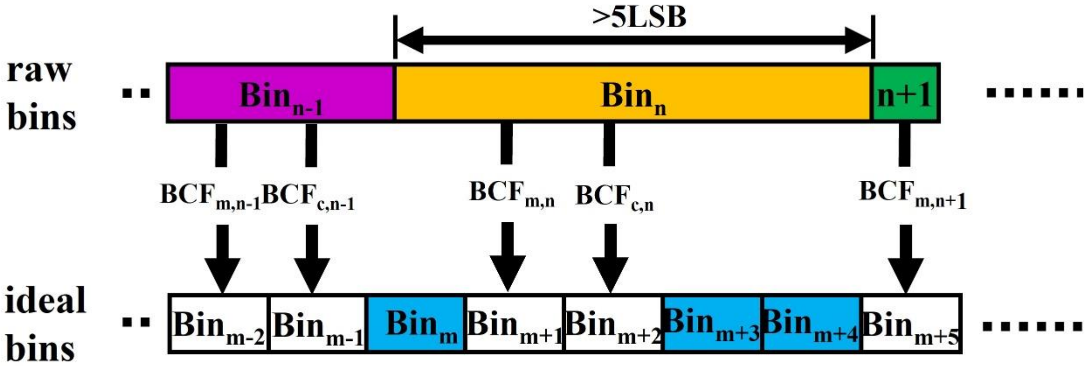


Figure 4.10 Compensation in the mixed calibration method.

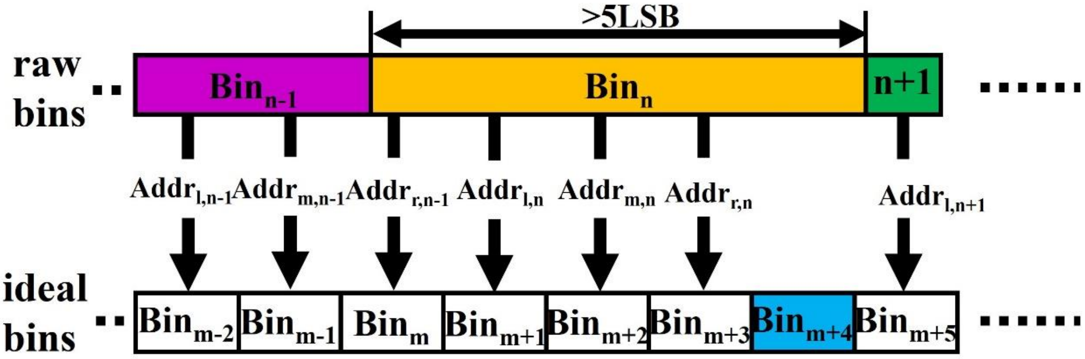


Figure 4.11 Compensation in the weighted calibration method.

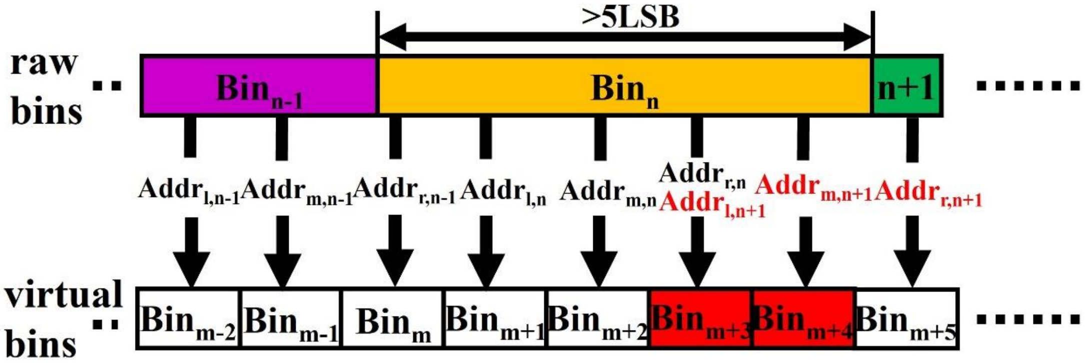
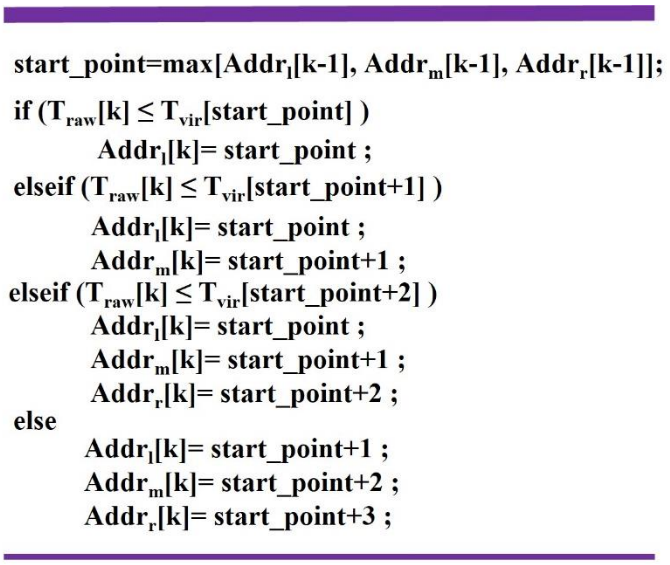


Figure 4.12 Compensation in the VBCM.



```
start_point=max[$Addr_l$[k-1], $Addr_m$[k-1], $Addr_r$[k-1]];
if ($T_{raw}$[k] ≤ $T_{vir}$[start_point] )
        $Addr_l$[k]= start_point ;
elseif ($T_{raw}$[k] ≤ $T_{vir}$[start_point+1] )
        $Addr_l$[k]= start_point ;
        $Addr_m$[k]= start_point+1 ;
elseif ($T_{raw}$[k] ≤ $T_{vir}$[start_point+2] )
        $Addr_l$[k]= start_point ;
        $Addr_m$[k]= start_point+1 ;
        $Addr_r$[k]= start_point+2 ;
else
        $Addr_l$[k]= start_point+1 ;
        $Addr_m$[k]= start_point+2 ;
        $Addr_r$[k]= start_point+3 ;
```

Figure 4.13 Pseudocodes of compensation in the VBCM.

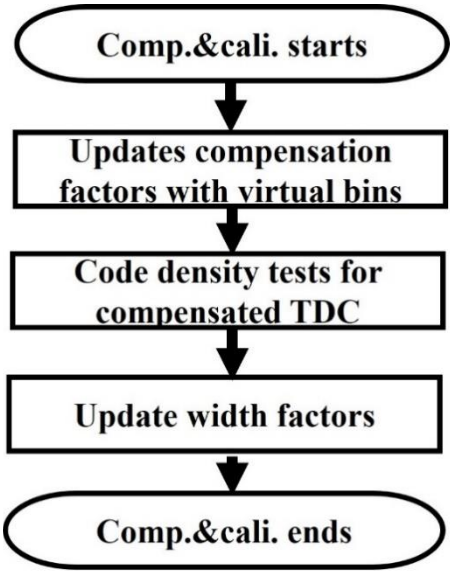

Comp.&cali. starts
Updates compensation factors with virtual bins
Code density tests for compensated TDC
Update width factors
Comp.&cali. ends

Figure 4.14 Workflow of the compensation and bin-width calibration.

4.2.4 Hardware implementation of the VBCM

The VBCM's hardware implementation mainly includes 1) the implementation of the real-time histogram and 2) the implementation of the C&C core. Fig. 4.8 shows the implementation of the real-time histogram, which employs a pipeline structure to minimise the number of the required histogram BRAMs. In the C&C BRAM, three pairs of factors, including compensation and bin-width calibration factors with 8-bit $Addr_l$ ($Addr_m = Addr_l + 1$ and $Addr_r = Addr_l + 2$, so they are not stored here) and 12-bit Coe_l, Coe_m and Coe_r, are merged into a calibration block and stored in a single address to reduce hardware utilisation. Then, through pipeline registers, the merged factors are split and transferred to the histogram BRAM in order within three system clock periods, as shown in Fig. 4.8. Due to only one histogram BRAM required, this design is more hardware-efficient than the design in Ref. [83] (which consumes two histogram BRAMs) and the design in Chapter 3 (which consumes three histogram BRAMs).

The C&C core is designed to calculate compensation and bin-width calibration factors, containing two modules: 1) a CFC module and 2) a bin-width calibration factor calculation (BCFC) module. The BCFC module works subsequent to the CFC module. Hence, some components are multiplexed in these two modules to achieve high hardware utilisation efficiency. Fig. 4.15 shows the hardware implementation of the CFC module. After code density tests, T_{vir} and T_{raw} are calculated according to Eqs. (4.5) and (4.6) and then stored in BRAM-2 and BRAM-1, respectively. Once all T_{vir} and T_{raw} are calculated, they are fetched from their respective BRAMs and processed following the pseudocodes shown in Fig. 4.13 to calculate compensation factors ($Addr_l$, $Addr_m$ and $Addr_r$). After the compensation stage is complete, compensation factors are updated into the C&C BRAM. Then, SCDTs are performed again for the compensated TDC, and results are stored in BRAM-1, as shown in Fig.4.16. Using the compensated TDC's bin widths, the bin-width calibration factors (Coe_l, Coe_m and Coe_r) are calculated according to Eq. (4.7). The BRAM-1, accumulator and the divider (highlighted in blue, yellow and red in Figs. 4.15 and 4.16) in the BCFC module are previously used in the

CFC module. Hence, they can be multiplexed. Besides, three bin-width calibration factors are calculated similarly. Hence, the multiplier-divider component (highlighted in gray in Fig. 4.16) is also multiplexed in the BCFC module, further enhancing the hardware utilisation efficiency.

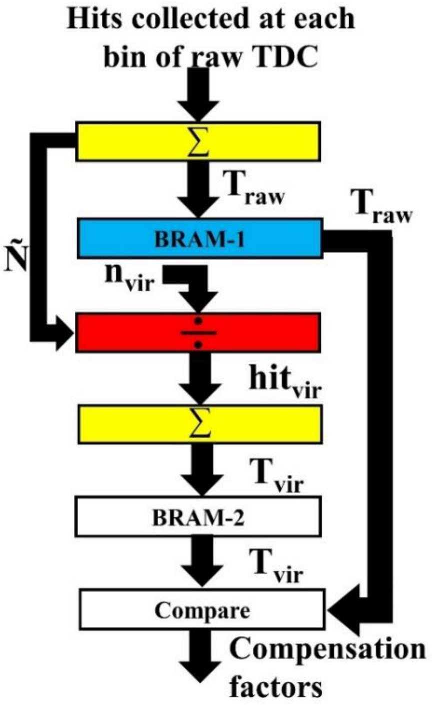
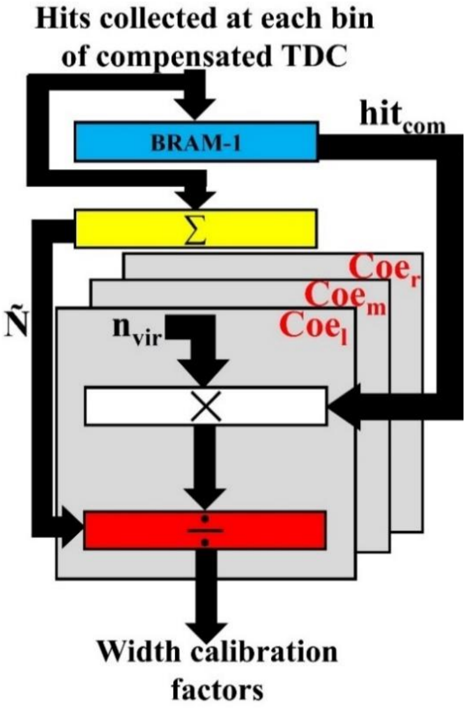


Figure 4.15 Hardware implementation of compensation factor calculations.



Hits collected at each bin of compensated TDC
hit_{com}
BRAM-1
Σ
Coe_{r}
Coe_{m}
Coe_{l}
Ñ
n_{vir}
×
÷
Width calibration factors

Figure 4.16 Hardware implementation of bin-width calibration factor calculations.

To minimise hardware utilisation, all data are presented and calculated using the fixed-point format in this design. Considering errors when discarding decimal parts, the last $\bar{M}$ bits of Coe are used to present decimals. Bin-width calibration factors are then expressed as:

$$\overline{Coe_{l,m,r}[k]} = Coe_{l,m,r}[k] \times 2^{\bar{M}} = \frac{\tilde{N}}{n_{vir}} \times \frac{2^{\bar{M}}}{hit_{com}[i]}, i = Addr_{l,m,r}[k]. \tag{4.8}$$

For selecting an appropriate $\bar{M}$, MATLAB simulations of the VBCM method were performed with an increasing $\bar{M}$ (setting $n_{vir} = N$). Table 4-2 summarises the simulation results, indicating that increasing $\bar{M}$ from 5 to 6 cannot further enhance the linearity of the calibrated GCO-TDC in Kintex-UltraScale and Virtex-7 FPGAs. However, only the INL$_{\text{pk-pk}}$ in the Kintex-UltraScale+ FPGA is improved. Therefore, $\bar{M} = 5$ is chosen for all three FPGAs to optimise the design.

Table 4-2 Results of the VBCM with a different $\bar{M}$

16nm UltraScale+						
$\bar{M}$	1	2	3	4	5	6
DNL$_{\text{pk-pk}}$ (LSB)	0.68	0.39	0.20	0.12	0.10	0.08
INL$_{\text{pk-pk}}$ (LSB)	3.21	1.39	0.50	0.42	0.49	0.31
20nm UltraScale						
$\bar{M}$	1	2	3	4	5	6
DNL$_{\text{pk-pk}}$ (LSB)	0.68	0.38	0.20	0.16	0.12	0.12
INL$_{\text{pk-pk}}$ (LSB)	2.56	0.95	0.77	0.48	0.36	0.35
28nm Virtex-7						
$\bar{M}$	1	2	3	4	5	6
DNL$_{\text{pk-pk}}$ (LSB)	0.65	0.36	0.20	0.16	0.09	0.10
INL$_{\text{pk-pk}}$ (LSB)	3.11	1.35	0.73	0.62	0.26	0.34

4.3 Experimental results

The proposed TDC was implemented and tested in the following evaluation boards: 1) ZCU-104 [158], powered by a 16 nm Kintex-UltraScale+ FPGA, as shown in Fig. 4.17; 2) KCU-105 [159], powered by a 20 nm Kintex-UltraScale FPGA, as shown in Fig. 4.18; and 3) NetFPGA-SUME [160], powered by a 28 nm Virtex-7 FPGA, as shown in

Fig. 4.19. Random hits for SCDTs were generated using a Stanford Research System (SRS) CG-635 [161], while TDC's sampling clocks were from low-jitter crystal oscillators on the boards, specifically the IDT-8T49 on ZCU-104, the SI-570 on KCU105, and the DSC-1103 on NetFPGA-SUME. The period of the GCO-TDC's sampling clock should be less than the period of the GCO, and GCOs have different frequencies in three FPGAs. Therefore, the frequencies of TDCs' sampling clocks were 226MHz (ZCU-104), 156MHz (KCU-105) and 156MHz (NetFPGA-SUME), respectively. Besides, both voltage and temperature were maintained in the experiments.

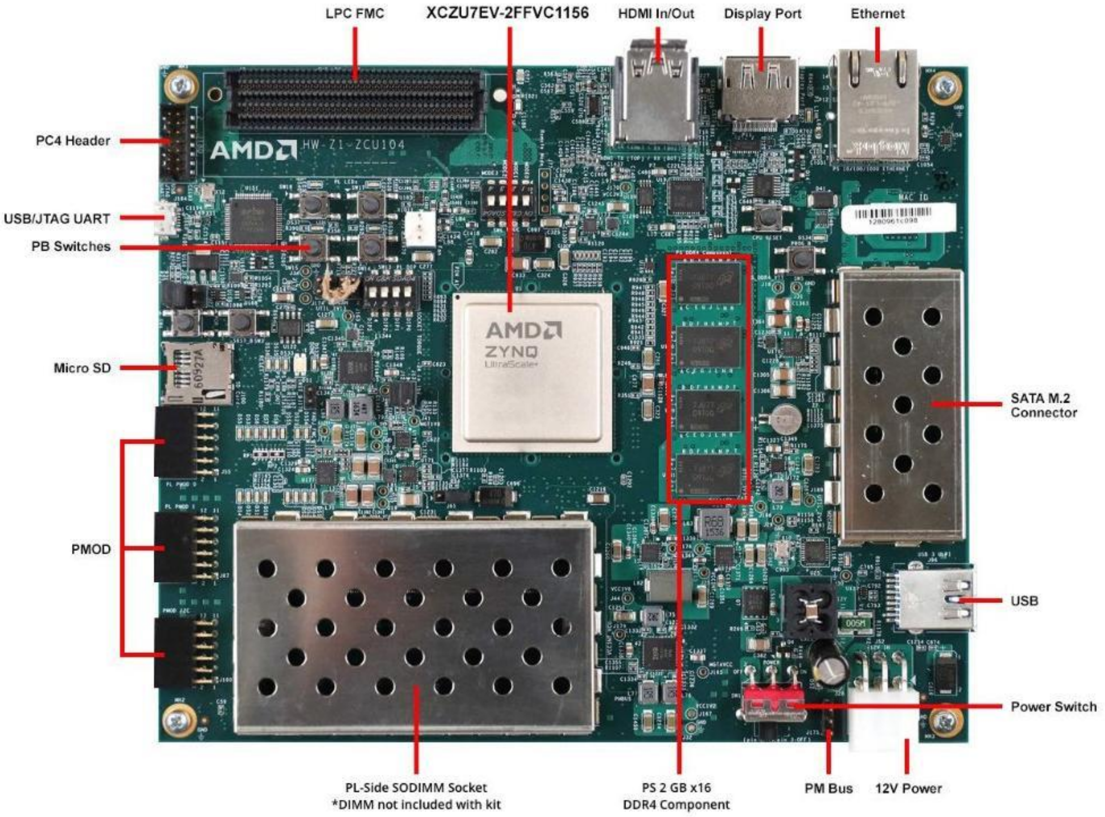


Figure 4.17 ZCU-104 evaluation board [158].

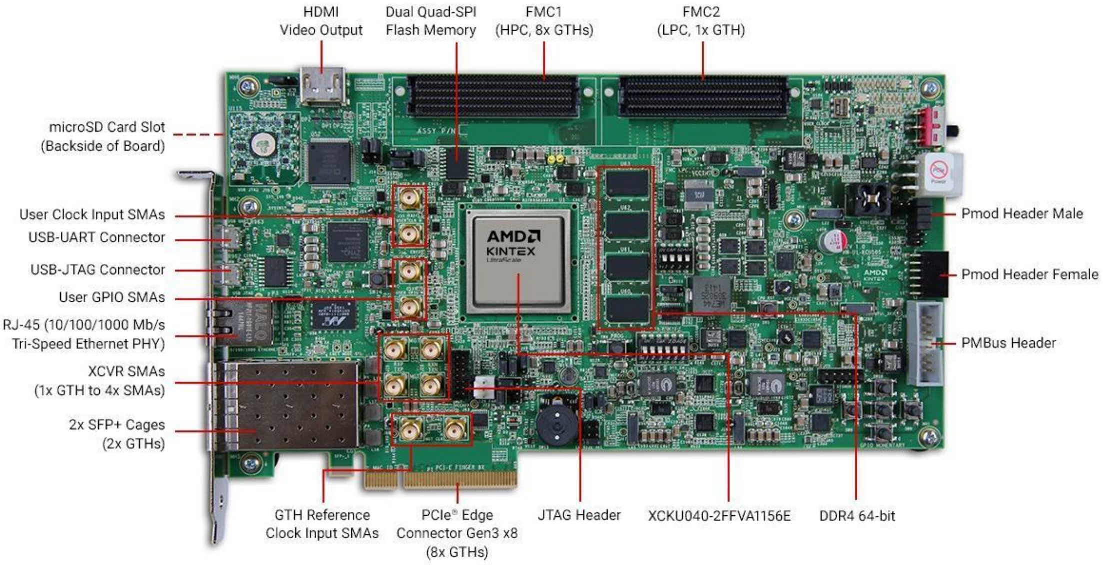


Figure 4.18 KCU-105 evaluation board [159].

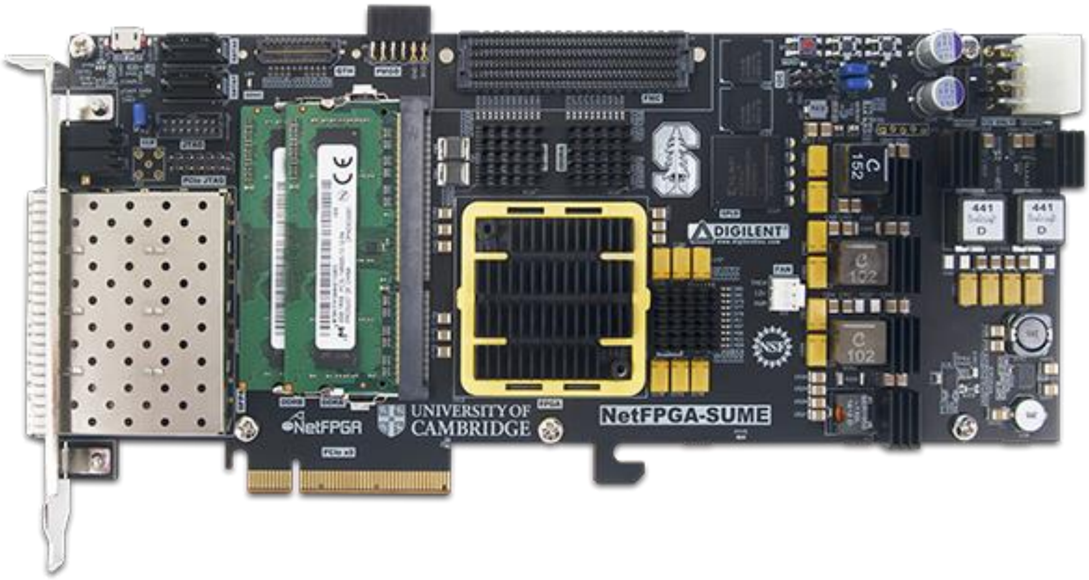


Figure 4.19 NetFPGA-SUME evaluation board [160].

4.3.1 Resolution configuration and linearity

With different resolution configurations, the TDC's linearity was characterised by DNL, INL and their standard deviations (σ_{DNL} and σ_{INL}). Moreover, the equivalent bin-width (ω_{eq}) and its deviation (σ_{eq}) were also used to evaluate the overall impact of TDC's linearity on measurements. Table 4-3 summarises the experimental results, showing that the proposed TDC's linearity is significantly improved with the VBCM. For the TDC in the 16 nm Kintex-UltraScale+ FPGA, $\text{DNL}_{\text{pk-pk}}$ is improved by more than 44-fold (from 3.98 LSB to less than 0.09 LSB), and $\mathrm{INL}_{\mathrm{pk\text{-}pk}}$ is improved by more than 35-fold (from 7.17 LSB to less than 0.20 LSB). Similarly, TDCs in 20 nm Kintex-UltraScale and 28 nm Virtex-7 FPGAs also have enhanced linearity, with $\mathrm{DNL}_{\mathrm{pk\text{-}pk}}$, respectively, improved by more than 59-fold (from 4.76 LSB to less than 0.08 LSB) and 78-fold (from 5.48 LSB to less than 0.07 LSB), and $\mathrm{INL}_{\mathrm{pk\text{-}pk}}$, respectively, improved by more than 88-fold (from 12.38 LSB to less than 0.14 LSB) and 38-fold (from 11.23 LSB to less than 0.29 LSB). The results indicate the proposed TDC achieves high linearity with different resolutions.

Table 4-3 Linearity of the proposed TDC with different resolutions.

		LSB (ps)	***$DNL_{pk\text{-}pk}$ (LSB)***	***σ_{DNL} (LSB)***	***$INL_{pk\text{-}pk}$ (LSB)***	***σ_{INL} (LSB)***	***ω_{eq} (ps)***	***σ_{eq} (LSB)***
Kintex-UltraScale+	***Raw-TDC***	19.41 (n = 228)	3.98	0.96	7.17	1.39	43.11	0.64
	VBCM-TDC	20.97 (n_{vir} = 211)	0.09	0.01	0.20	0.04	20.98	0.29
		29.90 (n_{vir} = 148)	0.05	0.01	0.12	0.02	29.91	0.29
		39.86 (n_{vir} = 111)	0.05	0.01	0.12	0.03	39.86	0.29
		50.28 (n_{vir} = 88)	0.05	0.01	0.11	0.02	50.29	0.29
		80.45 (n_{vir} = 55)	0.03	0.01	0.09	0.02	80.46	0.29
Kintex-UltraScale	***Raw-TDC***	35.42 (n = 181)	4.76	0.93	12.38	2.92	76.94	4.72
	VBCM-TDC	36.01 (n_{vir} = 178)	0.08	0.01	0.14	0.03	36.02	0.29
		40.06 (n_{vir} = 160)	0.07	0.01	0.12	0.03	40.07	0.29
		50.08 (n_{vir} = 128)	0.05	0.01	0.13	0.03	50.09	1.00
		80.13 (n_{vir} = 80)	0.05	0.01	0.08	0.02	80.15	0.29
		100.16 (n_{vir} = 64)	0.04	0.01	0.11	0.03	100.18	0.29
Virtex-7	***Raw-TDC***	32.54 (n = 197)	5.48	0.95	11.23	2.35	72.96	0.65
	VBCM-TDC	34.84 (n_{vir} = 184)	0.07	0.02	0.29	0.08	34.85	0.29
		40.06 (n_{vir} = 160)	0.07	0.01	0.18	0.04	40.07	0.29
		50.08 (n_{vir} = 128)	0.07	0.01	0.16	0.04	50.09	0.29
		80.13 (n_{vir} = 80)	0.04	0.01	0.15	0.03	80.15	0.29
		100.16 (n_{vir} = 64)	0.05	0.01	0.11	0.03	100.18	0.29

4.3.2 Time interval tests

The precision of the proposed TDC is calculated as the standard deviation (σ) of repetitive measurements for a constant TI. Moreover, FPGA-inside macros (IDELAY3 in Kintex-UltraScale+ and Kintex-UltraScale FPGAs [18], and IDELAY2 in the Virtex-7 FPGA [153]) were used to generate controllable delays to avoid jitters introduced by external inputs. The test setups were similar to that in Fig. 3.14, with the clock frequencies adjusted accordingly. And the constant TI was measured by a single TDC channel, as shown in Fig. 3.14.

