HHS Public Access

Author manuscript

IEEE Trans Instrum Meas. Author manuscript; available in PMC 2020 October 30.

Published in final edited form as:

IEEE Trans Instrum Meas. 2019 October ; 68(10): 3647–3660. doi:10.1109/tim.2018.2880940.

A 2.3-ps RMS Resolution Time-to-Digital Converter Implemented in a Low-Cost Cyclone V FPGA

Tengjie Sui,
State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Zhixiang Zhao,
School of Biomedical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

Siwei Xie,
State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Yangze Xie,
State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Yanyan Zhao,
State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Qiu Huang,
School of Biomedical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

Jianfeng Xu,
State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Qiyu Peng [Member, IEEE]
Department of Molecular Biophysics and Integrated Bioimaging, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA

Abstract

We present a nonuniform multiphase (NUMP) method to construct a high-resolution time-to-digital converter (TDC) for low-cost field-programmable gate array (FPGA) devices. The NUMP method involves a system clock being passed through a series of delay elements to generate multiple clocks with different phase shifts. The phases of the rising and falling edges of all the

Corresponding authors: qiuhuang@sjtu.edu.cn Qiu Huang; jfxu@hust.edu.cn Jianfeng Xu; qpeng@lbl.gov Qiyu Peng.

clocks are sorted in order and the states of all the clocks are latched when a hit signal arrives. The sizes of the time bins (and precision) of the NUMP method are not limited by the uniformity and minimum value of the time delays of the delay lines. In theory, any delay sources with small jitters in an FPGA, not just very fine carry chains, can be used in the NUMP method to delay and randomize the clocks. Thus, the NUMP method can achieve excellent TDC timing resolutions in low-cost FPGAs without very fine delay lines. We implemented four NUMP TDC channels in a low-cost FPGA device (an Altera Cyclone V 5CEBA4F23C7N). The performance of the four NUMP TDCs was evaluated using both internal and external pulses. The root mean square (rms) for the timing resolution measured using the internal and the external pulses with short-time intervals (less than 1 ns) was 2.3 and 5.2 ps, respectively. A 14.1-ps rms timing resolution was measured at a time interval of 517 ns. The NUMP method is suitable for applications that require a number of high-performance TDC channels in a low-cost FPGA.

Index Terms

Field-programmable gate array (FPGA); nonuniform multiphase (NUMP) method; root mean square (rms) resolution; time-to-digital converter (TDC)

I. Introduction

HIGH-PRECISION time-to-digital converters (TDCs), which can measure short-time intervals accurately, are widely used in light detection and ranging (LiDAR), laser-ranging devices, automated test equipment (ATE), medical time-of-flight (TOF) positron emission tomography (PET) cameras, and high energy physics (HEP) experiments [1]–[5]. Many applications require TDCs with a resolution better than 10 ps that can measure the traveling distance of a photon in a vacuum with 3-mm precision [6], [7].

Conventionally, the development of TDCs with a resolution better than 50 ps requires using dedicated application-specific-integrated-circuits (ASIC) [8] – [13]. However, the development of ASICs demands special expertise and tools. It is an expensive and long process involving iterative design, fabrication, and evaluation, which is prohibitive to most developers and researchers outside of a few large research centers and commercial companies.

Complex programmable logic device-based TDCs [14] and field-programmable gate array (FPGA)-based TDCs [15]–[20], constructed with off-the-shelf low-cost components, form a promising practical alternative to the conventional ASIC-based TDCs. FPGA-based TDCs are inexpensive, easy to implement and reconfigure within short development cycles, highly scalable, and versatile across a variety of applications. Hundreds of channels of TDCs can be implemented in a single low-cost FPGA with a timing performance that outstrips conventional ASIC-based TDCs.

The most commonly used approach to implementing high-performance TDCs in an FPGA is to combine a coarse timestamp, which is used to extend the dynamic range, and a fine timestamp, which is responsible for yielding better time resolution [21]. Over the past few decades, various methods have been developed to construct the fine time module, which can

be grouped into two categories: 1) type I: the clock-sample-shifted-events (CSSEs) method. It is also known as the tapped delay line (TDL) method illustrated in Fig. 1(a) and 2) type II: the event-latch-shifted-clocks (ELSCs) method that is also known as the multiphase method illustrated in Fig. 1(b). Note that the Vernier method is not within the scope of this paper [22]–[24].