Figs. 4.20 - 4.22 show the precisions for different resolutions in Kintex-UltraScale+, Kintex-UltraScale, and Virtex-7 FPGAs. TI tests were conducted at TIs shorter than one sampling clock period. However, for each resolution option with different TIs, neither the average nor maximum precision can represent precision because they overestimate or underestimate it [128]. Therefore, the valid precision (σ_{valid}) is used, and it is defined as [128]:

$$\sigma_{valid}^2 = \sum_1^H \frac{\sigma_i^2}{H}, \tag{4.9}$$

where σ_i is the measurement standard deviation for the i-th constant TI, and H is the total number of different TIs. With the improvements in resolutions, the valid precisions have a deteriorating trend in all three FPGAs. The TDC in the Kintex-UltraScale+ FPGA achieves the best valid precision of 0.36 LSB when LSB = 80.45 ps. Similarly, TDCs in the Kintex-UltraScale and Virtex-7 FPGAs, respectively, achieve the best valid precision of 0.46 LSB and 0.49 LSB when LSB = 100.16 ps.

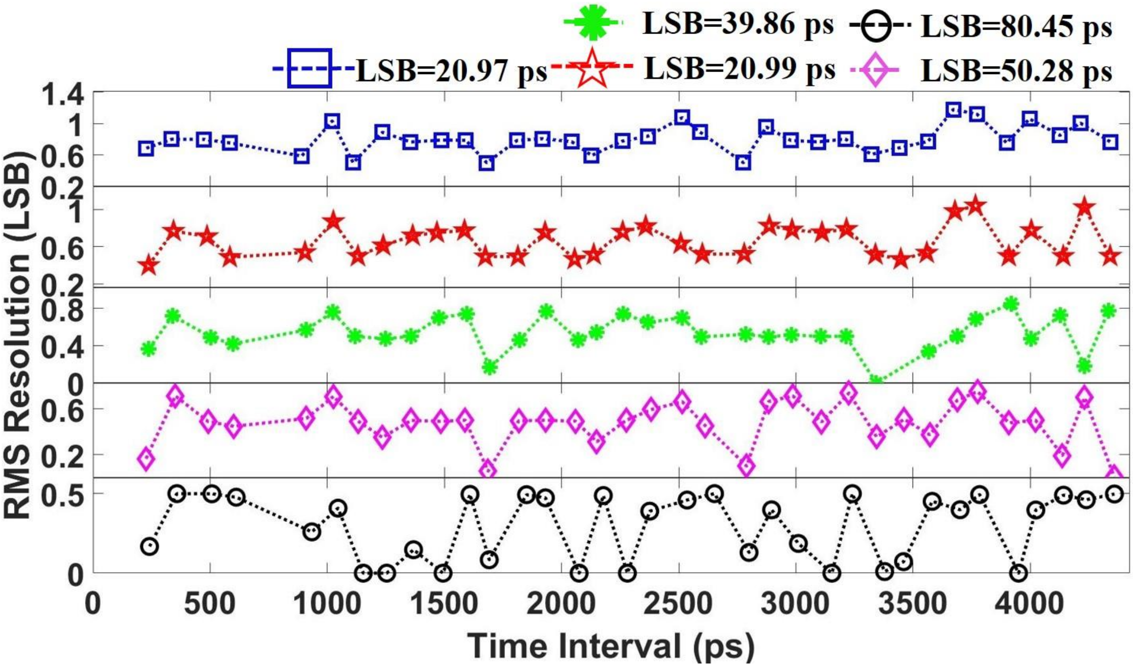


Figure 4.20 RMS precisions with different resolutions in the Kintex-UltraScale+ FPGA.

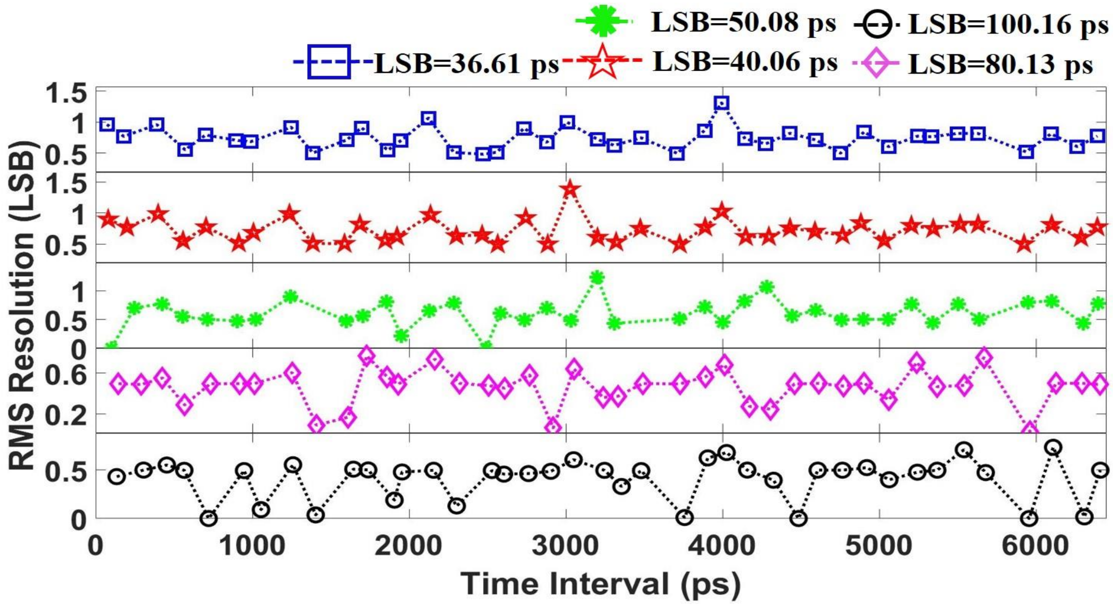


Figure 4.21 RMS precisions with different resolutions in the Kintex-UltraScale FPGA.

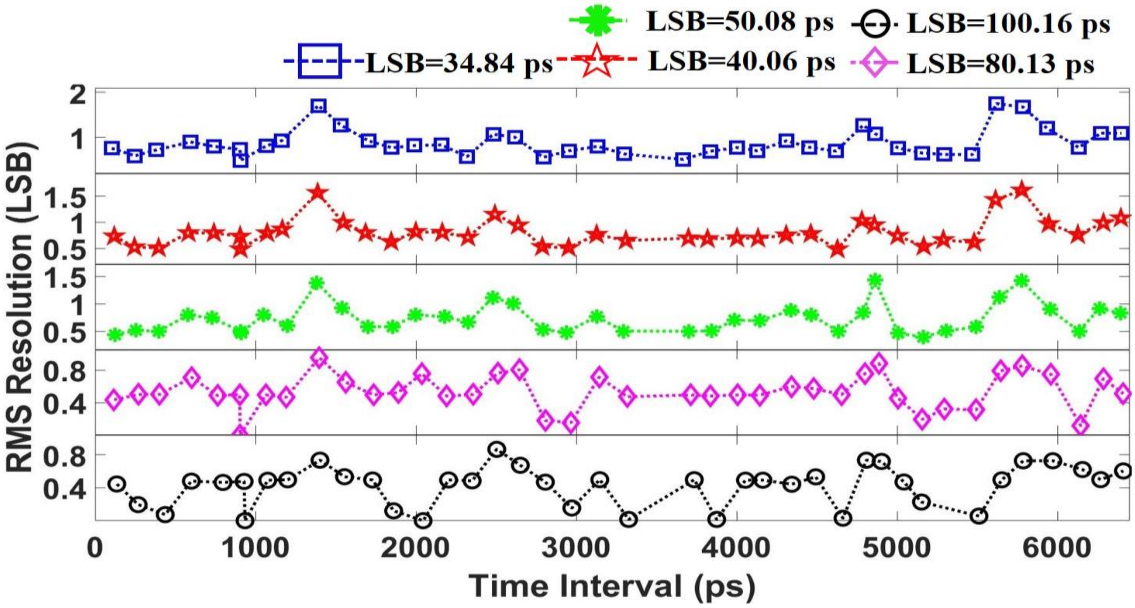


Figure 4.22 RMS precisions with different resolutions in theVirtex-7 FPGA.

4.3.3 Multi-channel design

The proposed 16-channel TDC was implemented in all three FPGAs. Each channel consumes an 18k-BRAM as the histogram BRAM and a 36k-BRAM as the C&C BRAM. Meanwhile, building a GCO-TDC with a sampling matrix also requires less than 230 LUTs and 270 DFFs. Moreover, two 18k-BRAMs, no more than 3800 LUTs and 1600 DFFs are used for the C&C core to calculate compensation and calibration factors. Table 4-4 summarises the hardware utilisation of the proposed TDC, indicating that this design is more hardware-efficient than TDL-TDCs in Ref. [83] and Ref. [128], and has similar hardware utilisation compared with the RO-TDC presented in Ref. [99] (detailed comparisons are shown in Table 4-6).

Table 4-4 Hardware utilisation of the proposed TDC

		LUT	***DFF***	***CARRY[1]***	***BRAM[2]***
UltraScale+ 16 nm	Available	230400	460800	28800	312
	1-ch.	222	268	9	1.5
	16-ch.	3548	4288	144	24
	C&C core	3719	1596	310	1
UltraScale 20 nm	Available	242400	484800	30300	600
	1-ch.	220	268	9	1.5
	16-ch.	3518	4288	144	24
	C&C core	3725	1569	310	1
Virtex-7 28 nm	Available	433200	866400	108300	1470
	1-ch.	204	268	18	1.5
	16-ch.	3263	4288	288	24
	C&C core	3753	1597	574	1

[1] CARRY8s in UltraScale+ and UltraScale FPGAs, and CARRY4s in Virtex-7 FPGA; [2] 36K-BRAM.

Fig. 4.23 shows implementation layouts in the Kintex-UltraScale+ FPGA. As shown in Fig. 4.23b, to minimise jitters from routing resources, each “Input Shaper” is constrained close to its corresponding GCO. Moreover, the sampling DFFs shown in Fig. 4.23b are also constrained to ensure uniformity between channels. For the proposed TDC in the Kintex-UltraScale+ FPGA, DFF groups are placed contiguously [SLICE XnYm and SLICE X(n+1)Ym] in a row. However, DFFs are placed at a fixed distance [SLICE XnYm and SLICE X(n+2)Ym] for TDCs in the Kintex-UltraScale and Virtex-7 FPGAs.

SCDTs were performed for 16-channel TDCs in all three FPGAs. Table 4-5 summarises the linearity with different resolutions, indicating that the proposed TDC has high linearity and good uniformity.

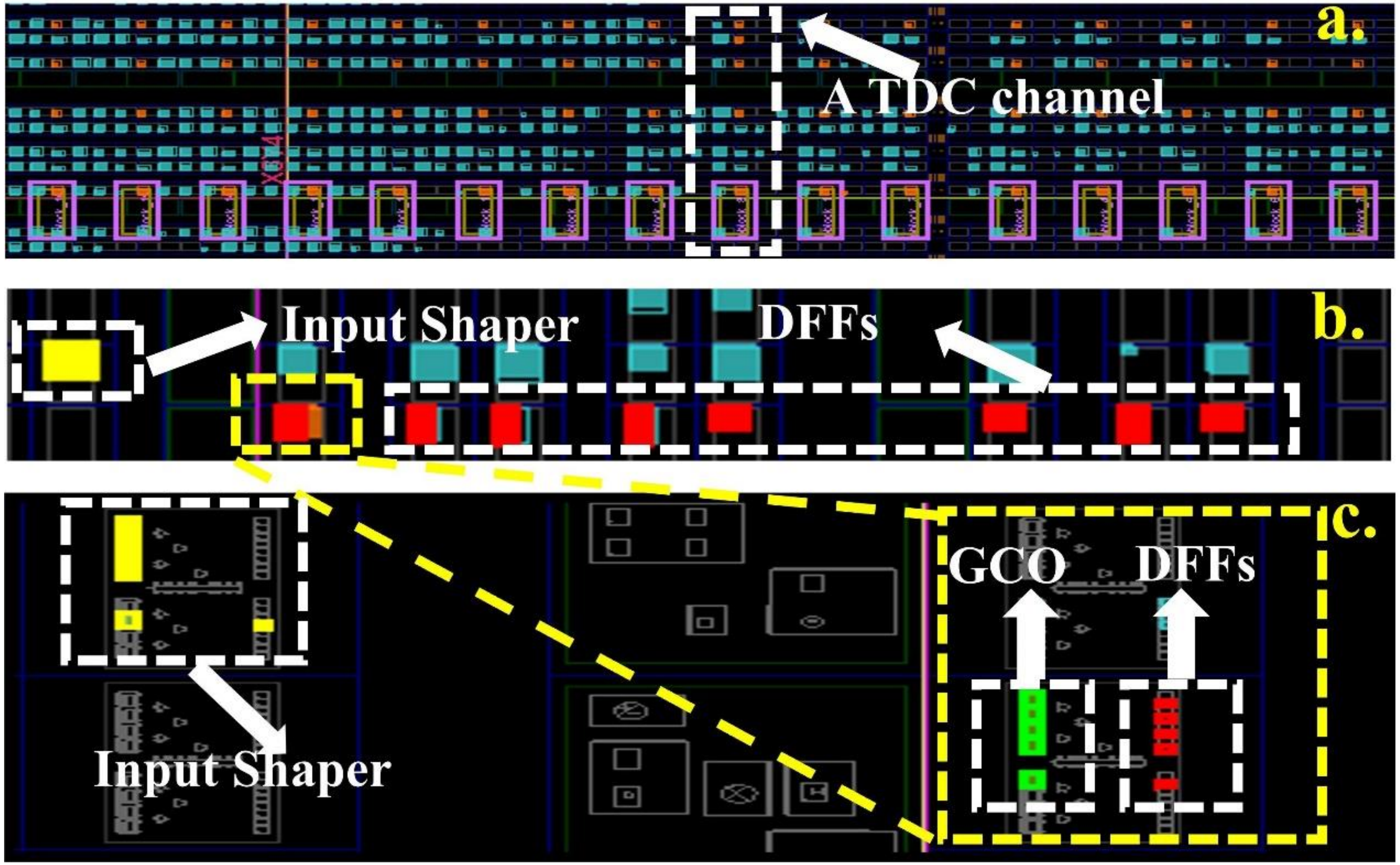


Figure 4.23 Implementation layouts of (a) 16 channels, (b) a single channel, and (c) an input shaper and a GCO.

Table 4-5 DNL$_{pk\text{-}pk}$ of 16-channel TDCs in 16 nm, 20 nm, and 28 nm FPGAs

LSB	***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***Ave.***
（ps）	***LSB***																
							UltraScale+										
20.97	0.09	0.07	0.10	0.08	0.07	0.18	0.08	0.07	0.11	0.06	0.07	0.07	0.07	0.13	0.07	0.08	0.09
29.90	0.05	0.06	0.06	0.06	0.05	0.08	0.06	0.07	0.06	0.06	0.06	0.07	0.06	0.06	0.06	0.05	0.06
39.86	0.05	0.04	0.05	0.04	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.04	0.05	0.05	0.06	0.05
50.28	0.05	0.04	0.04	0.04	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.04	0.05	0.04
80.45	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
							UltraScale										
36.01	0.08	0.16	0.08	0.13	0.08	0.08	0.08	0.09	0.07	0.18	0.09	0.07	0.08	0.08	0.21	0.09	0.10
40.06	0.07	0.11	0.10	0.20	0.10	0.02	0.06	0.12	0.09	0.10	0.07	0.09	0.09	0.10	0.25	0.16	0.11
50.08	0.05	0.15	0.14	0.05	0.17	0.07	0.06	0.05	0.07	0.06	0.07	0.09	0.06	0.09	0.06	0.05	0.08
80.13	0.05	0.06	0.05	0.08	0.05	0.15	0.05	0.05	0.05	0.05	0.05	0.05	0.09	0.06	0.05	0.06	0.06
100.16	0.04	0.18	0.05	0.06	0.04	0.16	0.05	0.04	0.05	0.05	0.05	0.04	0.06	0.04	0.05	0.05	0.06
							Virtex-7										
34.84	0.07	0.07	0.06	0.06	0.10	0.07	0.07	0.06	0.05	0.11	0.06	0.15	0.09	0.07	0.07	0.10	0.08
40.06	0.07	0.11	0.09	0.07	0.09	0.08	0.08	0.08	0.09	0.15	0.08	0.08	0.07	0.09	0.06	0.09	0.09
50.08	0.07	0.07	0.06	0.06	0.05	0.06	0.06	0.07	0.05	0.06	0.06	0.07	0.06	0.06	0.06	0.07	0.06
80.13	0.04	0.04	0.05	0.05	0.05	0.05	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
100.16	0.05	0.04	0.05	0.04	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.05	0.04

Table 4-6 Comparisons of recently published FPGA-TDCs

Ref- year	Methods	Devi. Proc.(nm)	LSB (ps)	ω_{eq} (ps)	RMS Preci. (ps)	DNL(LSB)	INL(LSB)	LUT (%)[1]	DFF (%)[1]	Carry (%)[1]	36K- BRAM
TDL-TDCs											
[83]- 19	Tuned-TDL, Sub-TDL, Mixed Calibration	28	10.54	10.55	14.59[2]	[-0.05, 0.08]	[-0.09, 0.11]	1145 0.26	1916 0.22	N/S[3]	1.5
		20	5.02	5.03	7.80[2]	[-0.12, 0.11]	[-0.18, 0.46]	703 0.29	1195 0.24	80[4] 0.26	
[149] -21	Slide Scale, Gain & Error cal., Moving Ave.	28	4.88	N/S[3]	2.90~ 8.03	[-0.10, 0.15]	[-0.23, 0.28]	2962 N/S[3]	4157 N/S[3]	N/S[3]	N/S[3]
[128] -21	Mixed-binning	20	51.28	51.29	15.89[2]	[-0.018, 0.021]	[-0.019,0.035]	663 0.27	1124 0.23	74[4] 0.24	2.5
			83.33	83.34	21.67[2]	[-0.017, 0.016]	[-0.028,0.003]				
			105.26	105.26	26.32[2]	[-0.008, 0.008]	[-0.009,0.007]				
[157] -21	Half single-chain, Real states-based Coding	16	5	N/S[3]	19[5]	[-0.99, 1.44]	[-2.84, 1.62]	N/S[3]	N/S[3]	N/S[3]	N/S[3]
			21.56	N/S[3]	30.18[5]	[-0.16, 0.19]	[-0.50, 0.33]	N/S[3]	N/S[3]	N/S[3]	N/S[3]
			87.74	N/S[3]	105.29[5]	[-0.07, 0.05]	[0.00, 0.11]	N/S[3]	N/S[3]	N/S[3]	N/S[3]
Other TDCs											
[99]- 20	Bidirectional, RO Vernier	65	24.50	N/S[3]	28.00	[-0.20,0.25]	[0.03,0.82]	172 N/S[3]	986 N/S[3]	N/S[3]	N/S[3]
[156] -21	NUMMP, Timing Scale Marking	28	1.87	N/S[3]	2.79	[-0.54,1.30]	[-2.21,3.51]	1679 0.82	1103 0.27	N/S[3]	12
			11.24	N/S[3]	8.07	[-0.43,0.26]	[-0.55,0.30]	1328 0.65	857 0.2	N/S[3]	12
			20.00	N/S[3]	12.81	[-0.05,0.06]	[-0.15,0.08]	634 0.41	828 0.16	N/S[3]	12

GCO-TDCs											
[109]-2019	GCO, Bin-by-bin Cali.	28	256	N/S[3]	155	[-0.53,0.72][6]	N/S[3]	8 N/S[3]	8 N/S[3]	N/S[3]	N/S[3]
[111]-2020	GCO, Manual Routing	28	380.9	N/S[3]	290	[-0.38,0.38]	[0.01,0.70]	6 N/S[3]	10 N/S[3]	N/S[3]	N/S[3]
[110]-2021	GCO, Double Sampling	16	69	N/S[3]	54.99	[-0.95,0.81]	[-1.01,0.49]	5 N/S[3]	19 N/S[3]	N/S[3]	N/S[3]
This TDC	**GCO, Sampling Matrix, VBCM**	**16**	**20.97**[7]	**20.98**	**17.11**[2,8]	**[-0.055, 0.034]** **0.087**[9]	**[-0.196, 0.000]** **0.224**[9]	**222**[10] **0.10** **455**[11] **0.20**	**268**[10] **0.06** **368**[11] **0.08**	**9**[4] **0.03**	**1.5**[12]
		20	**36.01**[7]	**36.02**	**27.37**[2,8]	**[-0.036, 0.046]** **0.102**[9]	**[-0.057, 0.081]** **0.262**[9]	**220**[10] **0.09** **453**[11] **0.19**	**268**[10] **0.06** **367**[11] **0.08**	**9**[4] **0.03**	**1.5**[12]
		28	**34.84**[7]	**34.85**	**32.33**[2,8]	**[-0.033, 0.034]** **0.078**[9]	**[-0.016, 0.277]** **0.203**[9]	**204**[10] **0.05** **437**[11] **0.10**	**268**[10] **0.03** **368**[11] **0.04**	**18**[13] **0.02**	**1.5**[12]

[1] Percentage of hardware utilisation for the target device; [2] Single-shot precision; [3] N/S=not specified; [4] CARRY8s; [5] FWHM of the residual; [6] Approximate values from figures presented in literature; [7] Proposed TDCs' best resolution in this device; [8] Valid RMS precision; [9] Average peak-to-peak DNL or INL of multi-channel TDCs; [10] Each channel's hardware utilisation without the C&C core; [11] Each channel's average hardware utilisation with the C&C core; [12] Each channel's BRAM utilisation; [13] CARRY4s.

4.4 Comparison

Table 4-6 summarises recently published FPGA-TDCs and the TDC proposed in this chapter. As Table 4-6 shows, the TDL is the mainstream of FPGA-TDCs. Moreover, other architectures like the RO-based Vernier [99] and the nonuniform monotonic multiphase (NUMMP) [156] architectures are also well-developed. However, this chapter has further developed the new GCO-TDC architecture first proposed by Wu and Xu [109].

Unlike high-resolution (<10 ps) TDCs designed for scientific applications, the TDC proposed in this chapter aims for multi-channel LiDAR applications, offering variable resolutions and high linearity. The proposed TDC is more hardware-efficient than TDL-based and NUMMP-based TDCs with similar metrics (for example, a TDC with 20 ps resolution or high linearity). The design proposed in this chapter consumes only one-third of the LUTs and one-fourth of the DFFs compared with the TDC in Ref. [128], and one-third of the LUTs and one-third of the DFFs compared with the TDC in Ref. [156]. Although the RO-TDC in Ref. [99] has similar hardware utilisation, it suffers from a long dead time, peaking at 602 ns. Moreover, the proposed GCO-TDC consumes slightly more hardware resources than previously published GCO-TDCs [109], [110], [111]. However, it is acceptable because the TDC proposed in this chapter has significant improvements in resolution and linearity. Simultaneously, it offers configurable resolutions.

Various calibration methods, including bin-by-bin calibration [68], bin-width calibration [120], and mixed calibration [83] methods, can enhance TDC's linearity and precision. However, they all require manual calibration. For automatic calibration without manual intervention, Choi and Jee proposed the gain and error calibration method [149], and I also proposed the weighted histogram calibration method in Chapter 3. However, these two methods only offer fixed resolutions. To my knowledge, the proposed VBCM is the first to achieve online resolution configuration and automatic calibration simultaneously in FPGA-TDCs. In the current design, the C&C

core serves 16 channels, but it can serve more if required. This scalability allows for an easy extension in the number of channels and reduces each channel's average hardware utilisation (with the C&C core) when more channels are implemented.

Verilog is utilised for the FPGA implementation of the proposed TDC. Moreover, GCO-TDC's linearity and resolution are sensitive to placing and routing strategies. Hence, some constraints are required to ensure that GCO-TDC's placements and routes are immobile. For instance, Vivado Tcl commands "set_property BEL" and "set_property LOC" were first used to place LUTs and DFFs manually [162]. Then, Tcl commands "set_property LOCK_PINS" and "set_property FIXED_ROUTE" were used to lock LUTs' input pins and fix routing resources, respectively [162]. The "Input Shaper", GCO and sampling matrix can be verified using post-implementation simulations, while the C&C core can be verified using behaviour simulations. Although the implementation of the proposed TDC is slightly more complex than conventional TDL-TDCs, it is acceptable since the proposed TDC with automatic calibration has low hardware utilisation, high linearity and configurable resolutions.

4.5 Conclusion

To enhance GCO-TDC's resolution, a novel sampling architecture, the sampling matrix, was proposed in this chapter. With this new architecture, GCO-TDCs achieve excellent performance and low hardware utilisation. This chapter also proposed the VBCM, which allows the GCO-TDC to achieve automatic calibration and online resolution configuration. Moreover, the hardware implementation of the VBCM is presented in this chapter, achieving high hardware utilisation efficiency through multiplexing critical components.

In this chapter, I implemented the proposed 16-channel TDC in Kintex-UltraScale+, Kintex-UltraScale and Virtex-7 FPGAs to evaluate the design. Experimental results indicate that the proposed TDCs have competitive linearity and excellent uniformity. Due to online resolution configuration, they can have broad applications in dToF-LiDAR and FLIM. Moreover, the features of automatic calibration and low hardware utilisation make this design suitable for use as a TDC-core in both rapid prototype verification and commercial products.

Chapter 5 Ultra-high resolution 4-edge WU TDCs with a bidirectional encoder

5.1 Motivation

In high-energy physics, high-resolution TDCs are key components in front-end data acquisition and readout systems. They are used to measure the ToF and ToA to determine the interactions and decay of particles [163]. For example, Ablikim *et al.* proposed a TDC with less than 20 ps resolution for the BESIII [164], Xue *et al.* proposed a 128-channel TDC with a resolution from 45.3 ps to 57.7 ps for resistive-plate chambers [165], and Zhang *et al.* proposed a 17.8 ps TDC for the high-luminosity LHC upgrade [58]. Moreover, high resolutions also benefit bioimaging. For example, tumors (typically < 5 mm in size) are detectable if a detector can achieve a resolution of picoseconds [166], [167]. Hence, researchers have proposed several designs for high resolutions. For example, Szplet *et al.* utilised the multi-channel merging method to achieve a 2.9 ps resolution [152], and Chen *et al.* utilised the 2-D Vernier method to achieve a 2.5 ps resolution [114]. Although these methods are effective in enhancing resolutions, they also lead to a significant increase in hardware utilisation. Therefore, the WU method [115], which enhances the resolution with acceptable extra hardware utilisation, is the most used method to break the process-related limitation of the resolution.

WU TDCs, including WU A and WU B, were proposed by WU and Shi in 2008, achieving 25 ps and 10 ps precisions, respectively, in a Cyclone II FPGA [115]. As introduced in Chapter 2, the WU method is similar to the multi-chain merging method, enhancing the resolution and precision by using multiple measurements for the same TI. However, unlike multi-chain merging, the WU method achieves multiple measurements by inputting a pulse series (including both rising and falling edges) into the TDL rather than implementing parallel TDLs. Therefore, the WU method (including WU A and WU B) is much more hardware-efficient than multi-chain merging.

Since the WU method was proposed, it has been widely applied in high-resolution TDL-TDCs [141]. For example, Wang *et al.* implemented a 4-snapshot TDC (WU B) in a Virtex-4 FPGA with a resolution of 12 ps [168], Szplet *et al.* implemented a 6-edge TDC (WU A) in a Spartan-6 FPGA with a resolution of 5 ps [127], and Xie *et al.* implemented a 2-edge TDC (WU A) in an UltraScale FPGA with a resolution of 2.47 ps [82].

However, compared with plain TDL-TDCs, the encoding process of TDCs using the WU method is much more complex due to the need to locate multiple logical transitions in the TDL's output. Moreover, WU TDCs also face greater challenges when implemented in more advanced FPGA devices (manufactured by more advanced CMOS processes) due to the more severe bubbles. To my knowledge, no WU TDC was previously implemented in 16-nm FPGA devices. Therefore, this chapter designs and implements a 4-edge WU TDC (WU A) in a 16-nm UltraScale+ MPSoC device. This design combines the dual-sampling and WU methods for an ultra-high resolution. It also employs the sub-TDL and the proposed bidirectional encoder to suppress bubbles and encode bubble-free four-transition pseudo thermometer codes, respectively.

5.2 Architecture and design

As shown in Fig. 5.1, the TDC system consists of a start channel and a stop channel (Start Ch. and Stop Ch. in Fig. 5.1). These two channels are driven by the same clock and are responsible for recording start and stop timestamps for a TI, respectively. The timing diagram of TI measurement is shown in Fig. 1.2a. The coarse counter in Fig. 5.1 can be easily achieved using a binary or Gray-code counter. Therefore, this chapter focuses only on fine-time measurements with an ultra-high resolution.

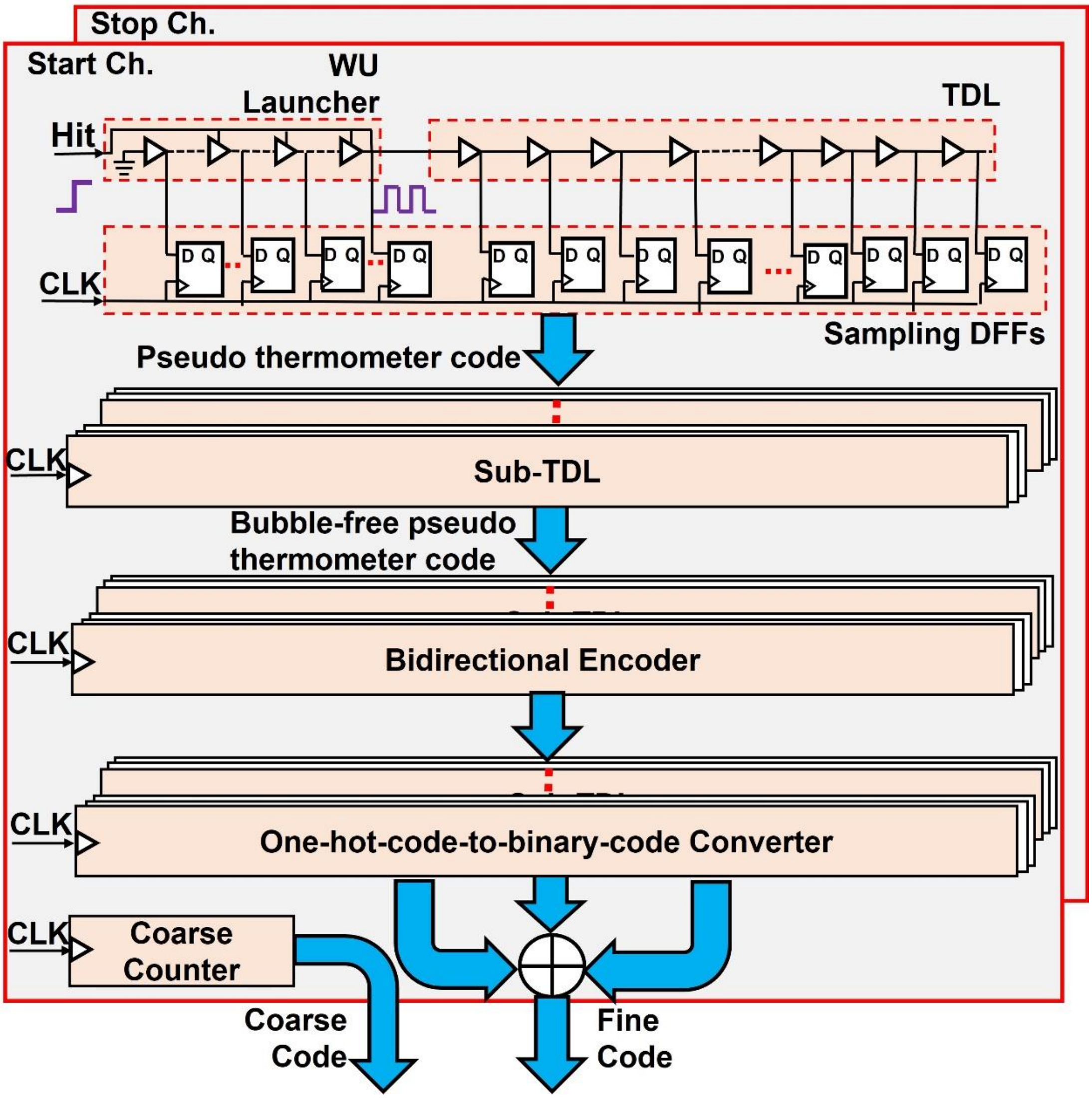


Figure 5.1 Diagram of the proposed TDC system.

5.2.1 CARRY8 and dual-sampling

When a hit is detected, the WU launcher generates a pulse series in each channel and inputs the series into the TDL, as shown in Fig. 5.1. When the sampling clock (CLK in Fig. 5.1) toggles from “0” to “1”, the TDL’s outputs are then simultaneously sampled by DFFs to measure the τ_{Start} and τ_{Stop} in Fig. 1.2a.

Therefore, as the delay cell, the CARRY8 is the backbone of the proposed TDC. Implemented by CARRY8s, the dual-sampling method [61], [82], [169] was first proposed by Wang and Liu [126]. As shown in Fig. 5.2, the TDL (highlighted in red) consists of cascaded CARRY8s, and each CARRY8 contains eight MUX-based delay elements [16]. Similar to its predecessor (CARRY4), each delay element in CARRY8s has two outputs (O and CO shown in Fig. 5.2). However, unlike CARRY4s, where either "O" or "CO" can be sampled within a single slice, CARRY8s allow for simultaneous sampling of both outputs by DFFs [15]. Therefore, each CARRY8 can output sixteen taps, equivalent to splitting one time bin into two time bins. Hence, the dual-sampling method can enhance the resolution without increasing the system's complexity.

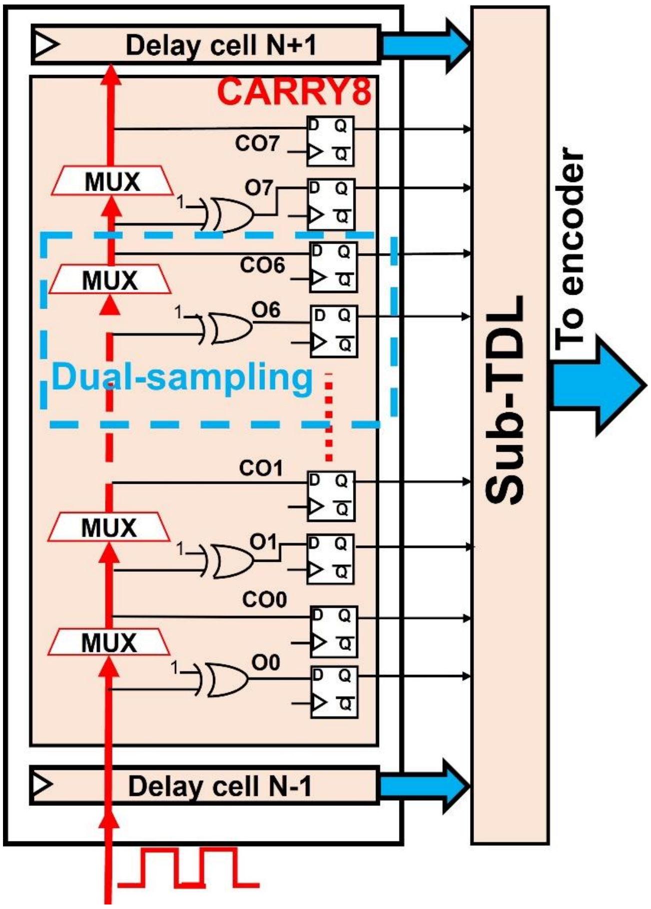


Figure 5.2 Diagram of the dual-sampling method with CARRY8.

5.2.2 Sub-TDL and WU launcher

Despite outputs from the TDL being sampled by DFFs, they cannot be encoded directly due to the presence of bubbles. As introduced in Chapter 2, bubbles are caused by uneven propagation delays and clock skews, and methods like the bin realignment [125] and the ones-counter [126] were proposed to tackle bubbles. However, these two methods are unsuitable for this design because the ones-counter cannot encode the output from WU TDCs correctly, and the bin realignment requires many iterations to cover all bubble scenarios, which is time-consuming.

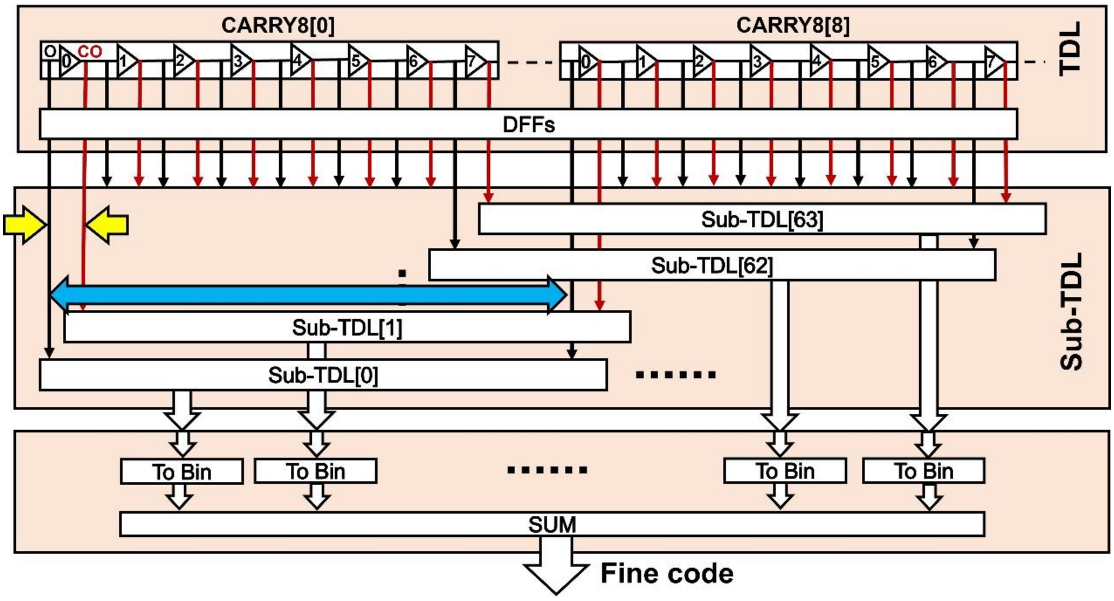


Figure 5.3 Architecture of the sub-TDL with the dual-sampling method.

Thus, this chapter uses the sub-TDL method to remove bubbles without increasing design complexity. Combined with the dual-sampling method, the sub-TDL is shown in Fig. 5.3. Similar to the design in Chapter 3, the sub-TDL extends TIs between taps by down-sampling TDL's taps to minimise the impact of uneven bin widths and clock skews for bubble-free outputs. However, the interval between adjacent taps in the sub-TDL depends on the MBD, and the observed MBD is sixty for the TDL with the dual-sampling method in the UltraScale+ MPSoC device. Therefore, to ensure all bubbles are removed, the interval between adjacent sub-TDL's taps in this design (highlighted in blue in Fig. 5.3) is set as sixty-four, a substantial increase from four in Chapter 3.

However, as shown in Fig. 5.4, the combination of the WU method and sub-TDLs may lead to undetectable logical transitions in sub-TDLs if the pulse width (highlighted in yellow in Fig. 5.4) is less than the bin width of the sub-TDLs (highlighted in blue in Fig.5.4). To avoid this and precisely control the pulse width, the WU launcher in this chapter is implemented using CARRY8s, as shown in Fig. 5.5. The input signal ("hit" in Fig. 5.5) is used as the select signal for the MUX-based carry element, allowing that the WU launcher can work in "Standby Mode" and "Launch Mode". When the input is

"0", the WU launcher stores the wave pattern in the carry chain in "Standby Mode". This pattern is then launched in "Launch Mode" once the input toggles to "1". The stored pattern is configured by the "S0" input (highlighted in green in Fig. 5.5) of the carry element, and the pulse width is determined by the number of delay elements between neighbouring "configure elements" (highlighted in blue in Fig. 5.5). This design configures the wave pattern as "01010" to include four logic transitions. Besides, in the designed WU launcher, the width of the positive pulse is configured as 80 taps (5 CARRY8s) in the dual-sampling TDL, ensuring all positive pulses are always detectable. However, the width of the negative pulse decreases when a pulse series propagates along the TDL due to rising edges propagating faster than falling edges [82]. Hence, the negative pulse in the designed WU launcher is configured as 112 taps (7 CARRY8s).

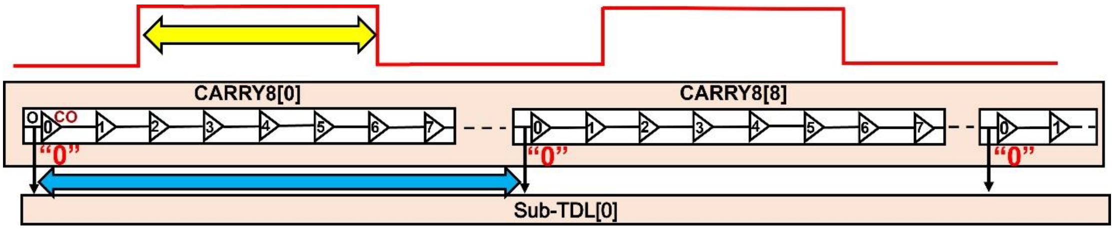


Figure 5.4 The concept of an undetectable pulse train in a sub-TDL.

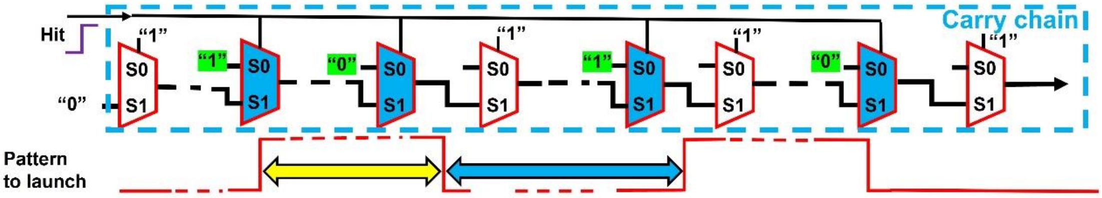


Figure 5.5 Block diagram of the wave union (WU A) launcher.

5.2.3 Bidirectional encoder

The TDC in Chapter 3 detects one rising and one falling edge in each sub-TDL and converts them into one-hot codes for subsequent binary-code generation, as shown in Fig. 5.6a. However, in this design, four logic transitions (including two rising edges and two falling edges) can always be detected in each sub-TDL, benefitting from the well-designed WU launcher. Therefore, the encoding strategy in Chapter 3 is unsuitable for the TDC proposed in this chapter due to the encoding errors shown in Fig. 5.6b. To address this issue, a bidirectional encoder is proposed to encode four-edge transitions.

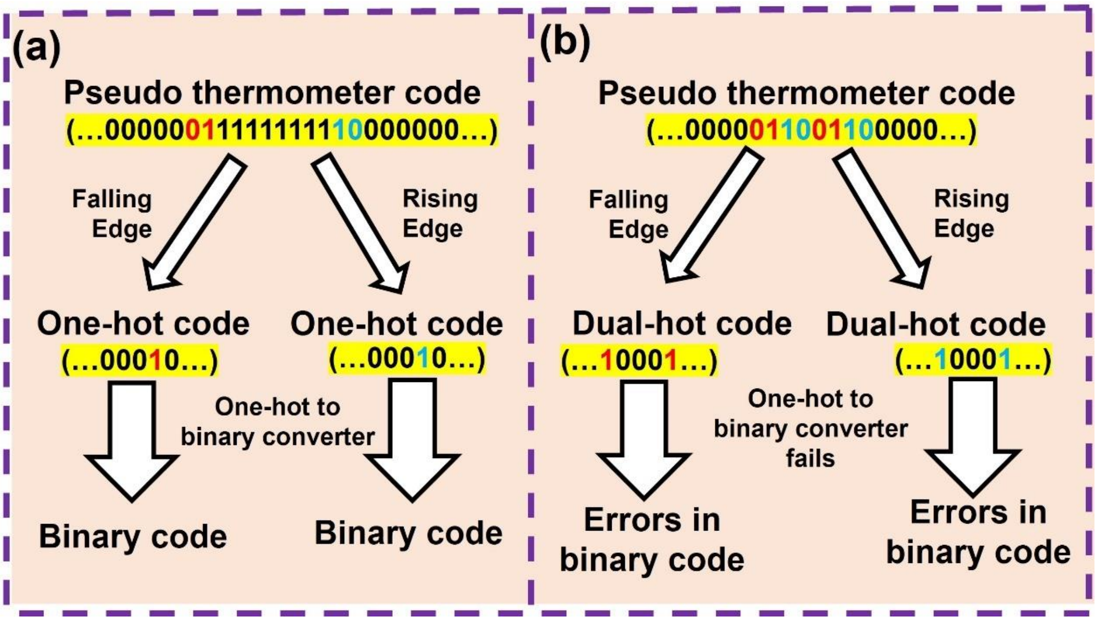
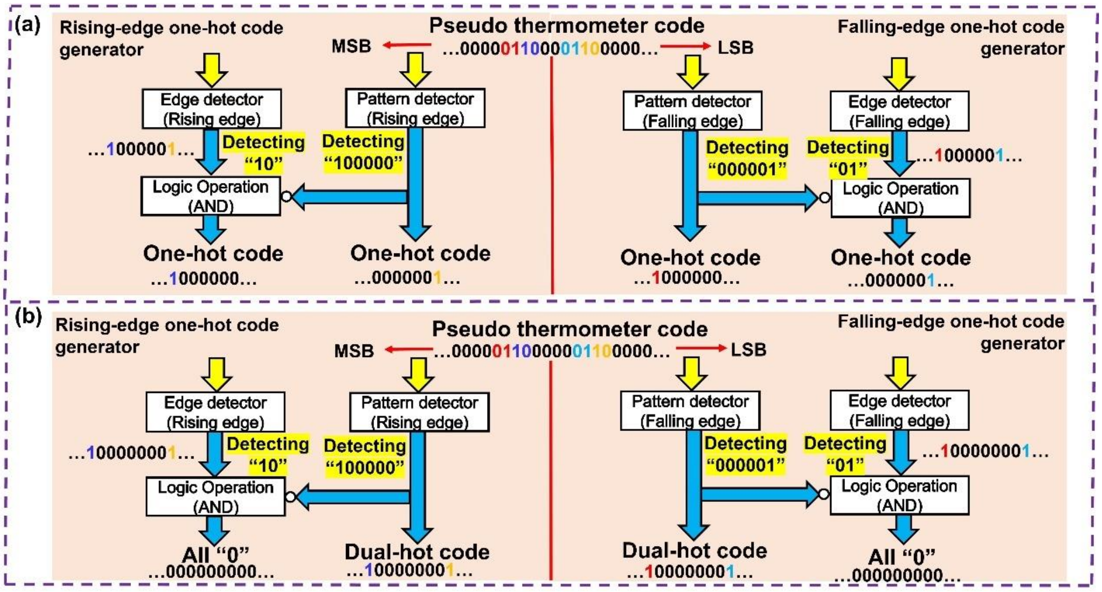


Figure 5.6 Encoding workflows of (a) two-edge WU A and (b) four-edge WU A TDCs with the encoding strategy in Chapter 3.

The bidirectional encoder's diagram and encoding flow are shown in Fig. 5.7a. It consists of rising-edge and falling-edge one-hot code generators responsible for converting a pseudo thermometer code into four one-hot codes. For a one-hot code generator, it contains an edge detector and a pattern detector, respectively responsible for detecting the specific logic-transition edge and the specific logic-transition pattern. For instance, in the rising-edge one-hot code generator, the pattern detector locates the pattern "100000", while the edge detector detects the rising edge "10". Then, the pattern detector can generate a one-hot code since the width of the negative pulse (highlighted in blue in Fig. 5.5) is precisely controlled to range from one tap to five taps in each sub-TDL. However, the pseudo thermometer code includes two "10" edges, as shown in Fig. 5.7a. Therefore, outputs from the edge detector and the pattern detector are input to an AND gate to get the other one-hot code for the rising edge, as shown in Fig. 5.7a. The mechanism of the falling-edge one-hot generator is similar to that of the rising-edge one-hot generator (shown in Fig. 5.7a). Differently, for the falling-edge detection, the pattern detector identifies "000001", and the edge detector identifies the edge "01".

Then, similar to the work in Chapter 3, all one-hot codes are converted into binary codes for the final result calculation. Besides, Fig. 5.7b also shows the encoding flow when the negative pulse exceeds 5 taps in a sub-TDL. Exceeding the 5-tap limitation in sub-TDLs causes unsuccessful one-hot code generation for the pattern detector, which also leads to an incorrect output from the AND gate.



(a)
Rising-edge one-hot code generator
Pseudo thermometer code
MSB
...000001100001100000...
LSB
Falling-edge one-hot code generator
Edge detector (Rising edge)
Pattern detector (Rising edge)
...1000001...
Detecting "10"
Detecting "100000"
Logic Operation (AND)
One-hot code ...1000000...
One-hot code ...0000001...
Pattern detector (Falling edge)
Edge detector (Falling edge)
Detecting "000001"
Detecting "01"
...1000001...
Logic Operation (AND)
One-hot code ...1000000...
One-hot code ...0000001...
(b)
Rising-edge one-hot code generator
Pseudo thermometer code
MSB
...00000110000001100000...
LSB
Falling-edge one-hot code generator
Edge detector (Rising edge)
Pattern detector (Rising edge)
...100000001...
Detecting "10"
Detecting "100000"
Logic Operation (AND)
All "0" ...000000000...
Dual-hot code ...100000001...
Pattern detector (Falling edge)
Edge detector (Falling edge)
Detecting "000001"
Detecting "01"
...100000001...
Logic Operation (AND)
Dual-hot code ...100000001...
All "0" ...000000000...

Figure 5.7 Block diagram and encoding flow of the bidirectional coder when the width of the negative pulse in the sub-TDL is (a) less than 5 bits and (b) more than 5 bits.

Fig. 5.8 shows the hardware implementation of the bidirectional encoder. A 6-input LUT is used to implement the pattern detector, while a 3-input LUT is used for the edge detector and the AND gate. In the design of this chapter, all LUTs for the bidirectional encoder are instantiated by Vivado Primitive [18]. However, the configurations of LUTs differ between the rising and falling-edge one-hot code generators. Fig. 5.9 shows the truth tables for the LUTs.