In type I CSSE TDCs [Fig. 1(a)], the hit signal propagates along a TDL and is sampled by a global clock as a fine timestamp. Currently, the most commonly used way of structuring TDLs is to cascade carry chains, which are predefined logic resources for arithmetic circuits in FPGAs [25], [26]. In 2006, Song *et al.* [27] achieved 69.5-ps resolution and 65.8-ps precision by using carry chains for TDL construction.

The wide and nonuniform tap delays in the carry chains limit the performance of the TDL TDCs. In 2008, Wu and Shi [28] proposed a novel wave union method to solve that issue. The wave-union TDC achieved an improved resolution by taking multiple measurements using a single TDL structure. The time resolution was improved to 25 ps using wave union launcher A (WU-A), and further improved to 10 ps in wave union launcher B (WU-B) by taking 16 × of measurements [28]. In the past decade, many works have been performed to further optimize the wave union design. This included automatic temperature correction, multichain measurement averaging, and implementation in different series of FPGA processes [29]–[32]. Szplet *et al.* [33] combined three six-edge multiedge coding (another name for wave union) and achieved a subpicosecond resolution (902 fs) with a precision better than 6 ps in Xilinx Spartan-6 device.

Multiple TDL architecture, which uses several independent TDLs, is another solution to an issue of the wide and nonuniform tap delays in the carry chains. The multiple TDL TDCs are often combined with the two-stage interpolation architecture to reduce the length of the TDLs [34]–[36]. For example, in Szplet *et al.*'s [34] design, an eight-phase clock was used in the first stage to divide the system clock into eight segments before applying the multiple TDL method in the second stage.

Despite the great progress that has been made in implementing type I CSSE TDC technology in FPGAs, there are still some challenges that need to be overcome. First of all, expensive high-end FPGAs with fast carry chains are required if outstanding timing performance is going to be achieved [37], [38]. Second, retuning or redesigning TDCs when migrating them from one FPGA to another with a different internal structure is tedious and time-consuming. For example, in some high-end FPGAs, the wave union method is not applicable because the TDL have different delay values for their rising and falling edges [39]. Third, the skew between the sampling clocks and the TDL flip-flops could cause the bubble problem, and lead to the imperfect thermometer code and the degraded time resolution in the conventional TDL TDCs [19], [32].

The type II ELSC TDCs uses the hit signal to latch the states of multiphase clocks. The type II ELSC TDCs are usually very resource-efficient, but have rather poor time resolution [39]–[41]. As shown in Fig. 1(b), a phase locking loop (PLL) or delay locked loop in an FPGA is normally employed to generate multiple clocks with a fixed phase difference. Here, it is 45°

across four clocks. The status of the D flip-flops is sampled using a test hit signal. The result is then sent to a decoding module to obtain the fine timestamp. For the multiphase method, every clock is able to provide two bins with both the rising and falling edges. In theory, the TDC bin size can be expressed as $T_{CLK}/2N$, where T_{CLK} is the period of the system clock and N represents the number of clocks. Since both the frequency and the number of clocks inside an FPGA are limited, it is not likely to achieve a root mean square (rms) resolution better than 10 ps in a conventional type II multiphase TDC. In addition, there could be large skews on the transmission path from the hit signal to all of the registers. That also limits the performances of the conventional type II multiphase TDCs.

In this paper, we present the nonuniform multiphase (NUMP) TDC that is a type II ELSC TDC with significantly improved timing performance. Very different from the type I CSSE TDCs and the conventional type II multiphase TDCs mentioned earlier, the NUMP TDC uses the hit signal to latch many copies of system clocks with random phase delays uniformly distributed between 0 and 2π. We validated the NUMP TDC design implemented in a Cyclone V FPGA. An excellent timing performance (a 1.56-ps bin size and 2.3-ps rms resolution) has been achieved.

The remaining parts of this paper are organized as follows. Sections II, III, and IV describe the operating principles, implementation, and experimental results of the NUMP TDC, respectively. Sections V and VI discuss the experimental results and conclusion.

II. Operating Principles

As shown in Fig. 2, the architecture of the NUMP TDC is very similar to that of the conventional multiphase TDCs. The major difference is that the NUMP TDC uses a phase-shifted clocks generator (PSCG) to produce many copies of system clocks with different phase shifts, instead of a few system clocks with a fixed uniform phase delay (for example, four clocks with a phase delay of 45°). Upon the rising edge of the hit signal, the registers latch a sequence, e.g., “1101… 11,” which is then decoded to obtain the fine timestamp.

In order to achieve a good timing performance, the clocks generated by the PSCG should have phase shifts distributed between 0 and 2π as uniformly as possible. As shown in Fig. 3, there are two typical ways to construct the PSCG. The first way is to adapt the parallel delay structure [Fig. 3(a)]. In this structure, many delay units are organized in parallel to generate the clocks. This is the simplest structure. However, the designers need to have excessive knowledge on the delay features of the delay units, carefully select and arrange them to achieve the uniform phase shifts distribution. Thus, it is very challenging to implement this structure in practice.

The second way is to adapt the series delay structure [Fig. 3(b)] that is essentially the same as the conventional TDL structure. In this structure, many units are organized to form a train of delays with the maximum phase shift φ_N [shown in Fig. 3(b)] larger than 2π. Any delay units with small jitters, large or small, uniform or nonuniform, can be used to construct the train. The usable delay units in the FPGA include the combinational logic delay units, carry chains, routing delays, and so on [42], [43].

Two general rules are suggested in constructing the train with delay units to achieve the optimal timing performance. First, the order of the delay units in the train should be arranged strategically to make phase shifts distributed between 0 and 2π as uniformly as possible. Second, the delay units with large jitters should be placed at the end of the train to minimize the average additive jitters along the train.

Note that a clock with a phase shift φ larger than 2π is equivalent to the clock with a phase shift of Mod (φ,2π). Thus, the train can have the maximum phase shift φ_N multiple times larger than 2π. However, the longer the train, the larger the additive jitters accumulated along the train. Architectures with multiple trains (Fig. 4) are suggested to reduce the accumulation of the additive jitters. In Fig. 4(a), the PSCG consists of M trains of N delay units ($\varphi_{i,N} > 2\pi, i \in [1, M]$), instead of one train of $M \times N$ delay units. In Fig. 4(b), the PSCG consists of four trains of N delay units ($\varphi_{i,N} > 0.5\pi$, $i \in [1, 4]$). A PLL is used to generate four clocks with 0.5π phase shifts.

In theory, the more delay units in the PSCG of the NUMP TDC, the more slices a reference clock cycle can be divided into. This results in a smaller and better distributed bin, and a better timing performance. Thus, the designer can conveniently balance the time performance and FPGA resource consumption for different applications.

III. TDC Implementation

A. Top-Level Diagram of NUMP TDC

The diagram of the NUMP TDC is shown in Fig. 5. The PSCG generates clocks with random phase shifts. The CAL_1 and CAL_2 are two internal signals for clock sorting and calibration. In the calibration mode, either CAL_1 or CAL_2 is selected to latch the states of the clocks in the register bank and the sampling results are sent to the clock sorting module for calibration.

In the normal operation mode, the states of the clocks with random phase shifts are latched simultaneously by the input hit signal. The latched states of the clocks are sent to the decoding module to calculate the fine timestamp. The input hit signal is also fed to the coarse time module. The coarse time module outputs a 16-bit course timestamp and a check bit for the correction of the metastable error caused by the asynchrony of the hit signal and the system clock. Last, the fine timestamp and the latched coarse timestamp are combined and saved in the FIFO to wait for being transmitted to the host PC through an USB cable.

B. Structure of PSCG

In this paper, the Altera's Cyclone V family FPGAs was chosen to construct the NUMP TDC. The logic array block in the Cyclone V device is composed of basic building blocks known as adaptive logic modules (ALMs). These can be configured to implement logic functions, arithmetic functions, and register functions. Two adders and four registers are integrated into every single ALM.

The carry chain buffers between dedicated adders are the widely used FPGA resources in TDL TDCs, and we also used carry chain buffers to validate the NUMP TDC method. The

series delay structure, or the TDL structure, was chosen to construct the PSCG. As shown in Fig. 6, when the system clock is fed into the first adder, multiple clocks are generated from the sum-out pins of the adders, whose inputs are preset to fixed values. Note that every adder is paired up with one register.

C. Clock Sorting

Since the phase relationships between the clocks generated by the PSCG are nonuniform, a clock sorting operation has to be undertaken to accurately identify and store the phase shifts among the clocks in a lookup table (LUT). After this, the TDCs are able to measure the time intervals. This clock sorting operation also works as a bin-by-bin calibration, thus avoiding the negative influence of any differential nonlinearity (DNL) and integral nonlinearity (INL) that might be caused by uneven bin sizes.

1) Tests Using CAL_1: In the clock sorting process, one of the clocks is chosen to serve as a reference among the multiple TDL-generated clocks. As indicated in Fig. 7, there are two typical types of phase shift relationships between C_0 and C_x. C_x can be any other clock generated by the PSCG. For one type of shift, the value of Cx would be "0" at the rising edge of C_0 [Fig. 7(a)]. The converse of this is shown in Fig. 7(b).