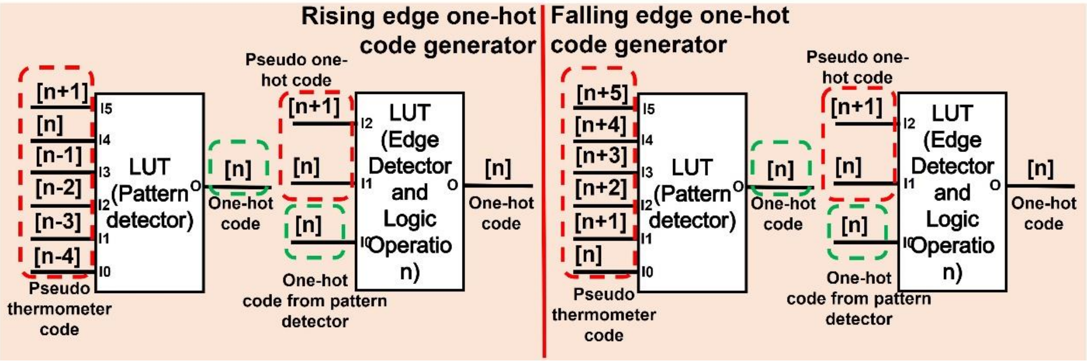
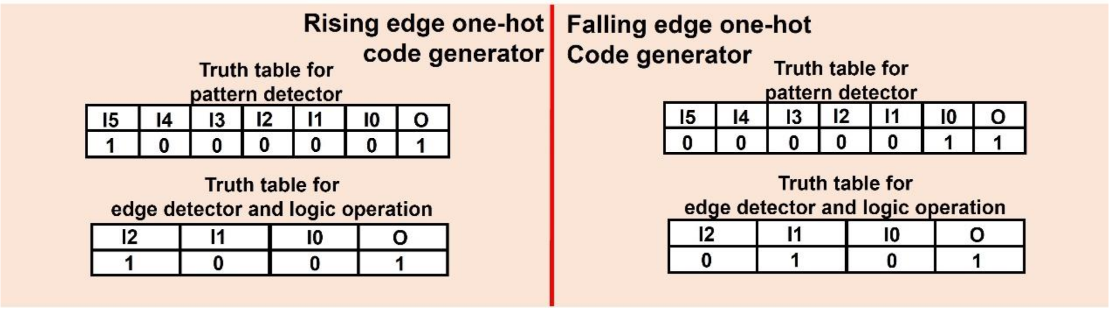


Figure 5.8 Hardware implementation of the bidirectional encoder.


Rising edge one-hot code generator

Truth table for pattern detector

I5	I4	I3	I2	I1	I0	O
1	0	0	0	0	0	1

Truth table for edge detector and logic operation

I2	I1	I0	O
1	0	0	1

Falling edge one-hot Code generator

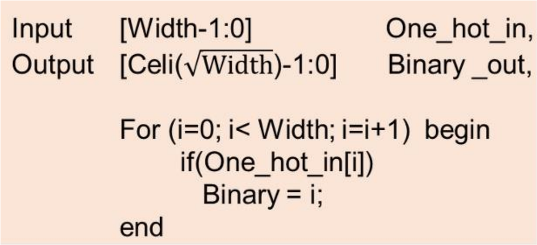
Truth table for pattern detector

I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	1	1

Truth table for edge detector and logic operation

I2	I1	I0	O
0	1	0	1

Figure 5.9 Truth tables for the bidirectional encoder (LUTs in other cases output “0”).


```
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
```

Figure 5.10 Pseudo codes of the one-hot-to-binary-code converter.

As shown in Fig. 5.1, one-hot codes from the bidirectional encoders are input into one-hot-code-to-binary-code converters, generating corresponding binary codes following the pseudo codes shown in Fig. 5.10. These binary codes are then summed together by 4-input adders, and the final result is sent to the personal computer (PC) using a universal asynchronous receiver-transmitter (UART). On the PC, the timestamp of each bin is determined by SCDTs [67] and the bin-by-bin calibration method [68] [details are expressed by Eq. (2.4) and Eq. (2.6)]. Finally, τ_{Start} and τ_{Stop} shown in Fig. 1.2a can be evaluated with the calibrated timestamps. This design uses sequential logic circuits for sampling TDLs, sub-TDLs, bidirectional encoders, one-hot-code-to-binary-code converters, and adders, as shown in Fig. 5.1. Benefiting from sequential logic design, the encoding flow is pipelined, with the total latency (from sampling DFFs to adders) being eight system clocks (it takes one clock to output results from sampling DFFs, sub-TDLs, bidirectional encoders, and one-hot-code-to-binary-code converters. However, summing all 64x4 subsets in parallel by the 4-input adder consumes four system clocks).

5.3 Experimental results

The proposed TDC was implemented in a ZCU-104 evaluation board shown in Fig. 4.17 [158]. Besides, the start and stop channels (Start Ch. and Stop Ch. shown in Fig. 5.1) were placed closely to minimise the offset. To evaluate TDC's performance, an SRS CG-635 [161] was utilised as an external signal source. The same signal was concurrently input into both channels to reduce measurement errors from the cables connecting the signal source and the evaluation board. This input can be treated as a random hit for the two channels since it is asynchronous with the TDC's sampling clock. Moreover, the period of the input signal was also used as a TI for RMS precision tests. The offset between the two channels was measured by calculating the difference between the same edge's timestamps recorded by the two channels. The sampling clock (also known as the system clock) was from an onboard crystal oscillator (IDT-8 T49) with a frequency of 450MHz.

5.3.1 Bin width and linearity

The bin width represents the quantization step of each time bin. To ensure the TDL can cover the entire sampling period, the TDL's length was increased until two continuous outputs could be detected when the input hit (with a 99.7777777 MHz frequency and less than 1 ps RMS) appeared near the rising edge of the sampling clock. Then, twenty million random hits were input into the two channels for the SCDTs to determine the bin width of each time bin. Moreover, linearity (including DNL and INL), equivalent bin width (ω_{eq}), and its standard deviation (σ_{eq}) [68] were also used to characterise the TDC's performance. These metrics are summarised in Table 5-1.

Figure 5.11 shows the bin width of the proposed TDC. In both channels, the first valid time bin (not a zero-width bin) appears at around the 1250-th bin, caused by the WU launcher. A segment of the carry chain constructs the WU launcher to generate and store a pulse series before a hit comes. Therefore, although there is no input, the TDC's output is non-zero, and it increases when an input comes. Besides, a cluster of narrow bins appears at the tail of the valid time bins in both channels, caused by clock jitters.

Table 5-1 Performance of the proposed TDC

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]
ω_{eq} (ps)	1.81	ω_{eq} (ps)	1.85
σ_{eq} (ps)	0.52	σ_{eq} (ps)	0.53

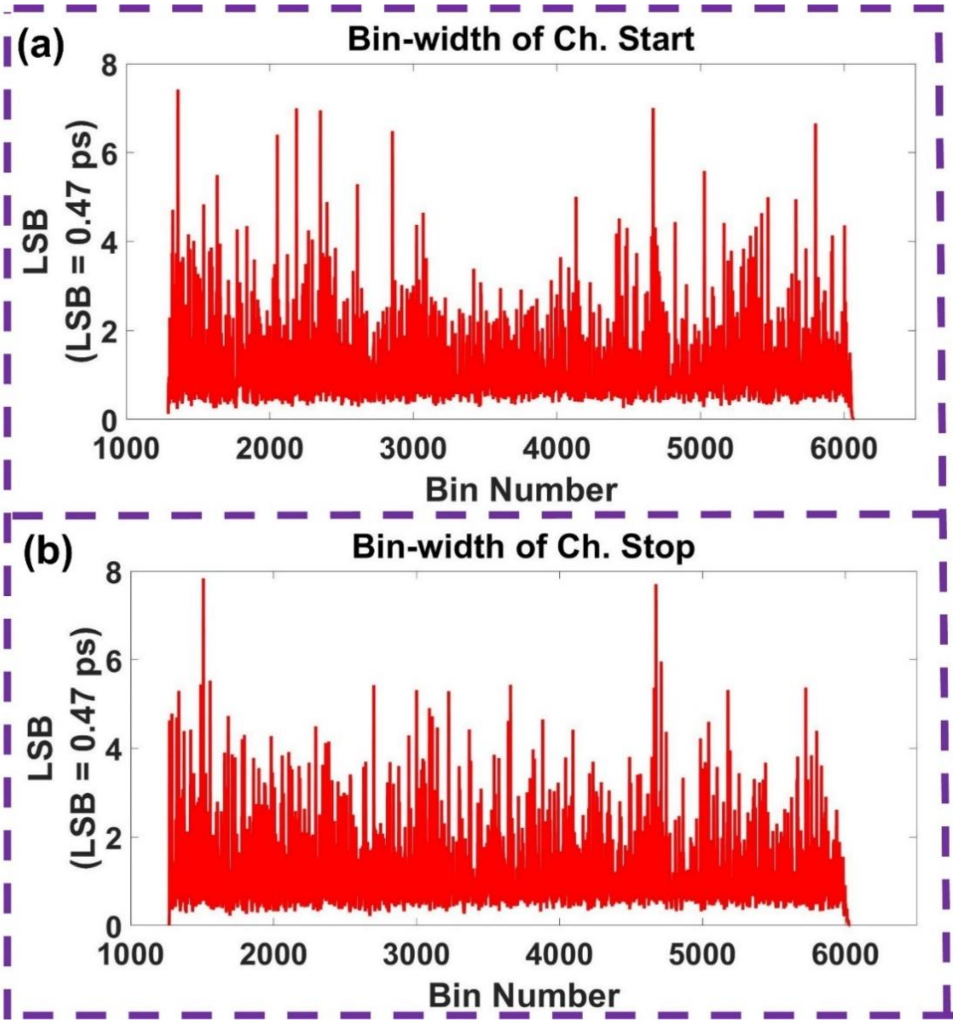


Figure 5.11 Bin width of (a) the start channel and (b) the stop channel.

5.3.2 Time interval tests

The RMS precision quantifies the measurement uncertainty caused by jitters and quantization errors, and it can be calculated by the standard deviation (σ) of repetitive measurements for a fixed TI. Moreover, the bin-by-bin calibration [68] can enhance the RMS precision.

In TI tests, the same signal is input into two channels simultaneously to minimise external measurement errors. Then, ten thousand samples are captured for a fixed TI. Fig. 5.12 shows the RMS precision of the proposed TDC. The TI ranges from 0 ns to 100 ns, increasing in a 5-ns step. Among all measured TIs, the best RMS precision appears when the TI equals 0 ns (shown in Fig. 5.13), achieving 3.06 ps corresponding to a 2.16 ps single-shot precision (SSP, SSP = RMS / $\sqrt{2}$). Besides, the average value of measurements in this scenario is 206.73 ps, equal to the offset between two channels (Ch. Start and Ch. Stop in Fig. 5.1). RMS precision deteriorates with the TI increasing, achieving the worst RMS precision of 8.97 ps at TI = 30 ns (as shown in Fig. 5.14, corresponding to a 6.34 ps SSP). The RMS precision fluctuates between 5 ps and 9 ps in the measurement range, except for TI = 0 ns. This phenomenon is caused by the combination of jitters from the coarse-counting clock [influencing *T* in Eq. (1.1)] and 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. 5.15). However, for the rest of the TI values in TI tests, the coarse counter is always required because TIs are more than one coarse-counting period. Besides, the delay is only caused by the internal offset between channels when TI = 0 ns. However, the delay of the external signal source is required for other scenarios (such as TI = 5 ns), causing the jitter from the signal source to be introduced into the measurement. Due to these two reasons, the RMS precision at TI = 0 ns is much better than that of the other TIs. Moreover, the measurement error is also analysed as:

$$E = \left(\mu - T_{offset}\right) - T_{actual}, \tag{5.1}$$

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 in Fig. 5.16 indicate the design in this chapter has less than 3 ps measurement errors in the measurement range from 5 ns to 100 ns.

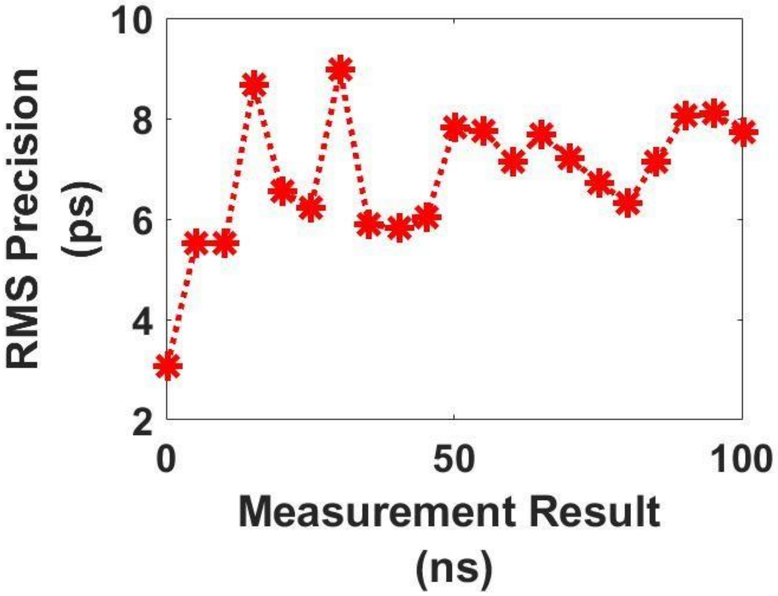


Figure 5.12 RMS precision of the proposed TDC.

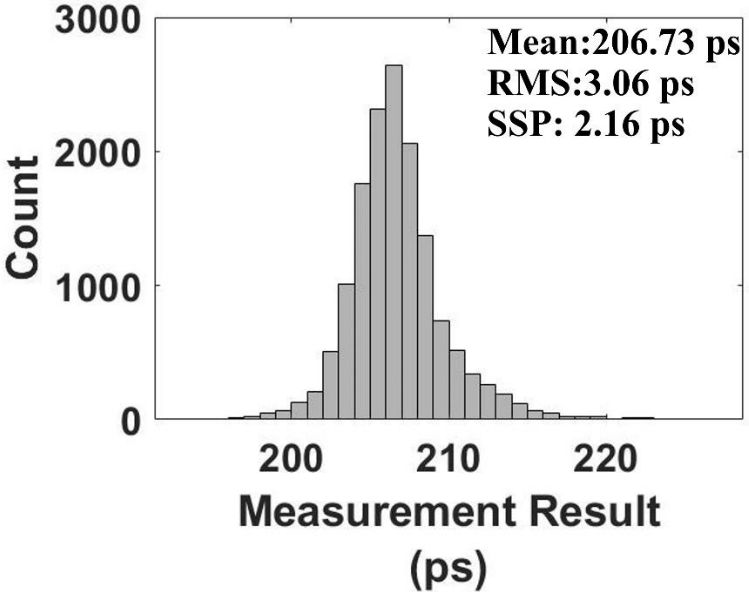


Figure 5.13 Measurement histogram for the TI = 0 ns.

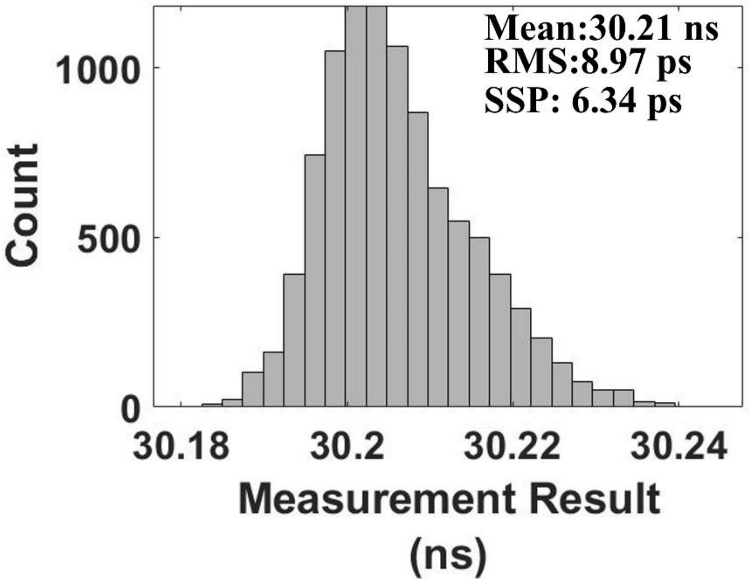


Figure 5.14 Measurement histogram for the TI = 30 ns.

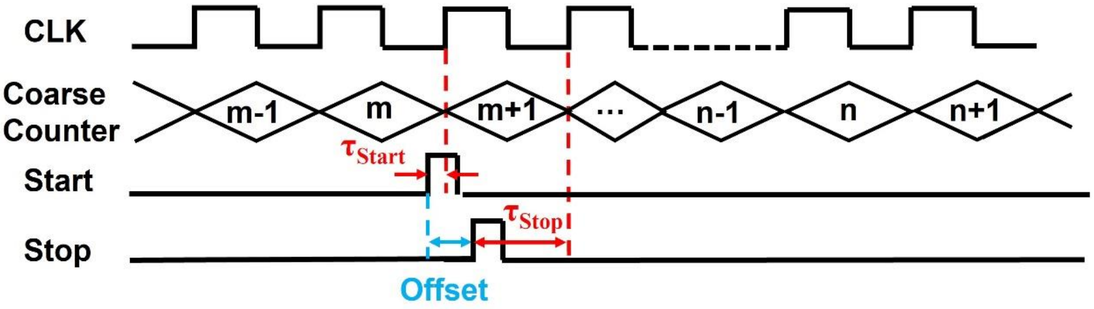


Figure 5.15 Timing diagram for the measurement with the coarse counter when TI = 0 ns (offset measurement).

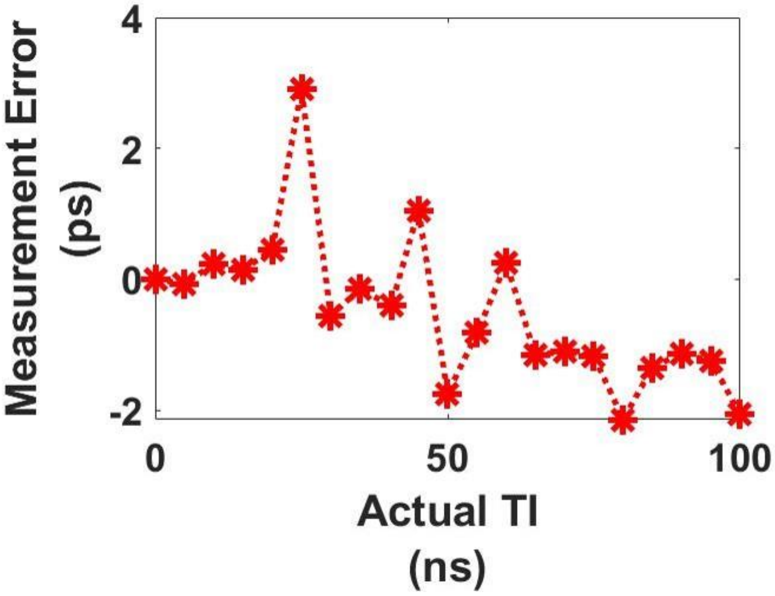


Figure 5.16 Measurement errors of the proposed TDC.

5.3.3 Hardware implementation

Table 5-2 summarises the hardware utilisation of the proposed TDC. Each channel has 1920 taps from the dual-sampling TDL, consuming 234 CARRY8s, 13547 DFFs, and no more than 11782 LUTs. Specifically, 22 CARRY8s are used to construct the WU launcher, and 98 CARRY8s are used for the TDL. Simultaneously, extra 1920 DFFs and 960 LUTs are used as sampling circuits to sample the outputs from the WU launcher and TDL. In addition to this, the remaining hardware utilisation is primarily for encoding, especially for the proposed bidirectional encoder. In the proposed 30-bit-input rising edge (falling edge) one-hot code generator, the pattern detector consumes 29 LUTs, while the combination of the edge detector and the AND gate consumes 28 LUTs. However, the EDA tool (Xilinx Vivado) inserts LUTs when DFFs sample outputs from the bidirectional encoder, leading to the global bidirectional encoder consuming 8976 LUTs and 7296 DFFs (including combinational logics and sequential logics for all 64 sub-TDLs' bidirectional encoders). And the inserted 1680 LUTs are used to generate a “hit detected” signal (to claim a valid input is detected) for each sub-TDL. Moreover, the hardware utilisation of the encoder also depends on the number of taps and the implemented device. With the dual-sampling method, the proposed TDC requires at least 1250 taps (about 78 CARRY8s) to cover the entire period of the sampling clock due to the low propagation delay of CARRY8s. Besides, extra 352 taps (22 CARRY8s) are required for wave pattern generation. They both increase the number of time bins, leading to the encoders’ significant hardware utilisation.

Figure 5.17a shows the placement of the start and stop channels. They are placed closely to minimise the offset. Besides, the WU launcher is manually routed to ensure steady wave pattern generation. The hardware implementation of the 4-edge WU launcher is shown in Fig. 5.17b.

Table 5-2 Hardware utilisation of the proposed TDC.

Resource	Utilisation (%)	
	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%)

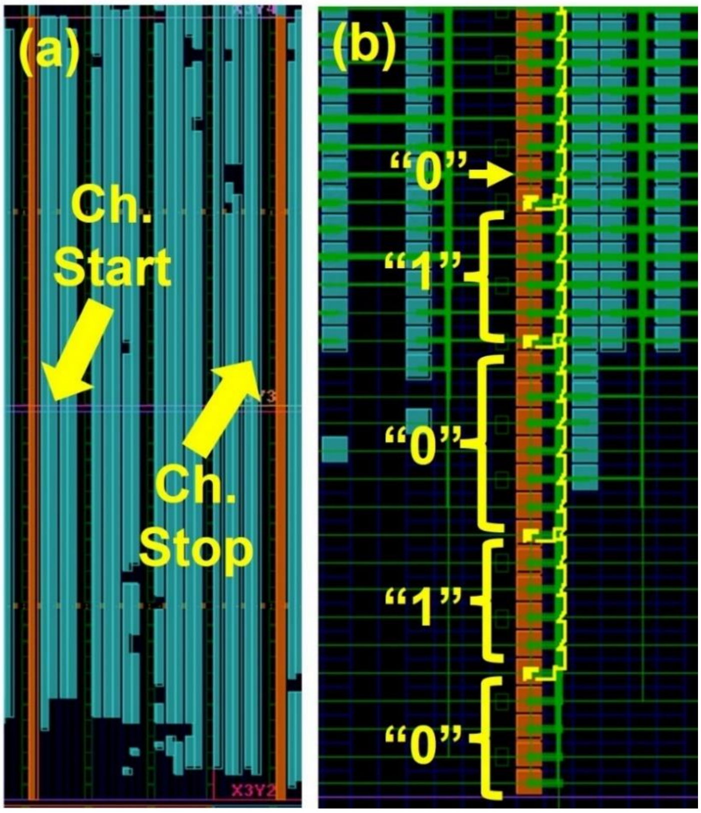


Figure 5.17 (a) Placement of the start and the stop channel, and (b) hardware implementation of the WU launcher.

5.4 Comparison

Table 5-3 Comparisons of recently published WU TDCs

Ref-year	Method	Device	LSB (ps)	RMS (ps)	LUT (%)[1]	DFF (%)[1]	CARRY (%)[1]	Real-time/Post Encoding
[127]-16	Super WU (6 edges , 3 coding lines)	Spartan-6	0.90	<6	-	-	-	-
[142]-19	WU-A (4 edges), Bin-by-bin Cali.	Kintex-7	2.65[2]	3.5	1410[2] (1.38)	2732[2] (1.34)	-	Real-time Encoding
	WU-A (8 edges), Bin-by-bin Cali.		1.33[2]	3.0	2005[2] (1.98)	3751[2] (1.85)	-	
[170]-19	Super WU (2 edges , 8 coding lines)	Kintex-UltraScale	0.31	12.32	-[3]	-[3]	-	Real-time Encoding
[171]-21	MSWU Bin-by-bin Cali.	Kintex-7	0.39	3.30[4]	600	800	200[5]	Post Encoding
This Work	**WU-A (4 edges), Sub-TDL, Dual-sampling, Bidirectional Encoder**	**UltraScale+ MPSoC**	**0.46**	**<9 (3.06)[6]**	**11773 (5.11)**	**13547 (2.94)**	**234[7] (0.81)**	**Real-time Encoding**

[1] Percentage of resource utilisation for the implemented devices; [2] Calculate from the literature; [3] 3200 SLICEs, 400 kbit RAM and 1 DSP are used; [4] Value calculated from single-shot precision; [5] CARRY4s; [6] RMS precision in the best-case scenario; [7] CARRY8s.

Recently published high-resolution WU TDCs are summarised in Table 5-3. TDCs in Refs. [127] and [170] utilised the super WU method (combining WU A and multi-chain merging) to achieve sub-picosecond resolutions but suffering from significant hardware utilisation. Although the multi-sampling WU method (MSWU, a method combining WU A and WU B) in Ref. [171] can achieve a high resolution with low hardware utilisation, its encoding is complex, and it is hard to perform real-time encoding in the implemented device. Therefore, compared with the MSWU-TDC [171], the proposed TDC has a trade-off between encoding complexity and hardware utilisation. Moreover, it achieves real-time encoding for a 4-edge pulse series with the proposed bidirectional encoder.

Compared with other 4-edge WU TDCs featuring real-time encoding, such as the design in Ref. [142], the TDC proposed in this chapter also has competitive hardware utilisation efficiency. This design is similar to the 4-edge WU TDC in Ref. [142], except for the encoder. Due to the high propagation delay of delay cells, the TDC in Ref. [142] requires 288 taps to build the WU launcher and cover the entire sampling period (with a 554MHz sampling clock), consuming 1410 LUTs and 2732 DFFs for each channel. In contrast, due to the low propagation delay of delay cells in the implemented device, the designed TDC in this chapter requires more than 6.5-fold taps (1920) for the WU launcher and dual-sampling TDL, consuming 8.35-fold LUTs and only 4.95-fold DFFs. The comparison of the hardware utilisation between the proposed TDC and TDC in Ref. [142] indicates that these two designs have similar hardware utilisation efficiency. However, this design only requires configuring LUTs for four cases (truth tables are shown in Fig. 5.9), reducing the implemented complexity significantly compared with the design in Ref. [142].

5.5 Conclusion

In this chapter, a 4-edge WU TDC, combining the dual-sampling and sub-TDL methods, was implemented in a 16-nm UltraScale+ MPSoC device. This chapter also proposed a bidirectional encoder to encode the 4-transition pseudo thermometer code in real-time, achieving a resolution of 0.46 ps and an RMS precision of less than 9 ps, with a measurement error below 3 ps. Moreover, it details the hardware implementation of the bidirectional encoder and the WU launcher. Experimental results indicate that the proposed TDC is suitable for particle physics, biomedical imaging (such as FLIM), and scientific instruments.

Chapter 6 Two-stage interpolation TDCs implemented in multiple FPGA devices

6.1 Motivation

Chapter 4 proposed a GCO-TDC with low hardware utilisation. However, its resolution is inherently limited, even with a sampling matrix. Hence, Chapter 5 proposed a TDL-based WU-TDC to achieve an ultra-high resolution but with extremely high hardware utilisation. Compared with the GCO-TDC, conventional TDL-TDCs have significant hardware utilisation for two primary reasons: 1) a large number of delay cells (CARRY4s/CARRY8s) are required to cover the entire sampling period due to the low propagation delay of delay cells, and 2) complex encoding circuits are necessary, responsible for removing bubbles and converting the output from the TDL into a binary code. Moreover, the hardware utilisation of encoding circuits is proportional to the length of the TDL. Hence, most TDL-TDCs prefer a high-frequency sampling clock to reduce the length of the TDL, thereby reducing hardware utilisation of encoding circuits. For example, in Ref. [125], only 131 CARRY4s are required to cover a 1.408 ns sampling period corresponding to a 710 MHz sampling frequency, and in Ref. [142], 216 CARRY4s are required to cover a 1.805 ns sampling period corresponding to a 544 MHz sampling frequency. However, the sampling frequency of TDL-TDCs cannot be faster than the maximum operation frequency of FPGAs, indicating this method still has a limitation. To address this limitation and further reduce the hardware utilisation of TDL-TDCs, this chapter proposes a two-stage interpolation TDC, aiming at less hardware utilisation but maintaining resolutions compared with plain TDL-TDCs.

The proposed TDC employs a LUT-based Vernier GCO-TDC (VGCO-TDC) for the first-stage interpolation and utilises the TDL-TDC for the second-stage interpolation. The TDL-TDC only measures the overtaking residue from the GCO-TDC in the proposed TDC. Therefore, without degrading resolutions compared with plain TDL-

TDCs, the TDL only needs to cover the resolution of the VGCO-TDC, reducing the length of the TDL and the TDL-TDC's hardware utilisation.

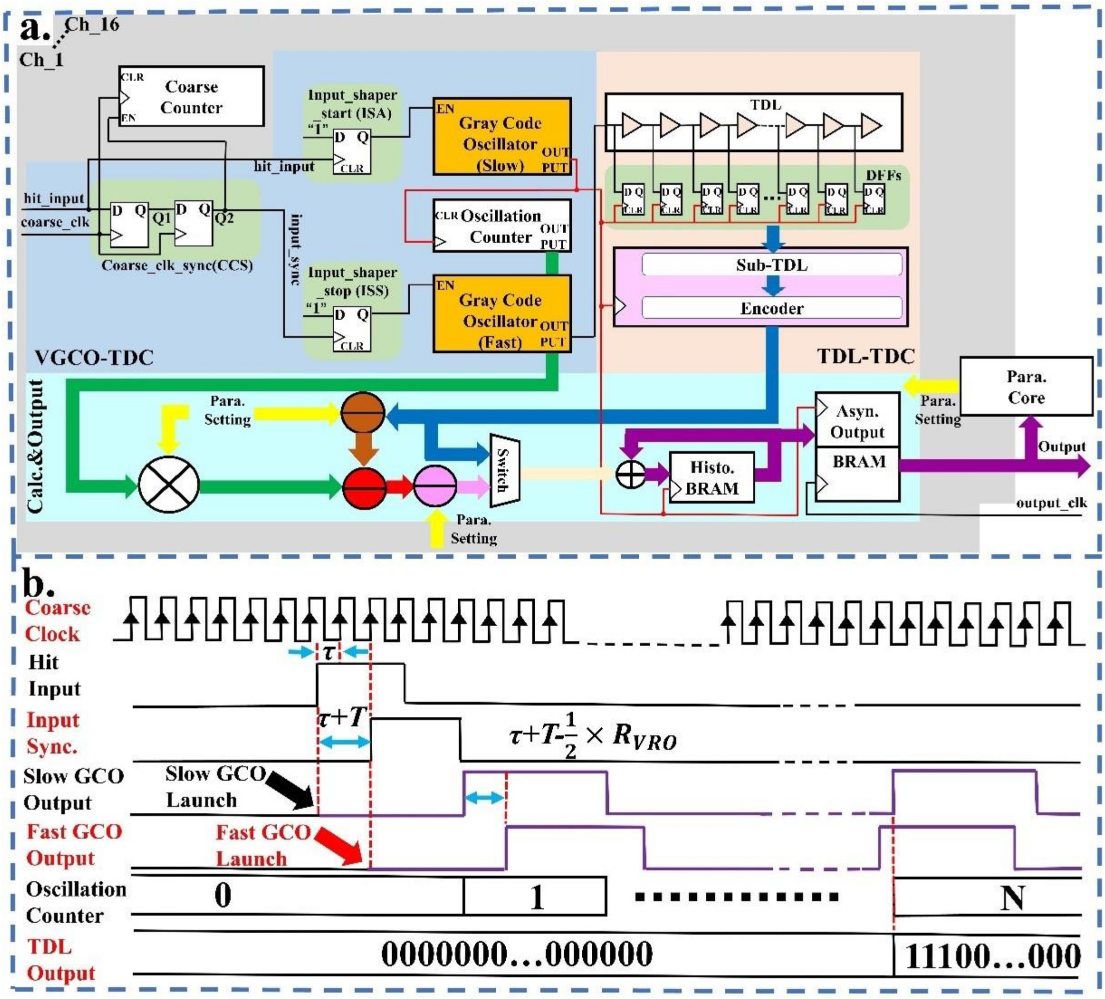


Figure 6.1 (a) Bock diagram and (b) timing diagram of the two-stage interpolation TDC.

6.2 Architecture and design

As shown in Fig. 6.1a, the designed TDC works with a coarse counter, and each fine counter consists of a VGCO-TDC, a TDL-TDC, and a calculation and output module (Calc.&Output in Fig. 6.1a). The VGCO-TDC is responsible for the first-stage fine-time measurement with a resolution of several hundred picoseconds. Then, for the second-stage fine-time measurement, the TDL-TDC measures the VGCO-TDC's overtaking residue (δ) with a resolution of less than 10 ps. After accomplishing both measurements, the VGCO-TDC and TDL-TDC output the measurement results from the oscillation counter and encoder shown in Fig. 6.1a, respectively. These two results are then processed in the Calc.&Output module to obtain the final result. Moreover, BRAMs are used for the on-chip histogram and asynchronous output (Histo. BRAM and Asyn. Output BRAM in Fig. 6.1a), respectively. For parameters (highlighted in yellow in Fig. 6.1a) input into the multiplier and subtractors, a state-machine-based parameter core (Para. Core in Fig. 6.1a) is used to calculate and then set them channel-by-channel based on SCDT results stored in Histo. BRAMs.

6.2.1 Measurement principle

When a hit comes, the input hit respectively launches the slow and fast GCOs (highlighted in orange), as shown in Fig. 6.1a. For launching the slow GCO, the input hit first arrives at the Input_shaper_start (ISA) and changes this module's output to "1" when the input hit's rising edge occurs. Then, the output of the ISA remains "1", keeping the slow GCO running until the global asynchronous clear (CLR in Fig. 6.1a) is asserted. Simultaneously, the input is also transferred to the fast GCO. Differently, the input hit first reaches the Coarse_clk_sync (CCS) module and is synchronous with the coarse-counting clock (coarse_clock in Fig. 6.1a) after two rising edges of the coarse-counting clock. Then, the synchronised input (input_sync in Fig. 6.1a) launches the fast GCO, similar to the input launching the slow GCO. Fig. 6.1b shows the timing diagram of the proposed TDC. The resolution of the VGCO-TDC (R_{VRO} in Fig. 6.1b) is determined by the oscillation speed difference between two GCOs. The output of the oscillation counter is stored immediately when the fast GCO first overtakes the slow GCO. However, the TI between two GCOs' launch is (τ+T) rather than τ, where τ is the measured TI between the rising edges of the input hit and the subsequent coarse-counting clock. Therefore, T should be subtracted from the VGCO-TDC measurement result. However, an extra T between GCOs' launches is necessary. Without this, when an input appears close to the rising edge of the coarse clock, the launching sequence of GCOs can disorder due to uneven propagation delays of inner connections, causing the VGCO-TDC to work incorrectly.

Unlike conventional VRO-TDCs, which employ a DFF as the phase detector to detect the overtaking [99], [172], this design uses a TDL-TDC for this purpose. The TDL-TDC is also used to measure the δ at a ~10 ps resolution. For the TDL-TDC in this design, the output from the fast GCO is input into the delay line, and the slow GCO's output is used as the clock of the sampling DFFs, encoder, and so on, as shown in Fig. 6.1a. When the fast GCO lags behind the slow GCO, the sampled outputs from the TDL are "0"s. But a thermometer code ("11100…000") is output when the fast GCO first overtakes the slow GCO, as shown in Fig. 6.1b. This thermometer code is then converted into a binary code, representing the measurement of δ. Therefore, the measured TI is calculated as:

$$\tau = \left(N_{Osci} + \frac{1}{2}\right) \times R_{VRO} - T - \delta, \qquad (6.1)$$

where N_{Osci} is the oscillation number of the slow GCO when the fast GCO first overtakes the slow GCO.

6.2.2 VGCO-TDC and TDL-TDC

Vernier-TDCs utilise the oscillation speed difference between fast and slow oscillators for fine-time measurements. In theory, two asynchronous clocks with different frequencies are appropriate for Vernier-TDCs. However, since each oscillator operates independently, a Vernier-TDC requires at least two phase-locked loops (PLLs) or mixed-mode clock managers (MMCMs) [17], [18], which is unaffordable for multi-channel designs. To address this, GCOs are used to construct the VGCO-TDC. Unlike the direct measurement of TIs by a single GCO presented in Chapter 4, this design uses a pair of GCOs in the Vernier way (as slow and fast oscillators). The GCO is immune to the "race and competition" phenomenon since only one bit toggles between two continuous states in the Gray code, as introduced in Chapter 4. Hence, as shown in Fig. 6.2a, the GCO is implemented by LUTs and operates without any driving clocks. The

“EN” signal (highlighted in red in Fig. 6.2a) is used to launch and reset the GCO. Moreover, only the G[4] signals in Fig. 6.2a are output from the slow and fast GCOs and are input into the TDL-TDC as the sampling clock and the TDL’s input, respectively. The timing diagram of the GCO is shown in Fig. 6.2b.

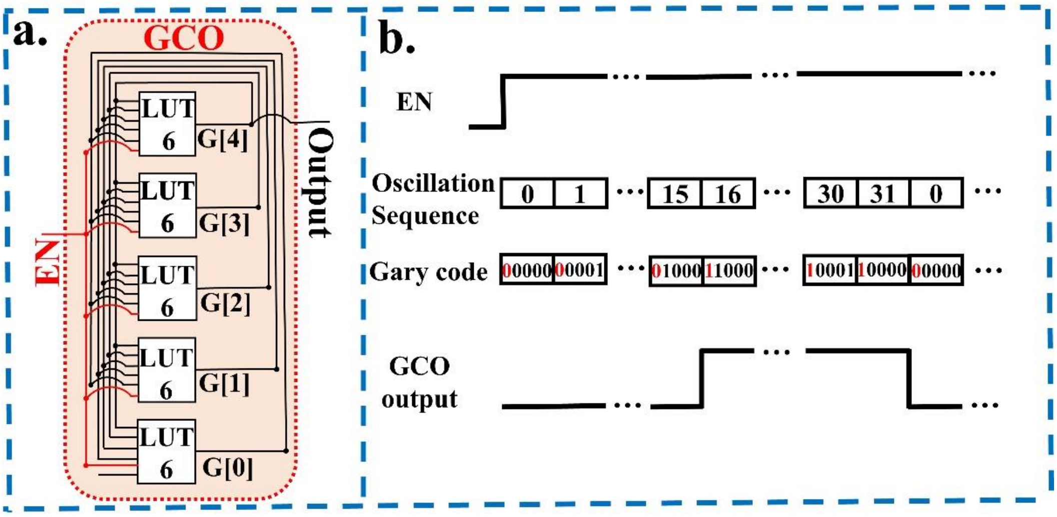


Figure 6.2 (a) Block diagram and (b) timing diagram of the GCO.

As shown in Fig. 6.1a, the proposed two-stage interpolation TDC uses the TDL-TDC to measure the δ from the VGCO-TDC. Since δ is less than the R_{VRO}, the TDL in the designed TDC only needs to cover the resolution of the VGCO-TDC, further reducing the length of the TDL and the hardware utilisation of encoding circuits. However, bubbles still appear due to uneven propagation delays and clock skews. Therefore, the sub-TDL method is also used, and it split TDL’s outputs into 4/8 groups (4 groups in the Virtex-7 FPGA and 8 groups in the Kintex-UltraScale FPGA) at a constant interval to remove bubbles [83]. Then, outputs from all sub-TDLs are encoded into binary codes, which are summed together as the result of the TDL-TDC.

6.2.3 Result calculation and parameter configuration

The TI is measured, and the corresponding results are output from the VGCO-TDC and TDL-TDC, respectively. However, these two outputs still require further post-processing for the final result. For this purpose, a Calc&Output module shown in Fig. 6.1a is employed. The post-processing follows Eq. (6.1). However, the TDL-TDC’s output is a raw binary code rather than a calibrated timestamp, making it unsuitable for direct use as δ in Eq. (6.1). Therefore, the designed two-stage interpolation TDC outputs a raw binary code directly instead of a calibrated timestamp, considering the complexity of the hardware-implemented bin-by-bin calibration [68]. The bin-by-bin calibration is then conducted on the PC, as shown in Fig. 6.3. Referring to Eq. (6.1), the two-stage interpolation TDC's output can also be calculated as:

$$Out_{TDC} = N_{Osci} \times N_{TDL} - Output_{TDL} - Offset, \tag{6.2}$$

where N_{TDL} is the number of TDL-TDC's time bins covering the VGCO-TDC's resolution, $Output_{TDL}$ is the raw output from the TDL-TDC, and $Offset$ is the measurement offset caused by the CCS module, uneven routing delays, etc. Moreover, in the second-stage sub-TDL TDC, the output is valid only when all sub-TDLs have non-zero outputs, making the minimal output from the TDL-TDC more than 1. To cancel this offset, N_{TDL} and $Output_{TDL}$ are then calculated as:

$$N_{TDL} = Out_{max} - Out_{min} + 1, \tag{6.3}$$

and

$$Output_{TDL} = Out_{bin} - Out_{min} + 1, \tag{6.4}$$

where Out_{bin}, Out_{max} and Out_{min} are the raw binary code, the maximum and the minimum output from the TDL-TDC.

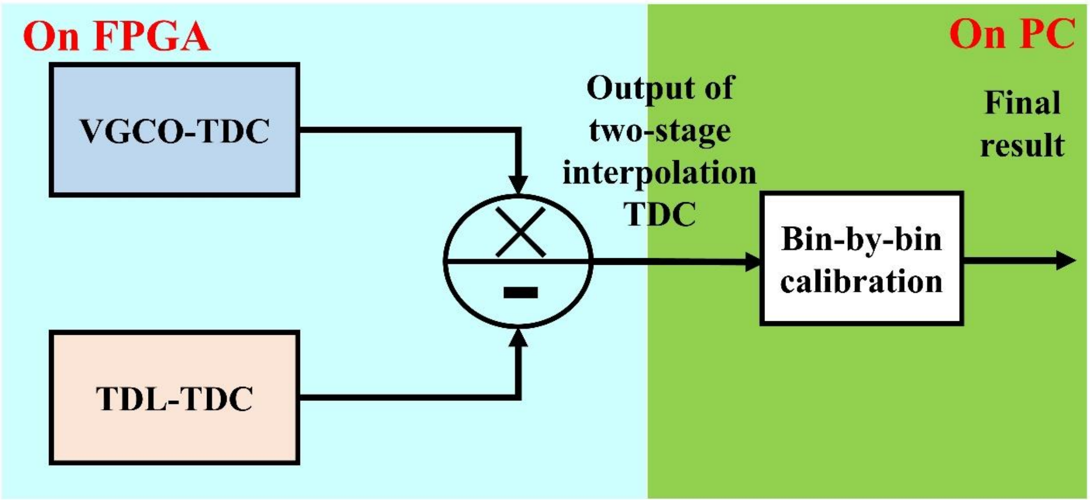


Figure 6.3 Data flow of the proposed two-stage interpolation TDC.

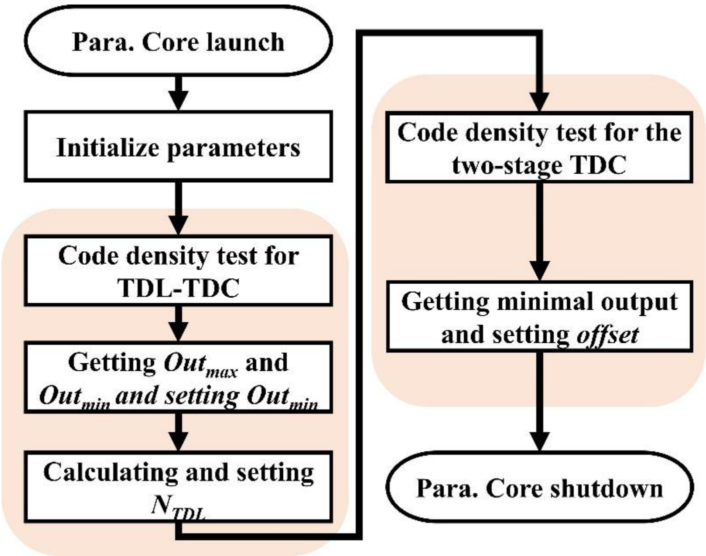

Para. Core launch
Initialize parameters
Code density test for TDL-TDC
Getting Out_{max} and Out_{min} and setting Out_{min}
Calculating and setting N_{TDL}
Code density test for the two-stage TDC
Getting minimal output and setting *offset*
Para. Core shutdown

Figure 6.4 Workflow of the Para. Core.

The parameters in Eq. (6.2) can be calculated and configured manually. However, manual configuration is time-consuming when applied to all 16 channels. Hence, a Para. Core shown in Fig. 6.1a is developed to automate the calculation and configuration of parameters for the two-stage interpolation TDC. The backbone of the Para. Core is a state machine, and the workflow of the Para. Core is shown in Fig. 6.4. Its operations require two SCDTs [8], referred to as SCDT1 and SCDT2. The SCDT1 is only for the TDL-TDC, with the switch in Fig. 6.1a selecting data from TDL-TDC (highlighted in blue in Fig. 6.1a) as the output. While the SCDT2 is for the two-stage interpolation TDC, with the switch selecting data after calculation (highlighted in pink in Fig. 6.1a).

Based on the result of SCTD1, the Para. Core can extract Out_{max} and Out_{min}, and calculate N_{TDL}. The Para. Core then configures N_{TDL} as the coefficient for the multiplier and configures Out_{min} as the subtrahend for the subtractor (both components are shown in Fig. 6.1a, with the subtractor highlighted in brown). After configuration, SCDT2 is performed for the two-stage interpolation TDC. $Offset$ in Eq. (6.2) can be extracted and is configured as the subtrahend for the subtractor highlighted in pink in Fig. 6.1a, ensuring that the SCDT histogram of the proposed two-stage interpolation TDC starts at 1. Fig. 6.5a and Fig. 6.5b show the SCDT histograms of the TDL-TDC and the designed two-stage interpolation TDC, respectively. The pattern highlighted in blue in Fig. 6.5b (which is inverted from the SCDT histogram of the TDL-TDC) appears periodically in the SCDT histogram of the proposed TDC, matching the expectation for this design.

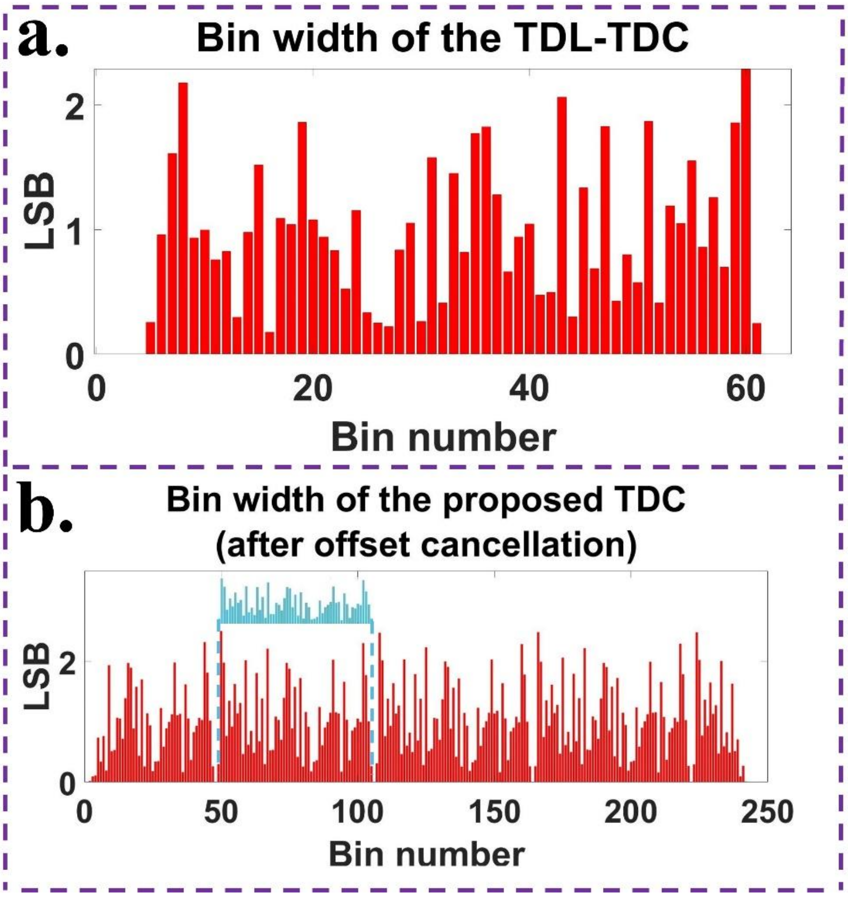


Figure 6.5 SCDT histograms of (a) the TDL-TDC and (b) the proposed two-stage interpolation TDC in the Virtex-7 FPGA.

6.3 Experimental results

The designed 16-channel two-stage interpolation TDC was implemented and evaluated in KCU-105 [159] (shown in Fig. 4.18) and NetFPGA-SUME [160] (shown in Fig. 4.19) evaluation boards, respectively. The experimental setup was similar to that described in Chapter 4, where the coarse-counting clocks were from low-jitter crystal oscillators on boards, but here, both clocks were configured to 400 MHz. Moreover, an uncorrelated 3.777777 MHz hit from SRS CG-635 [161] was used as a random input for SCDTs, and on-chip delay macros were employed to generate controllable delays (relative to coarse-counting clocks) for TI tests. Notably, the clocks for TDL-TDCs in this design are from designed GCOs rather than from the coarse-counting clock. Hence, timing constraints were conducted referring to GCOs' frequencies measured by Teledyne LeCroy 640Zi [173].

6.3.1 Resolution and Linearity

The oscillation periods of GCOs in both evaluation boards were measured to calculate the resolutions of VGCO-TDCs. Moreover, the resolutions, linearity and equivalent bin width of the designed two-stage interpolation TDCs were also used to evaluate the proposed TDC's performance.

Table 6-1 presents the oscillation periods of GCOs and resolutions of VCGO-TDCs. As shown in Table 6-1, the oscillation periods are designed from 7 ns to 11 ns for TDL-TDCs' fast timing closure in the implemented devices. In the Kintex-UltraScale FPGA, the resolutions of VGCO-TDCs range between 750 ps and 910 ps, with an average resolution of 831 ps. Differently, in the Vitrtex-7 FPGA, VGCO-TDCs offer resolutions between 460 ps and 800 ps, with an average resolution of 579 ps. GCOs in the Virtex-7 FGPA have faster oscillation frequencies than GCOs in the Kintex-UltraScale FPGA, allowing the VGCO-TDC in the Virtex-7 FPGA to achieve a similar dead time (for a 5-ns maximum measurement range, $\tau+T$) even with a finer resolution. Table 6-2 summarises the resolution, linearity and equivalent bin width of the designed two-stage interpolation TDCs in both evaluation boards. TDCs in the Kintex-UltraScale FPGA have resolutions between 4.50 ps and 4.66 ps, with an average resolution of 4.57 ps, showing good uniformity between channels. And the average DNL$_{\text{pk-pk}}$ and INL$_{\text{pk-pk}}$ are 4.36 LSB and 18.26 LSB, respectively, with a 5.72 LSB maximum DNL$_{\text{pk-pk}}$ and a 23.66 LSB maximum INL$_{\text{pk-pk}}$. Besides, the ω_{eq} varies from 9.16 ps to 10.68 ps, averaging at 9.93 ps, simultaneously determined by the resolution and linearity. For TDCs in the Virtex-7 FPGA, the resolutions range from 9.65 ps to 10.29 ps, with an average resolution of 10.05 ps. And the average DNL$_{\text{pk-pk}}$ and INL$_{\text{pk-pk}}$ are 2.85 LSB and 13.61 LSB, respectively, with a 4.29 LSB maximum DNL$_{\text{pk-pk}}$ and a 19.71 LSB maximum INL$_{\text{pk-pk}}$. Moreover, the ω_{eq} fluctuates between 14.26 ps and 19.54 ps, averaging at 15.97 ps. Compared with TDCs in the Viretex-7 FPGA, TDCs in the Kintex-UltraScale FPGA have an average resolution enhanced by more than 2-fold, from 10.05 ps to 4.57 ps. However, the ω_{eq} only improves by 1.6-fold (from 15.97 ps to 9.93 ps), due to worse linearity.