In order to distinguish between these two possibilities, the periodic signal labeled as CAL_1 in Fig. 5 is generated inside the FPGA to latch the registers bank. Note that CAL_1 is first generated by the PLL inside the FPGA, and then divided to obtain a specific frequency. As shown in Fig. 8(a), the clock period of CAL_1 is $T_{\mathrm{CAL\_1}}$, which can also be expressed as (n * T_{SYS} + C), where "n" is a large integer depending on how long it takes for the processor to handle one result, T_{SYS} stands for the period of the system clock, and C represents a short-time interval.

In the design using a system clock of 1000 MHz (period = 1000 ps), we configured the PLL to generate a 4.9-MHz clock. The 4.9-MHz clock was then divided by 49 135 to generate the clock CAL_1. The period of CAL_1 (10027551020.41 ps) is 10027 551 time of the period of 1000-MHz system clock (1000 ps) with an offset/reminder of C = 20.41 ps. Thus, CAL_1 is equivalent to a high-frequency clock that samples the D flip-flops of the TDL every 20.41 ps. This is shown in Fig. 8(b). After multiple measurements have been made by CAL_1, the value of C_x at the rising edge of C_0 can be obtained. The type of relationship existing between C_x and C_0 can now be determined as demonstrated by Fig. 7.

2) Code Density Tests With CAL_2: As a typical statistical testing method, code density tests are widely used in type I TDL TDC to construct bin-by-bin calibration so that accurate bin sizes can be obtained. There are two methods to perform statistical code density tests. The first method is to use a random hit signal as the input. However, it is not practical to generate a perfect random signal in an FPGA. The second method is to use a periodical hit signal that is asynchronous with the system clocks. Thus, the time differences of the rising edges of the periodical hit signal and the system clock distribute uniformly within the range of the period of the system clock. The second method is used in this paper. A periodic hit signal, represented as CAL_2 in Fig. 5, is generated by the PLL to latch the registers. The frequency of the CAL_2 is 6.7821 MHz, and it is asynchronous with the system clock for

the purpose of the statistical code density test. The bin width can be calculated from the times of various sampling results because there is an equal probability of a hit signal arriving at any time during the clock cycle.

The duty ratio of a clock can be calculated conveniently by dividing the summation of "1" by the total number of measurements made in the density measurements. In addition, as demonstrated in Fig. 7, there are four possible states for the pair (C_x, C_0) when they are sampled by the hit signal: "11," "10," "01," and "00." The ratio of a state to the clock period can also be obtained by counting how often it occurs.

After the tests with CAL_1 and CAL_2, the edge position of most clocks can be identified. For instance, if the measured time intervals of these four states ("11," "10," "01," and "00") are 300, 200, 200, and 300 ps, respectively, the test result from CAL_1 is "0." The positions of the rising and falling edges of C_x are 200 and 700 ps, respectively.

3) Other Considerations: In some cases, a different clock instead of C_0 has to be used as the reference clock for clock sorting. This is the case with the clocks tagged C_{X1} and C_{X2} in Fig. 9. For clock C_{X1}, its combination with C_0 only yields three states: "11," "10," and "00." The absence of "01" implies that it is not feasible to convert the code density test results into the rising and falling edge positions. With regard to C_{X2}, the test results using CAL_1 may be incorrect because the time interval between its falling edge and the rising edge of Co is smaller than the test time interval C. Therefore, another clock (Csr in Fig. 6) with edge positions determined in the previous steps is used as the reference clocks for the calculation of the rising and falling edges of C_{X1} and C_{X2}.

Note that all the calculations, statistical processing and clock sorting mentioned earlier were executed in a NIOS II processor, a 32-bit embedded-processor designed specifically for the Altera family of FPGAs. All other modules, including the NUMP TDL, the coarse counter, the decoding module, and FIFOs, were implemented directly in the FPGA. It takes the NIOS II processor about 15 s to calibrate a channel. It is necessary to perform dynamic calibration for applications with variable ambient conditions. There are different strategies to implement dynamic calibration for different applications. For example, the single events captured by a PET detector are ideal random events independent of the system clock. Those data can be used to calibrate the TDCs dynamically and in real time. In applications with very low event rate, the dynamic calibration can also be performed during the intervals between events.

D. Decoding

After the clock sorting operation is completed, an LUT is generated to record the time of the rising and falling edges for every single clock. Table I shows a demonstrative LUT with values selected from the real experimental data. Only five clocks are listed here for ease of comprehension. The basic decoding process involves confirming which two edges the hit signal is distributed across. The most straightforward way of going about this is to shorten the possible range by decoding the clocks one by one. For example, we assume that the latched states of "$C_0C_1C_2C_3C_4$" are "11101," as shown in Fig. 10. By decoding C_0, the range of the fine time of hit is narrowed down from [0, 1000 ps] to [0, 468 ps]. After decoding C_1, it is narrowed down to [45, 468 ps]. This procedure is repeated until the

latched state of last clock C_4 is decoded. Finally, the range of the fine time of hit is narrowed down to [203, 461 ps]. Thus, the fine timestamp, calculated by averaging the left and the right boundary of the final interval, is 332 ps.

Dead time for NUMP TDCs is largely determined by the speed of the decoding process. For an NUMP TDC with 400 delayed clocks and 800 clock edges, the sequential decoding method described earlier requires 400 clock cycles to decode the fine time. More efficient decoding algorithms, such as the binary search algorithm, could significantly reduce the dead time for an NUMP TDC.

Before performing the binary search, the rising and falling edges of the all the clocks are sorted, indexed and saved in an LUT. Table II shows a demonstrative LUT with 10 sorted edges. The initial range of the fine time of hit is set to [0, 1000 ps] and the midpoint index is 5.

The record in index 5 of Table II is the falling edge of C_0 (468 ps). Thus, the first step is to check the latched state of C_0. As shown in Fig. 10, the latched state of C_0 is 1. That indicates that the hit happens before the falling edges of C_0. Therefore, the range of the fine time of hit is narrowed down to [0, 468 ps] and the midpoint index is updated to 3.

The record in index 3 of Table II is the rising edge of C_2 (203 ps). Thus, the second step is to check the latched state of C_2. The latched state of C_2 is 1, indicating that the hit happens after the rising edges of C_2. Therefore, the range of the fine time of hit is narrowed down to [203, 468 ps] and the midpoint index is updated to 4.

The record in index 4 of Table II is the rising edge of C_3 (461 ps). Thus, the third step is to check the latched state of C_3. The latched state of C_3 is 0, indicating that the hit happens before the rising edges of C_3. Therefore, the range of the fine time of hit is narrowed down to [203, 461 ps].

It takes five steps to decode five latched clocks using the sequential decoding method. By contrast, it only takes three steps by applying the binary search algorithm. Generally, it takes the binary search algorithm Ceil($\log_2 2N$) steps to N latched clocks, where Ceil() is an operation of rounding to the nearest integer greater than the input. For an NUMP TDC with 400 delayed clocks and 800 clock edges, the binary search algorithm requires only 10 clock cycles to decode the fine time.

E. Coarse Time Module

The coarse time counter is driven by the system clock and is synchronous with the system clock, while the hit signal is asynchronous with the system clock. Thus, there is a chance that the outputs of the coarse time counter are in the transient or metastable state when the hit signal arrives. When that happens, the coarse time latched with the hit signal might be incorrect. One method to solve this problem is to use two coarse counters driven by two clocks with different phases [44]. Thus, there is at least one counter has stable outputs when the hit arrives. The outputs from that counter are latched as coarse time.

In this paper, we developed a different method to solve that issue. As shown in Fig. 11, the coarse time module consists of four registers (R_1, R_2, R_3, and R_4), an AND gate, a conventional binary coarse counter, and the registers to buffer the coarse time stamps. Note that $Q1$, $Q2$, and $Q3$ are synchronous with the system clock. Thus, there is no metastable problem to latch the outputs of the course counter with $Q3$.

However, the metastable states still exist in registers R_1 and R_4. The metastable states in registers R_1 can lead to one clock offset in Q_3. A check bit was introduced to fix this problem. Fig. 12(a) and (b) shows two possible timing diagrams of hit signal, the system clock, the outputs of the coarse counter, the outputs of three registers (Q_1, Q_2, and Q_4), the output of the AND gate (Q_3), the state of the check bit, and the values of the coarse timestamp, when the metastable state happens in register R_1.

In Fig. 12(a), Q_3 was set to 1 right after the rising edge of hit. Thus, the values of the check bit and course timestamp are 1 and 2, respectively. In Fig. 12(b), Q_3 was set to 1 one clock after the rising edge of hit. Thus, the values of the check bit and course timestamp are 0 and 3, respectively. Note that the summations of the check bit and course timestamp are the same in the two cases. Thus, the final metastable-state-free course timestamp can be calculated by adding the check bit into the course timestamp.