Table 6-1 Periods of GCOs and resolutions of VGCO-TDCs

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Unit (× 10 ps)																
Kintex-UltraScale																
Peri. (Slow GCO)	1092	1092	1090	1091	1101	1097	1099	1098	1107	1104	1108	1105	1100	1099	1094	1095
Peri. (Slow GCO)	1010	1008	1009	1002	1030	1020	1009	1007	1032	1021	1018	1020	1024	1008	1018	1012
Reso.	82	84	81	89	81	77	90	91	75	83	90	85	76	91	76	83
Virtex-7																
Peri. (Slow GCO)	788	796	794	798	788	785	793	790	786	785	793	788	788	784	788	785
Peri. (Slow GCO)	726	736	734	738	725	736	730	732	731	725	723	731	734	738	740	737
Reso.	62	60	60	60	63	49	63	58	55	60	80	57	54	46	48	48

Table 6-2 Performance of the proposed two-stage interpolation TDC

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***Ave.***
Kintex-UltraScale																	
Reso. (ps)	4.54	4.51	4.50	4.50	4.62	4.54	4.63	4.52	4.65	4.55	4.55	4.63	4.53	4.63	4.66	4.55	4.57
DNL$_{pk\text{-}pk}$ (LSB)	4.29	5.32	4.14	4.27	3.91	3.86	4.25	5.30	4.22	5.72	3.80	4.35	4.58	4.12	3.77	3.79	4.36
INL$_{pk\text{-}pk}$ (LSB)	18.23	21.61	18.24	19.64	20.32	19.12	18.05	18.14	15.95	23.66	16.91	16.34	19.29	16.11	12.88	17.70	18.26
ω_{eq} ***(ps)***	9.95	10.68	9.16	9.66	9.72	9.86	9.80	10.65	9.82	10.44	9.63	9.82	9.32	10.23	10.17	9.91	9.93
σ_{valid} ***(ps)***	23.03	23.11	22.71	22.94	23.28	22.92	22.91	22.57	23.55	22.89	22.31	22.74	23.32	22.20	22.75	22.80	22.88
Virtex-7																	
Reso. (ps)	10.25	10.25	10.12	10.20	10.00	9.96	9.73	10.12	9.65	10.20	10.29	9.84	10.16	10.08	10.25	9.68	10.05
DNL$_{pk\text{-}pk}$ (LSB)	2.46	4.29	3.29	2.56	2.24	3.67	2.60	2.97	2.36	4.22	2.29	2.25	2.43	3.01	2.67	2.31	2.85
INL$_{pk\text{-}pk}$ (LSB)	12.42	15.25	14.10	12.90	16.71	14.42	14.21	16.79	19.71	11.04	9.46	14.00	12.24	13.56	11.07	9.91	13.61
ω_{eq} ***(ps)***	15.39	19.54	16.76	15.69	14.98	17.19	15.19	16.27	14.69	19.28	14.62	14.62	15.05	16.22	15.72	14.26	15.97
σ_{valid} ***(ps)***	19.76	20.58	20.48	20.69	19.51	21.63	19.56	20.29	18.80	19.41	18.93	19.30	19.43	19.72	19.46	19.34	19.81

6.3.2 Time interval test

On-chip programmable delay macros (IDELAY3 [18] in the Kintex-UltraScale FPGA and IDELAY2 [17] in the Virtex-7 FPGA) were used to delay a 5 MHz clock (synchronised with the coarse-counting clock) as the input for TI tests. Moreover, the bin-by-bin calibration method [68] was also used to minimise the impacts of quantization errors and INL on measurements.

Fig. 6.6 shows the RMS precisions of TDCs implemented in both FPGAs. In the Kintex-UltraScale FPGA, 60 delay taps are used to cover a 2.5 ns period of the coarse-counting clock (@ 400MHz). However, only 32 delay taps are required for the same period in the Virtex-7 FPGA due to a worse resolution of IDELAY2 compared with IDELAY3. For TDCs in both FPGAs, most measured groups containing different TIs (highlighted in red in Fig. 6.6) have a sharp change, which is caused by spanning coarse-counting clock cycles. However, the exact points of the sharp changes differ due to different path delays of the input signal, and various path delays are caused by different placement and routing strategies. Generally, there is a deteriorating trend of RMS precisions when increasing measured TIs, caused by the accumulation of jitters from VGCO-TDCs. As shown in Fig. 6.1a, the output from the slow GCO drives the oscillation counter, and longer TIs require more oscillation cycles, leading to increased accumulation of jitters. The accumulated jitters then deteriorate the RMS precisions of the designed two-stage interpolation TDC. Moreover, the period of the coarse-counting clock also influences VGCO-TDCs' accumulated jitters since it determines the measurement range of VGCO-TDCs. Meanwhile, it is worth noting that the trend of the proposed TDC's RMS precision slightly differs from that of the measured TI, especially in the Kintex-UltraScale FPGA. The reason behind this phenomenon is that measurement uncertainty is not sourced from VGCO-TDCs only. Measurement uncertainty from the TDL-TDC, unrelated to the measured TI, also contributes to RMS precisions.

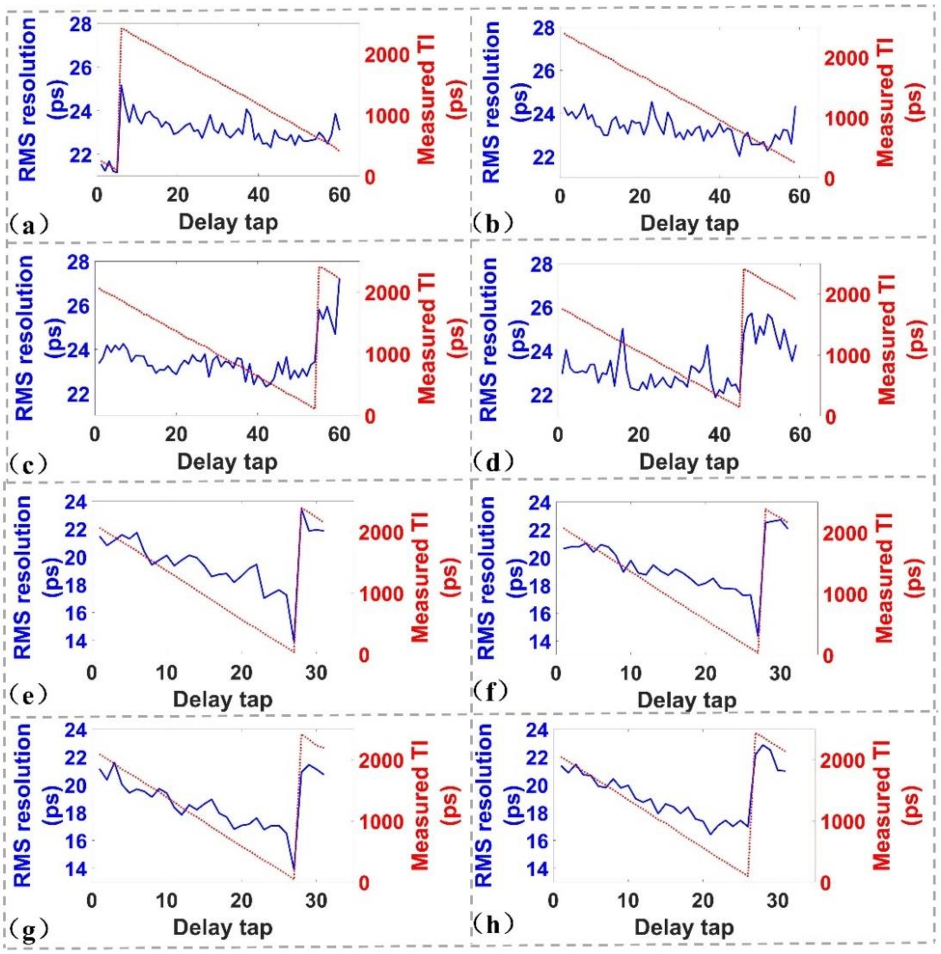


Figure 6.6 RMS precisions and measured TIs of the (a) channel-1, (b) channel-5, (c) channel-9 and (d) channel-13 in the Kintex-UltraScale FPGA, and (e) channel-1, (f) channel-5, (g) channel-9 and (h) channel-13 in the Virtex-7 FPGA.

Fig. 6.6 shows RMS precisions for different intervals within one coarse-clock period. Moreover, RMS precisions for the entire coarse-clock period are also characterised by the valid RMS precision referring to Eq. (4.9). They are summarised in Table 6-2.

6.3.3 Hardware utilisation and constraint

Table 6-3 Hardware utilisation

		LUT	DFF	CARRY[1]	CLB/Slice[2]
UltraScale	Available	242400	484800	30300	30300
	1-ch	402	544	52	155
	16-ch	6431	8704	832	2495
	Para. Core	614	419	1	165
Virtex-7	Available	433200	866400	108300	108300
	1-ch	257	360	58	177
	16-ch	4113	5760	928	2695
	Para. Core	574	396	5	248

[1] CARRY8s in Kintex-UltraScale FPGA and CARRY4s in Virtex-7 FPGA; [2] CLB in Kitex-UltraScale FPGA and slice in Virtex-7 FPGA.

The designed TDC was implemented in both FPGAs, with a detailed summary of hardware utilisation presented in Table 6-3. For TDCs in the Kintex-UltraScale FPGA, hardware utilisation per channel includes 402 LUTs, 544 DFFs, 1.5 BRAMs for Histo. BRAM, and 1.5 BRAMs for Asyn. Output BRAM. Moreover, 29 CARRY8s are used for TDL construction, and 23 CARRY8s are used for arithmetic operations such as accumulation, multiplication, and subtraction. In addition to the aforementioned hardware utilisation, a dedicated Para. Core, which calculates and configures parameters for all 16-channel TDCs, also consumes 614 LUTs and 419 DFFs in the Kintex-UltraScale FPGA. However, TDCs in the Virtex-7 FPGA have less hardware utilisation due to VGCO-TDCs' better resolutions and worse TDL propagation delays. For TDCs in the Virtex-7 FPGA, each channel consumes 257 LUTs and 360 DFFs. Besides, each channel also uses 1 BRAM for Histo. BRAM and 1 BRAM for Asyn. Output BRAM, and 20 CARRY4s are used for TDLs and 38 CARRY4s as arithmetic units. Unlike the Para. Core in the Kintex-UltraScale FPGA, the Para. Core in the Virtex-7 FPGA only consumes 574 LUTs and 396 DFFs due to the shorter bin width of parameters. The hardware utilisation indicates the proposed design is more hardware-efficient than conventional TDL-TDCs [83] and has similar hardware utilisation compared with the VRO-TDC presented in Ref. [99] (a detailed comparison is shown in Table 6-4).

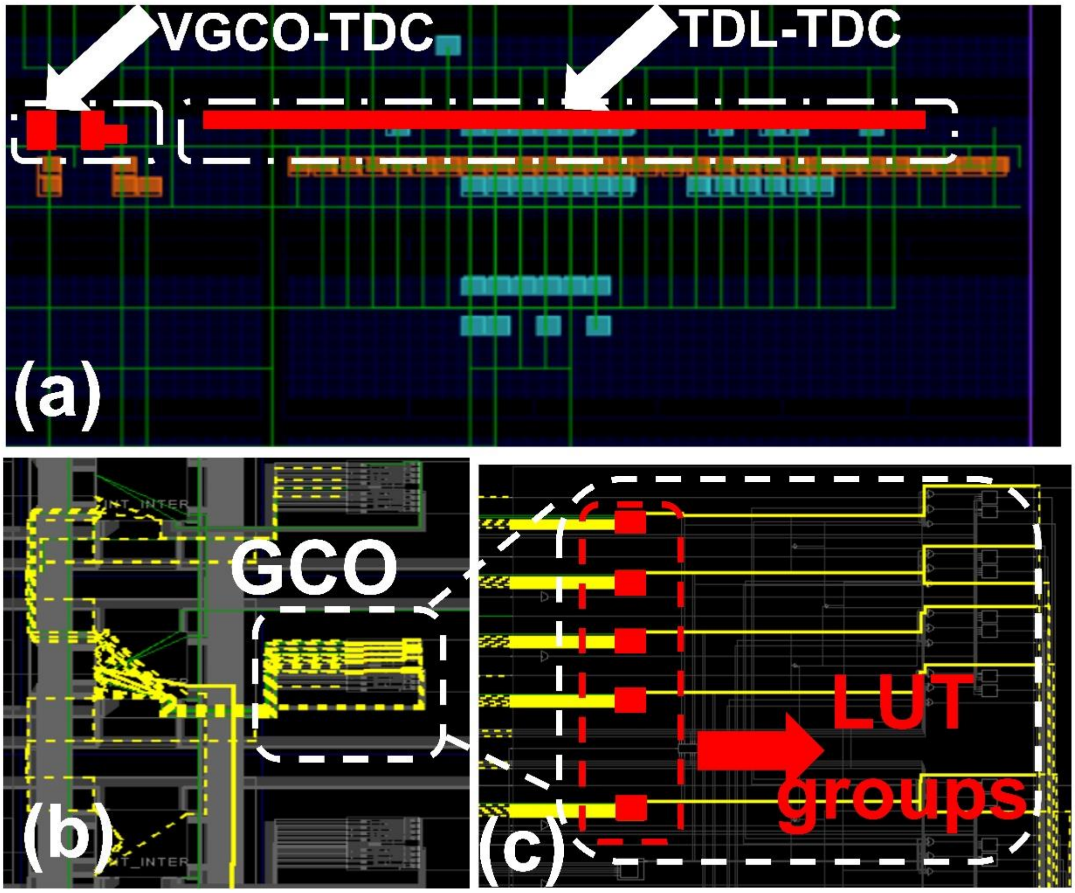


Figure 6.7 (a) Implementation layouts of the two-stage interpolation TDC, (b) routing details of the GCO, and (c) implementation layouts of the GCO in the UltraScale-Kintex FPGA.

Fig. 6.7a shows the implementation layout of the proposed TDC in the Kintex-UltraScale FPGA. The VGCO-TDC is placed close to the corresponding TDL-TDC to minimise jitters and skews induced by inner connections. Besides, as shown in Fig. 6.7b and Fig. 6.7c, the GCO is manually routed (manually constrained routes are highlighted as yellow dotted lines in Fig. 6.7b) and placed (manually placed LUTs are highlighted in red in Fig. 6.7c) to ensure uniformity between channels. Notably, the timing constraints of the TDL-TDC in this design differ from those of previous TDL-TDCs since the designed TDL-TDCs' sampling and encoding clocks are from GCOs rather than MMCMs/PLLs. Therefore, the command "create clock -period" is used to claim the GCO's output as a clock and specify its period. Meanwhile, the command "set_clock_groups -asynchronous" is also used to define asynchronous clock groups, avoiding unnecessary timing checks between different clock domains.

6.4 Comparison

Table 6-4 presents a summary of recently published TDCs. To evaluate the maximum dead time in the worst-case scenario for all 16 channels, the maximum oscillation number and maximum oscillation period of the proposed TDC are used. Therefore, the dead time of this design can be calculated as follows:

$$T_{Dead} = (N_{Osci}^{max} + T_{code} + T_{calc.} + T_{his.} + T_{reset}) \times P_{max}, \tag{6.5}$$

where N_{Osci}^{max} is the maximum oscillation number of the slow GCO for the 5-ns maximum measurement range, T_{code}, $T_{cacl.}$, $T_{his.}$, and T_{reset} are the required clock cycles for the encoder, result calculation, histogram and resetting the TDC (with 2, 3,1 and 1 clocks cycles, respectively), and P_{max} is the maximum oscillation period for the slow GCOs.

As shown in Table 6-4, TDL-TDCs are the most used architecture for FPGA-TDCs, and VRO-TDCs are also well-developed. However, the two-stage interpolation architecture combining a VGCO-TDC and a TDL-TDC is proposed in this chapter, achieving a better dead time and a finer resolution than conventional VRO-TDCs, and lower hardware utilisation compared with conventional TDL-TDCs.

Typically, TDL-TDCs have a one-cycle or two-cycle dead time, benefitting from pipeline sampling and encoding techniques, and the VRO-TDCs' dead time is much longer due to the measuring principle (the measurement is conducted by the fast oscillator "chasing" the slow oscillator). However, the proposed two-stage interpolation architecture significantly reduces the oscillation number, even with a finer resolution, which further reduces the dead time. For example, TDCs' dead time in Ref. [172] and [99] is 400 ns and 602 ns. However, the dead time of this design is 155 ns (in the Kintex-UltraScale FPGA) and 144 ns (in the Virtex-7 FPGA), respectively.

Table 6-4 Comparisons of recently published TDL-TDCs and VRO-TDCs

Ref-year	*Methods*	*Devi. Proc. (nm)*	*LSB (ps)*	ω_{eq} *(ps)*	*Prec. (ps)*	*DNL (LSB)*	*INL (LSB)*	*Dead Time (ns)*	*LUT*	*DFF*	*CARRY*	*CLB /Slice*
[83]-19	Sub-TDL, Bin-width compensation and calibration.	20	5.02	5.03	7.81[1]	[-0.12,0.11]	[-0.18,0.46]	NS[2]	703	1195	80[3]	NS[2]
		28	10.54	10.55	14.59[1]	[-0.05,0.08]	[-0.09,0.11]	NS[2]	1145	1916	NS[2]	712[4]
[174]-22	Dual-mode encoder.	28	22.1	NS[2]	22.35[1]	[-0.71,1.05]	[0.85,0.86]	4	228	678	48[5]	NS[2]
[175]-22	Wave union A, bin merging.	28	10	NS[2]	17	[-0.13,0.15]	[-2.26,3.54]	NS[2]	1136[6]	2716[6]	NS[2]	NS[2]
[116]-23	Folding-TDC.	28	4.4	NS[2]	4.6	NS[2]	NS[2]	4.4	339	740	NS[2]	NS[2]
			6.5	NS[2]	6.4	NS[2]	NS[2]	4.4	231	352	NS[2]	NS[2]
VRO-TDCs												
[172]-17	Period difference recording.	65	[23,37]	NS[2]	[32,39]	[-0.4,0.4]	[-0.7,0.7]	400	104	319	NS[2]	NS[2]
[99]-20	Bidirectional-Operating.	65	24.5	NS[2]	28	[-0.20,0.25]	[0.03,0.82]	602	172	986	NS[2]	NS[2]
Other-TDCs												
This work	**Two-stage interpolation.**	**20**	**4.57**	**9.93**	**22.88[1,7]**	**[-1,3.14]** **4.36[8]**	**[-8.81,9.44]** **18.26[9]**	**155**	**402** **440[10]**	**544** **570[10]**	**52[3]**	**155[11]**
		28	**10.05**	**15.97**	**19.81[1,7]**	**[-1,1.46]** **2.85[8]**	**[-6.01,6.40]** **13.61[9]**	**144**	**257** **293[10]**	**360** **385[10]**	**58[5]**	**177[4]**

[1] Single-shot precision; [2] N/S = Not specified; [3] CARRY8; [4] Slice; [5] CARRY 4; [6] Average value for 19 channels; [7] Valid precision; [8] Average $\mathrm{DNL}_{\text{pk-pk}}$ for 16 channels; [9] Average $\mathrm{INL}_{\text{pk-pk}}$ for 16 channels; [10] Each channel's average hardware utilisation with the Para. Core; [11] CLB.

Moreover, compared with conventional VRO-TDCs, the reduced oscillation number also enhances precision due to less jitter accumulation. For example, the proposed TDC has similar precision compared to those in Ref. [172] and [99], although this design has a much better resolution. Simultaneously, the designed TDC has a similar hardware utilisation compared with the design in Ref. [99]. However, this design is less hardware-efficient than the design in Ref. [172] due to the on-chip calculation and histogram.

TDL-TDCs in Ref. [83] and Ref. [175] have resolutions similar to that of the proposed TDCs. However, in the proposed two-stage interpolation architecture, the TDL-TDC only needs to cover the resolution of the VGRO-TDC, indicating the proposed TDC is more hardware-efficient than conventional TDL-TDCs. The designs in Ref. [174] and Ref. [116] have similar hardware utilisation compared with this design. The TDL-TDC in Ref. [174] consumes 228 LUTs and 678 DFFs, similar to the design in the Kintex-UltraScale FPGA. However, the designed TDC in the Kintex-UltraScale FPGA achieves a significantly finer resolution, improving it from 20 ps to 5 ps. Compared with the design in Ref. [116], the design in the Virtex-7 FPGA has similar hardware utilisation but underperforms in terms of resolution and precision. Here, the advantage of the proposed architecture is the need for fewer CARRY4s to construct the TDL and a lower-frequency clock required for TDL-TDC's sampling and encoding. In Ref. [116], over 200 taps (50 CARRY4s) cover the 554 MHz sampling clock. However, in this design, only 80 taps (20 CARRY4s) are required to construct the TDL to cover the resolution of the VGRO-TDC rather than the entire period of the coarse clock. Compared with conventional TDL-TDCs, this architecture allows for TDL's length to be independent of the coarse-counting clock, further reducing the difficulty of timing closure (a high-frequency sampling clock of the TDL-TDC is preferred to reduce the length of the TDL in conventional TDL-TDCs). Moreover, although 257 LUTs and 360 DFFs per channel are used for the TDC in the Virtex-7 FPGA (402 LUTs and 544 DFFs per channel are used for the TDC in the Kintex-UltraScale FPGA), only 86 LUTs and 116 DFFs are used for the VGCO-TDC and TDL-TDC (only 213 LUTs and 284 DFFs are used for the VGCO-TDC and TDL-TDC in the Kintex-UltraScale FPGA), indicating this design can be more compact.

6.5 Conclusion

This chapter uses GCOs to replace CARRY4s/CARRY8s to construct oscillators for Vernier-TDCs and proposes the two-stage interpolation architecture. The TDL-TDC in the proposed architecture only needs to cover the resolution of the VGCO-TDC, reducing hardware utilisation. Moreover, the length of the TDL is not related to the frequency of the TDL-TDC's sampling clock, simplifying timing closure (typically, a high-frequency TDL-TDC sampling clock is preferred to minimise the length of the TDL). Compared with previous VRO-TDCs, the proposed TDCs improve the dead time and precision even with a finer resolution by reducing oscillation numbers.

This chapter also implemented the proposed 16-channel TDC in Kintex-UltraScale and Virtex-7 FPGAs to evaluate the design. Experimental results indicate that the proposed TDC is hardware-efficient compared with VRO-TDCs and TDL-TDCs and has competitive performance compared with VRO-TDCs, making it appropriate for multi-channel and low-conversion-rate applications such as FLIM, although the precision still needs further improvement compared with TDL-TDCs. Besides, the multiphase clock method [120] for the VGCO-TDC and the WU method [115] for TDL-TDC can be implemented in future work to improve the precision.

Chapter 7 Conclusion

7.1 Summary of thesis achievements

FPGA-TDCs are mainstream for rapid prototype verification, benefiting from the short development cycle, low development cost and high flexibility. However, FPGA-TDCs' performance in terms of resolution and linearity is limited by FPGAs' inherent delays and imperfect manufacturing. The increasing demand for multi-channels also requires TDC to have low hardware utilisation. To address these challenges, four architectures were proposed in this thesis, aiming to enhance TDC's performance and hardware utilisation efficiency.

Chapter 2 began by introducing the parameters for TDC's performance. It then delved into architectures that can be applicable to both ASICs and FPGAs, as well as those unique to each platform. Moreover, recently reported FPGA-TDCs were also summarised and compared in this chapter. Generally, ASIC-TDCs perform better than FPGA-TDCs due to the customised cells, manual placement, and careful routing. However, FPGA-TDCs are more suitable for rapid prototype verification. Through comparisons between different FPGA-TDCs, Chapter 2 proposed potential enhancements that guide designs in the following chapters.

Aiming for high-linearity, a weighted histogram calibration method and an automatic calibration architecture were proposed in Chapter 3. The calibration method was executed in PS of the Zynq-7000 SoC, while the 16-channel TDCs were implemented in PL (equivalent to a 28 nm Artix-7 FPGA). By using PS to calibrate TDCs in PL, linearity was enhanced without any manual intervention. Combined with the WU method, the tuned-TDL method, and the sub-TDL method, the proposed TDC can offer a 9.83 ps resolution, with average $\text{DNL}_{\text{pk-pk}}$ of 0.27 LSB and average $\text{INL}_{\text{pk-pk}}$ of 0.67 LSB. This design has potential for commercial FPGA-TDC applications due to its on-chip automatic calibration and competitive resolution.

Although the design in Chapter 3 achieved automatic calibration, it has a dependency on PS. Therefore, Chapter 4 further improved it and proposed a resolution-configurable and low hardware utilisation TDC. The design in Chapter 4 utilised the GCO as the delay cell and integrated a sampling matrix to enhance the resolution. Moreover, by hardware implementation of the VBCM, the proposed TDC achieved automatic calibration and online resolution configuration. To demonstrate the device-independency of the proposed architecture, the 16-channel TDC system was implemented and evaluated in Zynq UltrsScale+, Kintex-UltraScale and Virtex-7 FPGAs, respectively. The UltraScale+ version achieved the best LSB of 20.97 ps with 0.09 LSB average $\mathrm{DNL}_{\text{pk-pk}}$. The UltraScale and Virtex-7 versions achieved the best resolutions of 36.01 ps with 0.10 LSB average $\mathrm{DNL}_{\text{pk-pk}}$ and 34.84 ps with 0.08 LSB average $\mathrm{DNL}_{\text{pk-pk}}$, respectively. Due to its high linearity, low hardware utilisation and reconfigurable resolution, this design is well-suited for integration into applications such as LiDAR and FLIM.

In addition to high linearity, TDCs also pursue high resolutions. Therefore, Chapter 5 proposed a 4-edge WU TDC for a sub-picosecond resolution. This design combined the WU and dual-sampling methods to enhance the resolution and utilised the sub-TDL to remove bubbles before encoding. Simultaneously, a bidirectional encoder and a manually placed and routed WU launcher were also designed for 4-edge transition real-time encoding. Implemented in the 16 nm Zynq UltraScale+ device, the designed TDC can achieve 0.4 ps resolution and a less than 9 ps RMS precision with a measurement range from 0 ns to 100 ns. This design achieved the best resolution among all previously reported WU-A FPGA-TDCs and can be used in high-time-resolution scientific instruments and high-energy physics. However, this design also suffers from significant hardware utilisation and is hence limited in multi-channel applications.