IV. Experimental Validation

To validate our proposed method, we constructed four identical channels of NUMP TDCs in the evaluation board (DE0-CV) of a low-cost FPGA (Altera Cyclone V 5CEBA4F23C7N, price: U.S. $66.63). 320 delay units were used in each TDC to generate 320 delayed 1-GHz clocks. The 1-GHz clocks are produced directly by the PLL inside the FPGA. Note that the PLLs inside FPGA are able to generate internal clocks with a frequency higher than 1 GHz, although the datasheet of Cyclone V suggests that the output frequency of the PLLs should not exceed 550 MHz. The 1 GHz is not a necessary condition for NUMP TDCs. However, the higher the clock frequency, the less is FPGA resource required to implement the NUMP TDCs.

A. Clock Sorting

Fig. 13(a) and (b) shows 32 of the 320 channels of delayed clocks before and after the sorting operation. Note that we sorted both the rising and falling edges. Thus, there are 64 sorted edges in Fig. 13(b). Fig. 13(c) shows the time delay for these sorted edges, which are distributed unevenly in the 0–1000-ps range. The images in Fig. 13 are taken from the results of the clock sorting operation.

Fig. 14 shows the distribution of the time intervals between the adjacent edges. The majority of the time bins had a width of less than 5 ps. The mean and standard deviation (STD) of the bin widths are 1.56 and 1.67 ps, respectively.

B. Differential and Integral Nonlinearities

The ununiformed time bin widths could cause serious nonlinear error. Fig. 15 shows the DNL and INL of the ununiformed time bins. The LSB in our design is 1.56 ps. The clock

sorting operation described in Section III was implemented to remove the nonlinear error of the NUMP TDCs by calibrating the ununiformed time bins one by one.

C. Timing Resolution

The intrinsic timing resolutions for the four TDCs (TDC A, TDC B, TDC C, and TDC D) were established using internal pulses. generated by an internal PLL of the FPGA. The frequency of the internal pulse signals was asynchronous to the system clock. The rms resolutions between the four NUMP TDCs [shown in Fig. 15(a)] range between 2.2 and 2.4 ps. The mean and STD were 2.3 and 0.08 ps, respectively. The intrinsic uncertainty was most likely caused by the nonuniform bin size (shown in Fig. 14), together with jitters along the routes of the delayed clocks and the hit signals. The measured time intervals in Fig. 16(a) were caused by the routing delays of the internal pulse signals. Note that all rms values in this paper represent the resolution of a single TDC channel (single-shot resolution), which is 0.707 of the dual-channel results.

We also measured the timing resolution for the four TDCs using two channels of external pulse signals generated by an Analog Devices Inc. clock generation board (AD9548/PCBZ). Once again, the frequency of the external pulse signals was not correlated with the system clock. The two external pulse signals were fed to the FPGA through two low-voltage differential signaling (LVDS) ports using two pairs of SubMiniature version A-to-DuPont cables. TDC A and TDC B shared one external pulse signal. TDC C and TDC D shared the other external pulse signal.

Fig. 16(b) shows the rms resolutions across the four NUMP TDCs, measured using external pulses. The timing resolutions measured between channel A and B (2.7 ps) and between channel C and D (2.5 ps) are essentially the intrinsic timing resolutions of the TDCs. This is because A and B, and C and D share the same external pulse signals, respectively. However, these two resolutions are not quite as good as those measured with the internal pulses [Fig. 16(a)]. This suggests that the routes from the FPGA IO ports to the TDCs have larger jitters than the routes from the internal PLL to the TDCs.

The timing resolutions measured between channel A and C (5.2 ps), channel A and D (5.1 ps), channel B and C (5.3 ps), and channel B and D (5.2 ps) are significantly larger than their intrinsic timing resolutions. The extra jitters were most likely introduced by either the signal generator and/or the cables, or by the FPGA’s LVDS ports and/or the low-cost dual in-line package connectors in the evaluation board.

The intrinsic timing resolutions for the NUMP TDCs are related to the number of the delayed clocks. The larger the number of delayed clocks, the smaller the average bin size and the better the rms resolution. The intrinsic timing resolution can also be improved by combining and averaging two NUMP TDCs. Note, however, that both methods are only able to improve the intrinsic timing resolution of the TDCs. The jitters introduced by external sources, including the signal generators, cables, connectors, and LVDS ports, cannot be reduced by increasing the number of the delayed clocks or by averaging two TDCs.

D. Effects of the Number and Frequency of the Clocks

The NUMP TDCs constructed with 160, 200, and 320 delayed clocks were tested using both internal and external pulses with the measured time intervals less than 1 ns. The results are shown in Fig. 17. The intrinsic timing resolution measured with internal pulses (σ_{IN}) improved from 3.5 to 2.3 ps (a 37.1% improvement). The timing resolution measured using external pulses (σ_{EX}) improved from 5.7 to 5.2 ps (an 8.8% improvement). Fig. 17 also shows that the timing resolution can be improved by averaging two TDCs. Again, the improvements gained by averaging NUMP TDCs are more significant for intrinsic timing resolutions that are measured with internal pulses than those measured with external pulses.

The clock frequency can also affect the timing resolution for the NUMP TDCs. The lower the clock frequency, the larger the size of the time bins, and the worse the rms resolution becomes. Thus, more delayed clocks are needed to achieve a similar timing resolution when the clock frequency is reduced. We implemented and tested NUMP TDCs constructed with 400 delayed 500-MHz clocks. The timing resolution measured with internal and external pulses was 3.5 and 6.2 ps, respectively.

E. Measurements of Longer Time Intervals

Fig. 18 shows the timing resolutions measured with pairs of external pulses with different time intervals. The results show that the longer the interval, the worse the timing resolution. The rms timing resolution decreases to 14.1 ps when the timing interval increases to 517 ns. Those results are expected since the accuracy of the measurement of the time difference of two signals with large time interval is determined by both the intrinsic resolution of the TDC and the long-term stability of the system clock. The longer the time intervals, the bigger the impacts of the long-term instability on the timing measurements. In most applications, such as LiDAR, TOF-PET cameras, and HEP experiments, the TDCs are used to measure the time differences between two events happen in a short-time interval. In those applications, a conventional crystal oscillator (XO) or temperature compensated crystal oscillator (TCXO) with a frequency fluctuations of about 1–20 ppm usually meet the requirements. For more sophisticated applications, more accurate oscillators, such as the rubidium atomic frequency standard (RbXO) and high-performance atomic standard (Cs) oscillators, should be considered.

F. Temperature Stabilities

It is well-known that process, voltage, and temperature can have a dramatic effect upon the timing resolution for an FPGA-based TDC. In practice, the variance of the core voltage of an FPGA can be easily stabilized in the range of ±12 mV, thus making any voltage variation influence negligible. The temperature of an FPGA, however, can vary over a larger range. We, therefore, also tested the effects of FPGA temperature on the resolution of the NUMP TDCs that were implemented with 400 delayed 500-MHz clocks. The FPGA evaluation board was placed on a heat panel and an infrared thermometer was used to monitor FPGA temperature. The red curve shown in Fig. 19 indicates that the TDC resolution degraded slightly when temperatures increased from 20° to 56°. Variations in TDC resolution caused by temperature fluctuations can be reduced by redoing the clock sorting operation (shown by

the blue curve in Fig. 19). However, for most applications, this is not likely to be necessary because the variation in TDC resolution is very small.

V. Discussion

In this paper, we have presented a novel NUMP method to construct high-precision and high-resolution TDCs in FPGA devices. We have validated this method by implementing four TDCs in a low-cost FPGA (an Altera Cyclone V 5CEBA4F23C7N). The intrinsic timing resolution for the NUMP TDCs measured with internal pulses was 2.3 ps ± 0.1 ps. Timing resolutions better than 14.1 ps were achieved at the time intervals up to 517 ns. The NUMP TDCs had good stabilities at temperatures ranging from 20 °C to 56 °C.

A. Uniqueness of the NUMP Architecture

Most FPGA-based TDCs (including both the type I CSSE TDCs such as wave-union TDCs and multiple TDL TDCs, and the type II ELSC TDCs) are based on internal delays in the FPGA. However, they have significantly different internal architectures. And their performances, in terms of timing resolution, cost, power consumption, complexity, and scalability, are also significantly different.

The conventional type I CSSE TDCs use one TDL chain consisting of n processing elements (carry chains) to generate many copies of the delayed input hit signal, and a system clock to latch them. The timing of the input hit signal is derived from the latched codes. The TDL architecture requires a chain of narrow and uniform delays, in order to achieve good timing performance. Unfortunately, the widths of the delays in different FPGA devices, normally ranging from a few picoseconds to tens of picoseconds, are very different. In addition, the delays among the carry chains in the same device could also be very different. For example, the time delays of the carry chain across sections of logic elements are usually significantly larger than those in the same section.