A clear trade-off between resolution and hardware utilisation was shown in the designs of Chapter 4 and Chapter 5. To achieve high resolutions and low hardware utilisation simultaneously, Chapter 6 proposed a two-stage interpolation TDC, combining a VGCO-TDC and a TDL-TDC. This architecture employed the LUT-based VGCO-TDC as the first-stage interpolation for fine-time measurement. The overtaking residue from the VGCO-TDC was then measured by the TDL-TDC. Therefore, the length of the delay line only needs to cover the resolution of the VGCO-TDC, which significantly reduces the hardware utilisation of the TDL-TDC. Simultaneously, the resolution of the proposed two-stage TDC was only determined by the TDL-TDC, achieving high hardware utilisation efficiency without resolution loss. The proposed architecture was implemented and evaluated in both Kintex-UltraScale and Virtex-7 FPGAs. The Kintex-UltraScale version achieved a resolution of 4.57 ps with the hardware utilisation of 440 LUTs and 570 DFFs, and the Virtex-7 version achieved a 10.05 ps resolution, consuming 293 LUTs and 385 DFFs.

The proposed four TDCs, including novel architectures and calibration methods, significantly enhance resolution, linearity and hardware utilisation efficiency. These innovations provide valuable insights for researchers and engineers targeting TDC designs. Simultaneously, these designs also have outstanding performance, indicating that they can be integrated as a TI-measurement component into applications such as LiDAR, FLIM, PET, etc.

7.2 Future work

Diving deeper into FPGA-TDCs, there are three primary points that could be further explored based on current studies: 1) packaging the designed automatic calibration TDCs, such as TDCs presented in Chapter 3 and Chapter 4, into intellectual property (IP) cores to achieve plug and play when prototyping; 2) outputting lossless data from multi-channel TDCs even when TDCs have a high-sampling rate; and 3) explore new resources within FPGAs, beyond the traditional use of carry elements and DSPs, to achieve high-resolution FPGA-TDCs.

Previously reported high-linearity FPGA-TDCs require manual calibration channel-by-channel and chip-by-chip [83], [120], [128]. This procedure is time-consuming and user-unfriendly, particularly for those unfamiliar with FPGA-TDCs. To address this, this thesis proposed the automatic calibration FPGA-TDCs. However, these designs also require the instantiation of several modules, which is complex. Hence, packaging the designs in Chapter 3 and Chapter 4 into IP cores can facilitate prototyping integration. Notably, these two designs also contain some constraints rather than the Verilog hardware description language [176] only. Constraints may differ across different platforms. Besides, scalability is also important for transforming fixed designs (such as those with a fixed number of channels and predefined functions) into configurable IP cores to ensure they can meet various requirements.

Photon-electronics sensors, such as SPADs, have a rising trend of multiple pixels. Meanwhile, applications including LiDAR and FLIM also benefit from multi-pixel imaging. Hence, multi-channel designs for FPGA-TDCs gain increasing attention. For example, designs in Ref. [83] and Ref. [165] achieved 128 channels. As the thesis introduced, hardware utilisation is a crucial parameter for multi-channel TDCs. Furthermore, the data output strategy and the output bandwidth are also significant concerns for multi-channel TDCs since they can determine the systems' ability to acquire and transmit data from multiple channels without bottlenecking. Therefore, the primary challenge lies in achieving lossless output with limited bandwidth [such as universal serial bus (USB) 3.0, which has transmission speeds up to 5Gbit/s]. Techniques such as advanced data compression and efficient encoding schemes could be feasible solutions. On the other hand, using application-specific algorithms to process the TDCs' output on-chip before transmitting may also be effective.

The final but equally important point for future research is to explore other resources within FPGAs for high-resolution TI measurements. This thesis utilises carry elements, LUTs and routing resources as delay cells for fine-time interpolation. Moreover, DSPs were also used for fine-time measurement in Ref. [105], Ref. [106] and Ref. [107]. However, other resources within Xilinx FPGAs may also have the potential for high-resolution TI measurements. For example, the hard macro "ISERDESE2", which has series input and parallel outputs, operates similarly to TDLs. Hence, it has the potential to be used for fine-time measurements. Besides, other hard macros, which have better timing performance than programable logic (such as LUTs and CARRY 4/8s), may also have the capability to measure a TI with high resolutions.

Reference

[1] 'International System of Units', International System of Units. [Online]. Available: https://en.wikipedia.org/wiki/International_System_of_Units

[2] United Sates Navy, 'Definitions of Systems of Time', https://www.cnmoc.usff.navy.mil/Our-Commands/United-States-Naval-Observatory/Precise-Time-Department/The-USNO-Master-Clock/Definitions-of-Systems-of-Time/. Accessed: Sep. 11, 2024. [Online]. Available: https://www.cnmoc.usff.navy.mil/Our-Commands/United-States-Naval-Observatory/Precise-Time-Department/The-USNO-Master-Clock/Definitions-of-Systems-of-Time/

[3] S. Tancock, 'The Efficient Design of Time-to-Digital Converters', PhD Thesis, University of Bristol, 2021.

[4] R. Scott, W. Jiang, and M. J. Deen, 'CMOS Time-to-Digital Converters for Biomedical Imaging Applications', IEEE Rev. Biomed. Eng., vol. 16, pp. 627–652, 2023, doi: 10.1109/RBME.2021.3092197.

[5] F. Bouyjou et al., 'HGCROC3: the front-end readout ASIC for the CMS High Granularity Calorimeter', J. Inst., vol. 17, no. 03, p. C03015, Mar. 2022, doi: 10.1088/1748-0221/17/03/C03015.

[6] S. Tancock, E. Arabul, and N. Dahnoun, 'A Review of New Time-to-Digital Conversion Techniques', IEEE Trans. Instrum. Meas., vol. 68, no. 10, pp. 3406–3417, Oct. 2019, doi: 10.1109/TIM.2019.2936717.

[7] R. Nutt, 'Digital Time Intervalometer', Review of Scientific Instruments, vol. 39, no. 9, pp. 1342–1345, Sep. 1968, doi: 10.1063/1.1683667.

[8] J. Kalisz, 'Review of methods for time interval measurements with picosecond resolution', Metrologia, vol. 41, no. 1, pp. 17–32, Feb. 2004, doi: 10.1088/0026-1394/41/1/004.

[9] G. Acconcia, F. Malanga, S. Farina, M. Ghioni, and I. Rech, 'A 1.9 ps-rms Precision Time-to-Amplitude Converter With 782 fs LSB and 0.79%-rms DNL', IEEE Trans. Instrum. Meas., vol. 72, pp. 1–11, 2023, doi: 10.1109/TIM.2023.3271755.

[10] J. Kalisz, R. Pelka, and A. Poniecki, 'Precision time counter for laser ranging to satellites', Review of Scientific Instruments, vol. 65, no. 3, pp. 736–741, Mar. 1994, doi: 10.1063/1.1145094.

[11] J. Kim, Y.-H. Kim, K. Kim, W. Yu, and S. Cho, 'A Hybrid-Domain Two-Step Time-to-Digital Converter Using a Switch-Based Time-to-Voltage Converter and SAR ADC', IEEE Trans. Circuits Syst. II, vol. 62, no. 7, pp. 631–635, Jul. 2015, doi: 10.1109/TCSII.2015.2415631.

[12] J. Szyduczyński, D. Kościelnik, and M. Miśkowicz, 'Time-to-digital conversion techniques: a survey of recent developments', Measurement, vol. 214, p. 112762, Jun. 2023, doi: 10.1016/j.measurement.2023.112762.

[13] AMD Xilinx, 'AMD Xilinx'. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga.html

[14] Intel Altera, 'Intel Altera'. [Online]. Available:

https://www.intel.com/content/www/us/en/products/programmable.html

[15] AMD Xilinx, ‘7 Series FPGAs Configurable Logic Block User Guide (UG474)’. 2014. [Online]. Available: https://docs.amd.com/v/u/en-US/ug474_7Series_CLB

[16] AMD Xilinx, ‘UltraScale Architecture Configurable Logic Block User Guide (UG574)’. 2017. [Online]. Available: https://docs.amd.com/v/u/en-US/ug574-ultrascale-clb

[17] AMD Xilinx, ‘Vivado Design Suite 7 Series FPGA and Zynq-7000 SoC Libraries Guide’. 2023. [Online]. Available: https://docs.amd.com/r/en-US/ug953-vivado-7series-libraries

[18] AMD Xilinx, ‘UltraScale Architecture Libraries Guide (UG974)’. 2018. [Online]. Available: https://docs.xilinx.com/r/en-US/ug974-vivado-ultrascale-libraries/Introduction

[19] W. Xie, ‘Strategies Towards High Performance (High-Resolution/Linearity) Time-to-digital Converters on Field-programmable Gate Arrays’.

[20] M. I. Masud, ‘FPGA Routing Structures: A Novel Switch Block and Depopulated Interconnect Matrix Architectures’, Bachelor Thesis, Dalhousie University, 1998.

[21] D. Gomez-Prado and M. Ciesielski, ‘A Tutorial on FPGA Routing’, 2006.

[22] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, ‘Moving event detection from LiDAR point streams’, Nat Commun, vol. 15, no. 1, p. 345, Jan. 2024, doi: 10.1038/s41467-023-44554-8.

[23] J. Wang, M. Xu, G. Zhao, and Z. Chen, ‘3-D LiDAR Localization Based on Novel Nonlinear Optimization Method for Autonomous Ground Robot’, IEEE Trans. Ind. Electron., vol. 71, no. 3, pp. 2758–2768, Mar. 2024, doi: 10.1109/TIE.2023.3270523.

[24] X. Sun, M. Wang, J. Du, Y. Sun, S. S. Cheng, and W. Xie, ‘A Task-Driven Scene-Aware LiDAR Point Cloud Coding Framework for Autonomous Vehicles’, IEEE Trans. Ind. Inf., vol. 19, no. 8, pp. 8731–8742, Aug. 2023, doi: 10.1109/TII.2022.3221222.

[25] H. Wang, Y. Yin, Q. Jing, Z. Cao, Z. Shao, and D. Guo, ‘Berthing assistance system for autonomous surface vehicles based on 3D LiDAR’, Ocean Engineering, vol. 291, p. 116444, Jan. 2024, doi: 10.1016/j.oceaneng.2023.116444.

[26] M. Perenzoni, D. Perenzoni, and D. Stoppa, ‘A 64 $\times$ 64-Pixels Digital Silicon Photomultiplier Direct TOF Sensor With 100-MPhotons/s/pixel Background Rejection and Imaging/Altimeter Mode With 0.14% Precision Up To 6 km for Spacecraft Navigation and Landing’, IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 151–160, Jan. 2017, doi: 10.1109/JSSC.2016.2623635.

[27] S. Oehmcke et al., ‘Deep point cloud regression for above-ground forest biomass estimation from airborne LiDAR’, Remote Sensing of Environment, vol. 302, p. 113968, Mar. 2024, doi: 10.1016/j.rse.2023.113968.

[28] X. Dai, W. Song, Y. Wang, Y. Xu, Y. Lou, and W. Tang, ‘LiDAR–Inertial Integration for Rail Vehicle Localization and Mapping in Tunnels’, IEEE Sensors J., vol. 23, no. 15, pp. 17426–17438, Aug. 2023, doi: 10.1109/JSEN.2023.3286437.

[29] J. Zhu, H. Li, and T. Zhang, ‘Camera, LiDAR, and IMU Based Multi-Sensor Fusion SLAM: A Survey’, Tsinghua Sci. Technol., vol. 29, no. 2, pp. 415–429, Apr. 2024, doi: 10.26599/TST.2023.9010010.

[30] W. Xie, Y. Wang, H. Chen, and D. D.-U. Li, ‘128-Channel High-Linearity

Resolution-Adjustable Time-to-Digital Converters for LiDAR Applications: Software Predictions and Hardware Implementations', IEEE Trans. Ind. Electron., vol. 69, no. 4, pp. 4264–4274, Apr. 2022, doi: 10.1109/TIE.2021.3076708.

[31] H. Chen, 'Design and implementation of high linearity FPGA-TDCs and an integrated large scale TCSPC system for time-resolved applications', PhD Thesis, University of Strathclyde, 2019.

[32] W. Becker, Ed., Advanced Time-Correlated Single Photon Counting Applications, vol. 111. in Springer Series in Chemical Physics, vol. 111. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-14929-5.

[33] S. A. Rodimova et al., 'Metabolic activity and intracellular pH in induced pluripotent stem cells differentiating in dermal and epidermal directions', Methods Appl. Fluoresc., vol. 7, no. 4, p. 044002, Sep. 2019, doi: 10.1088/2050-6120/ab3b3d.

[34] C. Biskup and G. Thomas, 'Fluorescence lifetime imaging of ions in biological tissues', in Fluorescence Lifetime Spectroscopy and Imaging, CRC Press, 2014, pp. 514–551.

[35] M. K. Kuimova, 'Mapping viscosity in cells using molecular rotors', Phys. Chem. Chem. Phys., vol. 14, no. 37, p. 12671, 2012, doi: 10.1039/c2cp41674c.

[36] A. Vyšniauskas and M. K. Kuimova, 'A twisted tale: measuring viscosity and temperature of microenvironments using molecular rotors', International Reviews in Physical Chemistry, vol. 37, no. 2, pp. 259–285, Apr. 2018, doi: 10.1080/0144235X.2018.1510461.

[37] K. Okabe, N. Inada, C. Gota, Y. Harada, T. Funatsu, and S. Uchiyama, 'Intracellular temperature mapping with a fluorescent polymeric thermometer and fluorescence lifetime imaging microscopy', Nat Commun, vol. 3, no. 1, p. 705, Feb. 2012, doi: 10.1038/ncomms1714.

[38] J. Jenkins, S. M. Borisov, D. B. Papkovsky, and R. I. Dmitriev, 'Sulforhodamine Nanothermometer for Multiparametric Fluorescence Lifetime Imaging Microscopy', Anal. Chem., vol. 88, no. 21, pp. 10566–10572, Nov. 2016, doi: 10.1021/acs.analchem.6b02675.

[39] X. Wang and O. S. Wolfbeis, 'Optical methods for sensing and imaging oxygen: materials, spectroscopies and applications', Chem. Soc. Rev., vol. 43, no. 10, pp. 3666–3761, 2014, doi: 10.1039/C4CS00039K.

[40] D. B. Papkovsky and R. I. Dmitriev, 'Imaging of oxygen and hypoxia in cell and tissue samples', Cell. Mol. Life Sci., vol. 75, no. 16, pp. 2963–2980, Aug. 2018, doi: 10.1007/s00018-018-2840-x.

[41] F. M. D. Rocca et al., 'Real-time fluorescence lifetime actuation for cell sorting using a CMOS SPAD silicon photomultiplier', Opt. Lett., vol. 41, no. 4, p. 673, Feb. 2016, doi: 10.1364/OL.41.000673.

[42] D. Xiao et al., 'Dynamic fluorescence lifetime sensing with CMOS single-photon avalanche diode arrays and deep learning processors', Biomed. Opt. Express, vol. 12, no. 6, p. 3450, Jun. 2021, doi: 10.1364/BOE.425663.

[43] J. Rong, A. Haider, T. E. Jeppesen, L. Josephson, and S. H. Liang, 'Radiochemistry for positron emission tomography', Nat Commun, vol. 14, no. 1, p. 3257, Jun. 2023, doi: 10.1038/s41467-023-36377-4.

[44] Jiang, Chalich, and Deen, ‘Sensors for Positron Emission Tomography Applications’, Sensors, vol. 19, no. 22, p. 5019, Nov. 2019, doi: 10.3390/s19225019.

[45] S. Surti, A. Kuhn, M. E. Werner, A. E. Perkins, J. Kolthammer, and J. S. Karp, ‘Performance of Philips Gemini TF PET/CT Scanner with Special Consideration for Its Time-of-Flight Imaging Capabilities’.

[46] L. Van Elmbt, S. Vandenberghe, S. Walrand, S. Pauwels, and F. Jamar, ‘Comparison of yttrium-90 quantitative imaging by TOF and non-TOF PET in a phantom of liver selective internal radiotherapy’, Phys. Med. Biol., vol. 56, no. 21, pp. 6759–6777, Nov. 2011, doi: 10.1088/0031-9155/56/21/001.

[47] C. S. Levin, S. H. Maramraju, M. M. Khalighi, T. W. Deller, G. Delso, and F. Jansen, ‘Design Features and Mutual Compatibility Studies of the Time-of-Flight PET Capable GE SIGNA PET/MR System’, IEEE Trans. Med. Imaging, vol. 35, no. 8, pp. 1907–1914, Aug. 2016, doi: 10.1109/TMI.2016.2537811.

[48] F. Monet and R. Kashyap, ‘On multiplexing in physical random number generation, and conserved total entropy content’, Sci Rep, vol. 13, no. 1, p. 7892, May 2023, doi: 10.1038/s41598-023-35130-7.

[49] A. Chan, M. Khalil, K. A. Shahriar, D. V. Plant, L. R. Chen, and R. Kuang, ‘Encryption in phase space for classical coherent optical communications’, Sci Rep, vol. 13, no. 1, p. 12965, Aug. 2023, doi: 10.1038/s41598-023-39621-5.

[50] A. Saini, A. Tsokanos, and R. Kirner, ‘Quantum Randomness in Cryptography—A Survey of Cryptosystems, RNG-Based Ciphers, and QRNGs’, Information, vol. 13, no. 8, p. 358, Jul. 2022, doi: 10.3390/info13080358.

[51] A. Khanmohammadi, R. Enne, M. Hofbauer, and H. Zimmermanna, ‘A Monolithic Silicon Quantum Random Number Generator Based on Measurement of Photon Detection Time’, IEEE Photonics J., vol. 7, no. 5, pp. 1–13, Oct. 2015, doi: 10.1109/JPHOT.2015.2479411.

[52] A. Tontini, L. Gasparini, N. Massari, and R. Passerone, ‘SPAD-Based Quantum Random Number Generator With an $N^{\rm{th}}$ -Order Rank Algorithm on FPGA’, IEEE Trans. Circuits Syst. II, vol. 66, no. 12, pp. 2067–2071, Dec. 2019, doi: 10.1109/TCSII.2019.2909013.

[53] M. Arredondo-Velázquez, L. Rebolledo-Herrera, B. De Celis Alonso, and E. Moreno-Berbosa, ‘Potential Use of State-of-the-Art TDCs for Particle Identification in Particle Physics Experiments’, IEEE Instrum. Meas. Mag., vol. 26, no. 6, pp. 13–20, Sep. 2023, doi: 10.1109/MIM.2023.10217031.

[54] G. A. Rinella et al., ‘TDCpix pixel detector ASIC with 100 ps time stamping’, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 1053, p. 168331, Aug. 2023, doi: 10.1016/j.nima.2023.168331.

[55] R. Wang et al., ‘Picosecond Timing Electronics for a DIRC-Like Detector at the Super Tau-Charm Facility’, IEEE Trans. Nucl. Sci., vol. 69, no. 11, pp. 2262–2270, Nov. 2022, doi: 10.1109/TNS.2022.3211863.

[56] M. Rudigier et al., ‘FATIMA — FAst TIMing Array for DESPEC at FAIR’, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 969, p. 163967, Jul. 2020, doi:

10.1016/j.nima.2020.163967.

[57] S. Liu, C. Feng, Q. An, Y. Heng, and S. Sun, ‘BES III Time-of-Flight Readout System’, IEEE Trans. Nucl. Sci., vol. 57, no. 2, pp. 419–427, Apr. 2010, doi: 10.1109/TNS.2009.2034520.

[58] W. Zhang et al., ‘A Low-Power Time-to-Digital Converter for the CMS Endcap Timing Layer (ETL) Upgrade’, IEEE Trans. Nucl. Sci., vol. 68, no. 8, pp. 1984–1992, Aug. 2021, doi: 10.1109/TNS.2021.3085564.

[59] Jinyuan Wu, Sten Hansen, and Zonghan Shi, ‘ADC and TDC implemented using FPGA’, in 2007 IEEE Nuclear Science Symposium Conference Record, Honolulu, HI, USA: IEEE, Oct. 2007, pp. 281–286. doi: 10.1109/NSSMIC.2007.4436331.

[60] H. Homulle, S. Visser, and E. Charbon, ‘A Cryogenic 1 GSa/s, Soft-Core FPGA ADC for Quantum Computing Applications’, IEEE Trans. Circuits Syst. I, vol. 63, no. 11, pp. 1854–1865, Nov. 2016, doi: 10.1109/TCSI.2016.2599927.

[61] L. Leuenberger, D. Amiet, T. Wei, and 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, Virtual Event USA: ACM, Feb. 2021, pp. 240–250. doi: 10.1145/3431920.3439287.

[62] Texas Instruments, ‘TDC7201 Time-to-Digital Converter forTime-of-Flight Applications in LIDAR, Range Finders, and ADAS’. 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/tdc7201.pdf?ts=1715892931659

[63] Analog Devices, ‘MAX35101 Time-to-Digital Converter with Analog Front-End’. 2015. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX35101.pdf

[64] Renesas, ‘ClockMatrix Time-to-Digital Converter’. 2021. [Online]. Available: https://www.renesas.com/us/en/document/apn/clockmatrix-time-digital-converter

[65] Swabian Instruments, ‘Time Tagger Series’. 2023. [Online]. Available: https://www.swabianinstruments.com/static/downloads/ TimeTaggerSeries.pdf

[66] R. Machado, J. Cabral, and F. S. Alves, ‘Recent Developments and Challenges in FPGA-Based Time-to-Digital Converters’, IEEE Trans. Instrum. Meas., vol. 68, no. 11, pp. 4205–4221, Nov. 2019, doi: 10.1109/TIM.2019.2938436.

[67] J. Doernberg, H.-S. Lee, and D. A. Hodges, ‘Full-speed testing of A/D converters’, IEEE J. Solid-State Circuits, vol. 19, no. 6, pp. 820–827, Dec. 1984, doi: 10.1109/JSSC.1984.1052232.

[68] J. Wu, ‘Several Key Issues on Implementing Delay Line Based TDCs Using FPGAs’, IEEE Trans. Nucl. Sci., vol. 57, no. 3, pp. 1543–1548, Jun. 2010, doi: 10.1109/TNS.2010.2045901.

[69] C. Zhang, S. Lindner, I. Antolovic, M. Wolf, and E. Charbon, ‘A CMOS SPAD Imager with Collision Detection and 128 Dynamically Reallocating TDCs for Single-Photon Counting and 3D Time-of-Flight Imaging’, Sensors, vol. 18, no. 11, p. 4016, Nov. 2018, doi: 10.3390/s18114016.

[70] C. Zhang, S. Lindner, I. M. Antolovic, J. Mata Pavia, M. Wolf, and E. Charbon, ‘A 30-frames/s, 252×144 SPAD Flash LiDAR With 1728 Dual-Clock 48.8-ps TDCs, and Pixel-Wise Integrated Histogramming’, IEEE J. Solid-State Circuits, vol. 54, no. 4, pp. 1137–1151, Apr. 2019, doi: 10.1109/JSSC.2018.2883720.

[71] C. Zhang et al., ‘A 240 × 160 3D-Stacked SPAD dToF Image Sensor With Rolling Shutter and In-Pixel Histogram for Mobile Devices’, IEEE Open J. Solid-State Circuits Soc., vol. 2, pp. 3–11, 2022, doi: 10.1109/OJSSCS.2021.3118332.

[72] T. E. Rahkonen and J. T. Kostamovaara, ‘The use of stabilized CMOS delay lines for the digitization of short time intervals’, IEEE J. Solid-State Circuits, vol. 28, no. 8, pp. 887–894, Aug. 1993, doi: 10.1109/4.231325.

[73] A. Mantyniemi, T. Rahkonen, and J. Kostamovaara, ‘A 9-channelIntegirated Time-to-DigitalConverter With Sub-nanosecondResolution’.

[74] P. Bailly et al., ‘A 16-channel digital TDC chip’, 1999.

[75] V. Kratyuk, P. K. Hanumolu, K. Ok, Un-Ku Moon, and K. Mayaram, ‘A Digital PLL With a Stochastic Time-to-Digital Converter’, IEEE Trans. Circuits Syst. I, vol. 56, no. 8, pp. 1612–1621, Aug. 2009, doi: 10.1109/TCSI.2008.2010109.

[76] S. Alahdab, A. Mantyniemi, and J. Kostamovaara, ‘A time-to-digital converter (TDC) with a 13-bit cyclic time domain successive approximation interpolator with sub-ps-level resolution using current DAC and differential switch’, in 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), Columbus, OH, USA: IEEE, Aug. 2013, pp. 828–831. doi: 10.1109/MWSCAS.2013.6674777.

[77] P. Keranen and J. Kostamovaara, ‘A Wide Range, 4.2 ps(rms) Precision CMOS TDC With Cyclic Interpolators Based on Switched-Frequency Ring Oscillators’, IEEE Trans. Circuits Syst. I, vol. 62, no. 12, pp. 2795–2805, Dec. 2015, doi: 10.1109/TCSI.2015.2485719.

[78] Y. Wang, W. Xie, H. Chen, and D. Day-Uei Li, ‘High-resolution time-to-digital converters (TDCs) with a bidirectional encoder’, Measurement, vol. 206, p. 112258, Jan. 2023, doi: 10.1016/j.measurement.2022.112258.

[79] P. Kwiatkowski, D. Sondej, and R. Szplet, ‘Subpicosecond resolution time interval counter with multisampling wave union type B TDCs in 28 nm FPGA device’, Measurement, vol. 209, p. 112510, Mar. 2023, doi: 10.1016/j.measurement.2023.112510.

[80] D. R. Hoppe, ‘Differential Time Interpolator’, Sep. 07, 1982 [Online]. Available: https://patents.google.com/patent/US4433919A/en

[81] D. K. Xie, Q. C. Zhang, G. S. Qi, and D. Y. Xu, ‘Cascading delay line time-to-digital converter with 75 ps resolution and a reduced number of delay cells’, Rev. Sci. Instrum., 2005.

[82] W. Xie, H. Chen, and D. D.-U. Li, ‘Efficient Time-to-Digital Converters in 20 nm FPGAs With Wave Union Methods’, IEEE Trans. Ind. Electron., vol. 69, no. 1, pp. 1021–1031, Jan. 2022, doi: 10.1109/TIE.2021.3053905.

[83] H. Chen and D. D.-U. Li, ‘Multichannel, Low Nonlinearity Time-to-Digital Converters Based on 20 and 28 nm FPGAs’, IEEE Trans. Ind. Electron., vol. 66, no. 4, pp. 3265–3274, Apr. 2019, doi: 10.1109/TIE.2018.2842787.

[84] H. Seo, G. Cho, S.-J. Kim, J.-H. Chun, and J. Choi, ‘Multievent Histogramming TDC With Pre–Post Weighted Histogramming Filter for CMOS LiDAR Sensors’, IEEE Sensors J., vol. 22, no. 23, pp. 22785–22798, Dec. 2022, doi: 10.1109/JSEN.2022.3217303.

[85] I. Nissinen, A. Mantyniemi, and J. Kostamovaara, ‘A CMOS time-to-digital

converter based on a ring oscillator for a laser radar', in ESSCIRC 2004 - 29th European Solid-State Circuits Conference (IEEE Cat. No.03EX705), Estoril, Portugal: IEEE, 2003, pp. 469–472. doi: 10.1109/ESSCIRC.2003.1257174.

[86] K.-C. Choi, S.-W. Lee, B.-C. Lee, and W.-Y. Choi, 'A Time-to-Digital Converter Based on a Multiphase Reference Clock and a Binary Counter With a Novel Sampling Error Corrector', IEEE Trans. Circuits Syst. II, vol. 59, no. 3, pp. 143–147, Mar. 2012, doi: 10.1109/TCSII.2012.2184370.

[87] F. Arvani and A. C. Carusone, 'Peak-SNR Analysis of CMOS TDCs for SPAD-Based TCSPC 3D Imaging Applications', IEEE Trans. Circuits Syst. II, vol. 68, no. 3, pp. 893–897, Mar. 2021, doi: 10.1109/TCSII.2020.3023631.

[88] F. Arvani and T. C. Carusone, 'A Reconfigurable 5-Channel Ring-Oscillator-Based TDC for Direct Time-of-Flight 3D Imaging', IEEE Trans. Circuits Syst. II, vol. 69, no. 5, pp. 2408–2412, May 2022, doi: 10.1109/TCSII.2022.3161650.

[89] S. Berrima, Y. Blaquiere, and Y. Savaria, 'Ring-Oscillator-Based High Accuracy Low Complexity Multichannel Time-to-Digital Converter Architecture for Field-Programmable Gate Arrays', IEEE Trans. Instrum. Meas., vol. 70, pp. 1–10, 2021, doi: 10.1109/TIM.2021.3106100.

[90] Z. Cheng, X. Zheng, M. J. Deen, and H. Peng, 'Recent Developments and Design Challenges of High-Performance Ring Oscillator CMOS Time-to-Digital Converters', IEEE Trans. Electron Devices, vol. 63, no. 1, pp. 235–251, Jan. 2016, doi: 10.1109/TED.2015.2503718.

[91] M. Z. Straayer and M. H. Perrott, 'A Multi-Path Gated Ring Oscillator TDC With First-Order Noise Shaping', IEEE J. Solid-State Circuits, vol. 44, no. 4, pp. 1089–1098, Apr. 2009, doi: 10.1109/JSSC.2009.2014709.

[92] D. Kim and S. Cho, 'A Hybrid PLL Using Low-Power GRO-TDC for Reduced In-Band Phase Noise', IEEE Trans. Circuits Syst. II, vol. 66, no. 2, pp. 232–236, Feb. 2019, doi: 10.1109/TCSII.2018.2848218.

[93] J. Hu, X. Wang, D. Li, Y. Liu, R. Ma, and Z. Zhu, 'A 50-ps Gated VCRO-Based TDC With Compact Phase Interpolators for Flash LiDAR', IEEE Trans. Circuits Syst. I, vol. 69, no. 12, pp. 5096–5107, Dec. 2022, doi: 10.1109/TCSI.2022.3200944.

[94] Yue Liu et al., 'A 6ps resolution pulse shrinking Time-to-Digital Converter as phase detector in multi-mode transceiver', in 2008 IEEE Radio and Wireless Symposium, Orlando, FL, USA: IEEE, Jan. 2008, pp. 163–166. doi: 10.1109/RWS.2008.4463454.

[95] S. Tisa, A. Lotito, A. Giudice, and F. Zappa, 'Monolithic time-to-digital converter with 20ps resolution', in ESSCIRC 2004 - 29th European Solid-State Circuits Conference (IEEE Cat. No.03EX705), Estoril, Portugal: IEEE, 2003, pp. 465–468. doi: 10.1109/ESSCIRC.2003.1257173.

[96] L. Xiang, P. Yang, T. Wu, and M. Zhou, 'Ultra compact pulse shrinking TDC on FPGA', Measurement, vol. 203, p. 111874, Nov. 2022, doi: 10.1016/j.measurement.2022.111874.

[97] P. Lu, Y. Wu, and P. Andreani, 'A 2.2-ps Two-Dimensional Gated-Vernier Time-to-Digital Converter With Digital Calibration', IEEE Trans. Circuits Syst. II, vol. 63, no. 11, pp. 1019–1023, Nov. 2016, doi: 10.1109/TCSII.2016.2548218.

[98] K. Cui, Z. Ren, X. Li, Z. Liu, and R. Zhu, 'A High-Linearity, Ring-Oscillator-Based,

Vernier Time-to-Digital Converter Utilizing Carry Chains in FPGAs', IEEE Trans. Nucl. Sci., vol. 64, no. 1, pp. 697–704, Jan. 2017, doi: 10.1109/TNS.2016.2632168.

[99]K. Cui and 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., vol. 69, no. 8, pp. 5941–5949, Aug. 2020, doi: 10.1109/TIM.2019.2959423.

[100] J.-C. Liu and Y.-X. Chen, 'DPLL-Based VRO of Time-to-Digital Converter', IEEE Solid-State Circuits Lett., vol. 6, pp. 45–48, 2023, doi: 10.1109/LSSC.2023.3242764.

[101] C.-C. Wang, O. L. J. A. Jose, and A. Avilala, 'A 5.4 ps resolution TDC design with anti-PVT-variation mechanism using 90-nm CMOS process', International Journal of Electronics, pp. 1–14, May 2023, doi: 10.1080/00207217.2023.2210303.

[102] W. F. Lin and H. P. Chou, 'A fast single slope ADC with vernier delay line technique', in 2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC), Orlando, FL: IEEE, Oct. 2009, pp. 313–317. doi: 10.1109/NSSMIC.2009.5401729.

[103] B. Markovic, S. Tisa, F. A. Villa, A. Tosi, and F. Zappa, 'A High-Linearity, 17 ps Precision Time-to-Digital Converter Based on a Single-Stage Vernier Delay Loop Fine Interpolation', IEEE Trans. Circuits Syst. I, vol. 60, no. 3, pp. 557–569, Mar. 2013, doi: 10.1109/TCSI.2012.2215737.

[104] M. Z. Straayer and M. H. Perrott, 'An efficient high-resolution 11-bit noise-shaping multipath gated ring oscillator TDC', in 2008 IEEE Symposium on VLSI Circuits, Honolulu, HI, USA: IEEE, Jun. 2008, pp. 82–83. doi: 10.1109/VLSIC.2008.4585960.

[105] S. Tancock, E. Arabul, N. Dahnoun, and S. Mehmood, 'Can DSP48A1 adders be used for high-resolution delay generation?', in 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro: IEEE, Jun. 2018, pp. 1–6. doi: 10.1109/MECO.2018.8406083.

[106] K. Paweł, 'Employing FPGA DSP blocks for time-to-digital conversion', Metrology and Measurement Systems, Jul. 2023, doi: 10.24425/mms.2019.130570.

[107] X. Qin et al., 'A high resolution time-to-digital-convertor based on a carry-chain and DSP48E1 adders in a 28-nm field-programmable-gate-array', Review of Scientific Instruments, vol. 91, no. 2, p. 024708, Feb. 2020, doi: 10.1063/1.5141391.

[108] M. Zhang, K. Yang, Z. Chai, H. Wang, Z. Ding, and W. Bao, 'High-Resolution Time-to-Digital Converters Implemented on 40-, 28-, and 20-nm FPGAs', IEEE Trans. Instrum. Meas., vol. 70, pp. 1–10, 2021, doi: 10.1109/TIM.2020.3036066.

[109] J. Wu and J. Xu, 'A Novel TDC Scheme: Combinatorial Gray Code Oscillator Based TDC for Low Power and Low Resource Usage Applications', in 2019 5th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), Vienna, Austria: IEEE, May 2019, pp. 1–7. doi: 10.1109/EBCCSP.2019.8836892.

[110] S. Araujo, R. Machado, and J. Cabral, 'Double-sampling Gray TDC with a ROS Interface for a LiDAR System', in 2021 7th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), Krakow, Poland: IEEE, Jun. 2021, pp. 1–8. doi: 10.1109/EBCCSP53293.2021.9502403.

[111] R. Machado, F. S. Alves, and J. Cabral, ‘Gray-Code TDC with Improved Linearity and Scalability for LiDAR applications’, in 2020 6th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), Krakow, Poland: IEEE, Sep. 2020, pp. 1–8. doi: 10.1109/EBCCSP51266.2020.9291359.

[112] Q. Shen et al., ‘A 1.7 ps Equivalent Bin Size and 4.2 ps RMS FPGA TDC Based on Multichain Measurements Averaging Method’, IEEE Trans. Nucl. Sci., vol. 62, no. 3, pp. 947–954, Jun. 2015, doi: 10.1109/TNS.2015.2426214.

[113] X. Qin, L. Wang, D. Liu, Y. Zhao, X. Rong, and J. Du, ‘A 1.15-ps Bin Size and 3.5-ps Single-Shot Precision Time-to-Digital Converter With On-Board Offset Correction in an FPGA’, IEEE Trans. Nucl. Sci., vol. 64, no. 12, pp. 2951–2957, Dec. 2017, doi: 10.1109/TNS.2017.2768082.

[114] P. Chen, Y.-Y. Hsiao, Y.-S. Chung, W. X. Tsai, and J.-M. Lin, ‘A 2.5-ps Bin Size and 6.7-ps Resolution FPGA Time-to-Digital Converter Based on Delay Wrapping and Averaging’, IEEE Trans. VLSI Syst., vol. 25, no. 1, pp. 114–124, Jan. 2017, doi: 10.1109/TVLSI.2016.2569626.

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

[116] Y. Zhou, Y. Wang, Z. Song, and X. Kong, ‘A High-Precision Folding Time-to-Digital Converter Implemented in Kintex-7 FPGA’, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, vol. 72, 2023.

[117] Z. Song, Y. Wang, and J. Kuang, ‘Implementation of 5.3 ps RMS precision and 350 M samples/second throughput time-to-digital converters with event sampling architecture in a Kintex-7 FPGA’, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 944, p. 162584, Nov. 2019, doi: 10.1016/j.nima.2019.162584.

[118] J. Y. Won and J. S. Lee, ‘Time-to-Digital Converter Using a Tuned-Delay Line Evaluated in 28-, 40-, and 45-nm FPGAs’, IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 7, pp. 1678–1689, 2016, doi: 10.1109/TIM.2016.2534670.

[119] N. Dutton et al., ‘Multiple-event direct to histogram TDC in 65nm FPGA technology’, in 2014 10th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), Grenoble: IEEE, Jun. 2014, pp. 1–5. doi: 10.1109/PRIME.2014.6872727.

[120] H. Chen, Y. Zhang, and D. D.-U. Li, ‘A Low Nonlinearity, Missing-Code Free Time-to-Digital Converter Based on 28-nm FPGAs With Embedded Bin-Width Calibrations’, IEEE Trans. Instrum. Meas., vol. 66, no. 7, pp. 1912–1921, Jul. 2017, doi: 10.1109/TIM.2017.2663498.

[121] 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, doi: 10.1109/TNS.2016.2606627.

[122] S. Liu, Y. Zhou, S. Liao, and X. Li, ‘A Low-Dead-Time FPGA-Based Time-to-Digital Converter Employing Resource-Efficient Downsampling-Multiplexing

Encoding and Dual-Histogramming for LiDAR Applications', IEEE Trans. Ind. Electron., pp. 1–11, 2024, doi: 10.1109/TIE.2023.3347843.

[123] 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, doi: 10.1109/TBCAS.2015.2389227.

[124] Z. Song, Y. Wang, and 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., vol. 13, no. 05, pp. P05012–P05012, May 2018, doi: 10.1088/1748-0221/13/05/P05012.

[125] 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, doi: 10.1109/TNS.2015.2421319.

[126] Y. Wang, J. Kuang, C. Liu, and Q. Cao, 'A 3.9-ps RMS Precision Time-to-Digital Converter Using Ones-Counter Encoding Scheme in a Kintex-7 FPGA', IEEE Trans. Nucl. Sci., vol. 64, no. 10, pp. 2713–2718, Oct. 2017, doi: 10.1109/TNS.2017.2746626.

[127] R. Szplet, D. Sondej, and G. Grzeda, 'High-Precision Time Digitizer Based on Multiedge Coding in Independent Coding Lines', IEEE Trans. Instrum. Meas., vol. 65, no. 8, pp. 1884–1894, Aug. 2016, doi: 10.1109/TIM.2016.2555218.

[128] W. Xie, Y. Wang, H. Chen, and D. D.-U. Li, '128-Channel High-Linearity Resolution-Adjustable Time-to-Digital Converters for LiDAR Applications: Software Predictions and Hardware Implementations', IEEE Trans. Ind. Electron., vol. 69, no. 4, pp. 4264–4274, Apr. 2022, doi: 10.1109/TIE.2021.3076708.

[129] M. Zhang, H. Wang, and Y. Liu, 'A 7.4 ps FPGA-Based TDC with a 1024-Unit Measurement Matrix', Sensors, vol. 17, no. 4, p. 865, Apr. 2017, doi: 10.3390/s17040865.

[130] S. Tancock, J. Rarity, and 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., vol. 71, pp. 1–11, 2022, doi: 10.1109/TIM.2022.3141753.

[131] K. Cui, J. Yu, J. Zou, and X. Li, 'A High-Resolution TDC Design Based on Multistep Fine Time Measurement by Utilizing Delay-Adjustable Looped Carry Chains on FPGAs', IEEE Trans. Instrum. Meas., vol. 72, pp. 1–10, 2023, doi: 10.1109/TIM.2023.3265129.

[132] D. Li, M. Liu, R. Ma, and Z. Zhu, 'An 8-ch LIDAR Receiver Based on TDC With Multi-Interval Detection and Real-Time $In~Situ$ Calibration', IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 5081–5090, Jul. 2020, doi: 10.1109/TIM.2019.2954173.

[133] K. Kuzmenko et al., '3D LIDAR imaging using Ge-on-Si single–photon avalanche diode detectors', Opt. Express, vol. 28, no. 2, p. 1330, Jan. 2020, doi: 10.1364/OE.383243.

[134] S. Yang, B. Li, M. Liu, Y.-K. Lai, L. Kobbelt, and S.-M. Hu, 'HeteroFusion: Dense Scene Reconstruction Integrating Multi-Sensors', IEEE Trans. Visual. Comput.

Graphics, vol. 26, no. 11, pp. 3217–3230, Nov. 2020, doi: 10.1109/TVCG.2019.2919619.

[135] Z. Zang, D. Xiao, and D. Day-Uei Li, ‘Non-fusion time-resolved depth image reconstruction using a highly efficient neural network architecture’, Opt. Express, vol. 29, no. 13, p. 19278, Jun. 2021, doi: 10.1364/OE.425917.

[136] M. H. Nunes et al., ‘Recovery of logged forest fragments in a human-modified tropical landscape during the 2015-16 El Niño’, Nat Commun, vol. 12, no. 1, p. 1526, Mar. 2021, doi: 10.1038/s41467-020-20811-y.

[137] C. Wang et al., ‘A real time coincidence system for high count-rate TOF or non-TOF PET cameras using hybrid method combining AND-logic and Time-mark technology’, in 2009 16th IEEE-NPSS Real Time Conference, Beijing, China: IEEE, May 2009, pp. 321–325. doi: 10.1109/RTC.2009.5321818.

[138] AVET, ‘ZedBoard Hardware User’s Guide’. 2014. [Online]. Available: https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf

[139] Xilinx, ‘Zynq-7000 SoC Data Sheet: Overview (DS190)’, 2018, [Online]. Available: https://www.xilinx.com/ support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf

[140] ARM, ‘Learn the architecture - An introduction to AMBA AXI’. 2020. [Online]. Available: https://developer.arm.com/documentation/102202/0300/AXI-protocol-overview

[141] P. Kwiatkowski, D. Sondej, and R. Szplet, ‘A brief review of wave union TDCs’, in 2021 7th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), Krakow, Poland: IEEE, Jun. 2021, pp. 1–6. doi: 10.1109/EBCCSP53293.2021.9502360.

[142] Y. Wang, X. Zhou, Z. Song, J. Kuang, and 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., vol. 66, no. 10, pp. 2275–2281, Oct. 2019, doi: 10.1109/TNS.2019.2938571.

[143] P. Kwiatkowski and R. Szplet, ‘Multisampling wave union time-to-digital converter’, in 2020 6th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), Krakow, Poland: IEEE, Sep. 2020, pp. 1–5. doi: 10.1109/EBCCSP51266.2020.9291363.

[144] Y. Wang and C. Liu, ‘A Nonlinearity Minimization-Oriented Resource-Saving Time-to-Digital Converter Implemented in a 28 nm Xilinx FPGA’, IEEE Transactions on Nuclear Science, vol. 62, no. 5, pp. 2003–2009, 2015, doi: 10.1109/TNS.2015.2475630.

[145] P. Kwiatkowski and R. Szplet, ‘Efficient Implementation of Multiple Time Coding Lines-Based TDC in an FPGA Device’, IEEE Trans. Instrum. Meas., vol. 69, no. 10, pp. 7353–7364, Oct. 2020, doi: 10.1109/TIM.2020.2984929.

[146] N. Lusardi, F. Garzetti, and A. Geraci, ‘The role of sub-interpolation for Delay-Line Time-to-Digital Converters in FPGA devices’, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 916, pp. 204–214, Feb. 2019, doi: 10.1016/j.nima.2018.11.100.

[147] J. Wu, ‘Uneven bin width digitization and a timing calibration method using

cascaded PLL', in 2014 19th IEEE-NPSS Real Time Conference, Nara, Japan: IEEE, May 2014, pp. 1–4. doi: 10.1109/RTC.2014.7097534.

[148] Xilinx, '7 Series FPGAs SelectIO Resources User Guide (UG471)', 2018, [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug471_7Series_SelectIO

[149] K.-J. Choi and D.-W. Jee, 'Design and Calibration Techniques for a Multichannel FPGA-Based Time-to-Digital Converter in an Object Positioning System', IEEE Trans. Instrum. Meas., vol. 70, pp. 1–9, 2021, doi: 10.1109/TIM.2020.3011490.

[150] J. Zhang and D. Zhou, 'An 8.5-ps Two-Stage Vernier Delay-Line Loop Shrinking Time-to-Digital Converter in 130-nm Flash FPGA', IEEE Trans. Instrum. Meas., vol. 67, no. 2, pp. 406–414, Feb. 2018, doi: 10.1109/TIM.2017.2769239.

[151] P. Chen et al., 'High-Precision PLL Delay Matrix With Overclocking and Double Data Rate for Accurate FPGA Time-to-Digital Converters', IEEE Trans. VLSI Syst., vol. 28, no. 4, pp. 904–913, Apr. 2020, doi: 10.1109/TVLSI.2019.2962606.

[152] R. Szplet, Z. Jachna, P. Kwiatkowski, and K. Rozyc, 'A 2.9 ps equivalent resolution interpolating time counter based on multiple independent coding lines', Meas. Sci. Technol., vol. 24, no. 3, p. 035904, Mar. 2013, doi: 10.1088/0957-0233/24/3/035904.

[153] Xilinx, 'Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs (UG768)', 2013, [Online]. Available: https://docs.xilinx.com/v/u/en-US/7series_hdl

[154] D. Li, R. Ma, X. Wang, J. Hu, M. Liu, and Z. Zhu, 'DTOF Image LiDAR With Stray Light Suppression and Equivalent Sampling Technology', IEEE Sensors J., vol. 22, no. 3, pp. 2358–2369, Feb. 2022, doi: 10.1109/JSEN.2021.3136931.

[155] V. Sesta, A. Incoronato, F. Madonini, and F. Villa, 'Time-to-digital converters and histogram builders in SPAD arrays for pulsed-LiDAR', Measurement, vol. 212, p. 112705, May 2023, doi: 10.1016/j.measurement.2023.112705.

[156] J. Deng, P. Yin, X. Lei, Z. Shu, M. Tang, and F. Tang, 'A Tunable Parameter, High Linearity Time-to-Digital Converter Implemented in 28-nm FPGA', IEEE Trans. Instrum. Meas., vol. 70, pp. 1–12, 2021, doi: 10.1109/TIM.2021.3117373.

[157] Y. Hua and D. Chitnis, 'A Highly Linear and Flexible FPGA-Based Time-to-Digital Converter', IEEE Trans. Ind. Electron., vol. 69, no. 12, pp. 13744–13753, Dec. 2022, doi: 10.1109/TIE.2021.3128912.

[158] Xilinx, 'ZCU104 Evaluation Board User Guide (UG1267)', 2018, [Online]. Available: https://www.xilinx.com/support/documents/boards_and_kits/zcu104/ug1267-zcu104-eval-bd.pdf

[159] Xilinx, 'KCU105 Board User Guide'. 2019. [Online]. Available: https://www.mouser.com/datasheet/2/903/ug917-kcu105-eval-bd-1596175.pdf

[160] Digilent, 'NetFPGA-SUME Reference Manual'. 2016. [Online]. Available: https://digilent.com/reference/programmable-logic/netfpga-sume/reference-manual

[161] Stanford Research System, 'CG635 — DC to 2.05 GHz low-jitter clock generator'. [Online]. Available: https://www.thinksrs.com/downloads/pdfs/catalog/CG635c.pdf

[162] Xilinx, ‘Vivado Design Suite Tcl Command Reference Guide’, 2019, [Online]. Available: https://docs.amd.com/v/u/2019.2-English/ug835-vivado-tcl-commands

[163] P. Gui, ‘Analog-to-digital converters and time-to-digital converters for high-energy physics experiments’, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 1057, p. 168649, Dec. 2023, doi: 10.1016/j.nima.2023.168649.

[164] M. Ablikim et al., ‘Design and construction of the BESIII detector’, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 614, no. 3, pp. 345–399, Mar. 2010, doi: 10.1016/j.nima.2009.12.050.

[165] Z. Xue, H. Liang, J. Li, J. Ge, and Q. Li, ‘A 128-channel time-to-digital converter based on the 24-phase shift-clock sampling method for RPC’, J. Inst., vol. 18, no. 02, p. T02007, Feb. 2023, doi: 10.1088/1748-0221/18/02/T02007.

[166] G. Kaissis and R. Braren, ‘Pancreatic cancer detection and characterization—state of the art cross-sectional imaging and imaging data analysis’, Transl. Gastroenterol. Hepatol, vol. 4, pp. 35–35, May 2019, doi: 10.21037/tgh.2019.05.04.

[167] W. T. Kwong et al., ‘Low Rates of Malignancy and Mortality in Asymptomatic Patients With Suspected Neoplastic Pancreatic Cysts Beyond 5 Years of Surveillance’, Clinical Gastroenterology and Hepatology, vol. 14, no. 6, pp. 865–871, Jun. 2016, doi: 10.1016/j.cgh.2015.11.013.

[168] J. Wang, S. Liu, L. Zhao, X. Hu, and Q. An, ‘The 10-ps Multitime Measurements Averaging TDC Implemented in an FPGA’, IEEE Trans. Nucl. Sci., vol. 58, no. 4, pp. 2011–2018, Aug. 2011, doi: 10.1109/TNS.2011.2158551.

[169] B. Wu et al., ‘Design of Time-to-Digital Converters for Time-Over-Threshold Measurement in Picosecond Timing Detectors’, IEEE Trans. Nucl. Sci., vol. 68, no. 4, pp. 470–476, Apr. 2021, doi: 10.1109/TNS.2021.3060069.

[170] N. Lusardi, F. Garzetti, N. Corna, R. D. Marco, and A. Geraci, ‘Very High-Performance 24-Channels Time-to-Digital Converter in Xilinx 20-nm Kintex UltraScale FPGA’.

[171] P. Kwiatkowski, D. Sondej, and R. Szplet, ‘Bubble-Proof Algorithm for Wave Union TDCs’, Electronics, vol. 11, no. 1, p. 30, Dec. 2021, doi: 10.3390/electronics11010030.

[172] K. Cui, X. Li, Z. Liu, and R. Zhu, ‘Toward Implementing Multichannels, Ring-Oscillator-Based, Vernier Time-to-Digital Converter in FPGAs: Key Design Points and Construction Method’, IEEE Trans. Radiat. Plasma Med. Sci., vol. 1, no. 5, pp. 391–399, Sep. 2017, doi: 10.1109/TRPMS.2017.2712260.

[173] Teledyne LeCroy, ‘WaveRunner 6 Zi Oscilloscopes 400 MHz –4 GHz’. 2017. [Online]. Available: https://cdn.teledynelecroy.com/files/pdf/waverunner-6zi-datasheet.pdf

[174] M. Parsakordasiabi, I. Vornicu, A. Rodriguez-Vazquez, and R. Carmona-Galan, ‘An Efficient TDC Using a Dual-Mode Resource-Saving Method Evaluated in a 28-nm FPGA’, IEEE Trans. Instrum. Meas., vol. 71, pp. 1–13, 2022, doi: 10.1109/TIM.2021.3136267.

[175] D. Portaluppi, K. Pasquinelli, I. Cusini, and F. Zappa, ‘Multi-Channel FPGA

Time-to-Digital Converter With 10 ps Bin and 40 ps FWHM', IEEE Trans. Instrum. Meas., vol. 71, pp. 1–9, 2022, doi: 10.1109/TIM.2022.3152324.

[176] IEEE Computer Society, 'IEEE Standard for Verilog Hardware Description Language'. 2015. [Online]. Available: https://www.eg.bucknell.edu/~csci320/2016-fall/wp-content/uploads/2015/08/verilog-std-1364-2005.pdf