Many different strategies and architectures have been developed to overcome the issue caused by the wide and nonuniform delays in the delay chain of TDL TDCs. For example, the BOUNCE TDC presented by Salomon [45] employs n chains that consist of merely one element (internal signal path), rather than having one chain consisting of n processing elements. The wave-union TDCs presented by Wu and Shi [28] use one chain consisting of n processing elements (carry chains). The wave-union architectures were designed to generate multiple edges when a hit signal arrived. Those edges were designed in such a way that at least one edge is latched on the location of the chain with narrow and/or uniform delays. The multiple TDL TDCs [33], [35] use multiple chains (also called time coding lines) consisting of n processing elements to avoid the effects of nonuniform and wide delays between sections of logic elements. Note that the strategies of the wave-union TDCs and the multiple TDL TDCs are very similar to some extent. The major difference between the two methods is that the wave-union TDCs generate multiple edges and measure their timing in a single chain, instead of physically using multiple chains. Thus, the wave-union TDCs are more cost-efficient and power-efficient than the multiple TDL TDCs. There were also efforts to further improve the timing performance of type I CSSE TDCs by implementing wave-union in multiple TDL chains [33].

The NUMP TDC is a type II ELSC TDC that has a unique architecture different from any of the above-mentioned type I TDL TDCs.

First, the NUMP TDC is a multiphase clock TDC which use one copy of hit signal to latch many copies of the system clocks with different phases. Unlike the conventional multiphase clock TDCs that use several system clocks with a fixed uniform phase delay, the NUMP TDC employs many copies of system clock with different phase delays. Thus, this TDC is named NUMP TDC.

Second, the timing performance of an NUMP TDC is not impacted by nonuniform and wide delays in the delay chain, as soon as the phase delay of the clocks are adequately randomized. In theory, any delay sources with small jitters in an FPGA, no matter the delays are large or small, uniform or nonuniform, can be utilized to randomize the system clocks. There is no need for the NUMP TDC to avoid carry chains like the BOUNCE TDC, use multiple edges like the wave-union TDCs, or use multiple chains like the multiple TDL TDCs.

Third, the clock sorting operation has to be performed to characterize and sort the clocks with different phase delays in the NUMP TDC. By contrast, the delays along the TDLs of the conventional type I TDL TDCs are in order naturally. The phase shifts of the conventional type I multiphase TDCs are predetermined. Thus, there is no need to sort them in both the conventional type I and type II TDCs. Note that the “bubble” problem, an imperfection of the thermometer bit pattern, happens fairly often in type I TDL TDCs. The NUMP TDC has no “bubble” problem.

Last, it is neither feasible nor necessary to adapt the strategies of the wave-union TDCs to improve the performance of the NUMP TDC. The edges in a wave union are normally tens of picoseconds from each other. It is impossible to separate the clock signals latched by different edges in a wave union. Thus, the wave union method can only be used in the type I TDL TDCs, not in type II multiphase TDCs.

B. Features of the NUMP TDC

The NUMP TDC has some unique features compared to many conventional FPGA-based TDCs. First of all, it is resource-efficient and robust. Both the rising and falling edges of delayed clocks can be used to fractionize the cycle of the system clock. This means that an NUMP TDC is able to achieve the same bin width with only half the length of the carry chain required by many other TDL methods. We noticed significant changes in duty cycles after the clocks passed the delay lines. Fig. 20 shows the histogram of the duty cycles of 320 delayed clocks originating from the same 1 GHz clock with a 50% duty cycle. The duty cycles for these clocks ranged from 32% to 60%. The mean and the rms were 44.3% and 5.6%, respectively. Therefore, changes in clock duty cycles can significantly affect the timing resolution for a conventional type II multiphase TDC. However, they do not affect the performance of an NUMP TDC at all. In addition, NUMP TDC is immune to the “bubble” problem, an imperfection of the thermometer bit pattern happen fairly often in type I TDL TDCs [33], [44].

Second, the resolution of an NUMP TDC is not limited by the uniformity and minimum value of the time delays within a delay line. Any delay sources with small jitters in an FPGA, not only very fine carry chains, can be used in the NUMP method to delay and randomize the clocks. As a result, NUMP TDCs can achieve excellent timing resolution in low-cost FPGAs without the need for very fine delay lines.

Third, the intrinsic timing resolution of NUMP TDCs measured with internal pulses is related to the number of delayed clocks. The larger the number of delayed clocks, the smaller is the sizes of the time bins and the better the rms resolution. This means that one can either increase the number of delayed clocks to achieve a higher resolution or decrease the number of delayed clocks to save FPGA resources.

Last, a fundamental principle of the NUMP TDC method is the generation of hundreds of clocks, with their edges distributed as evenly as possible in a clock cycle. The clock sorting operation sorts the edges of all the clocks automatically. Thus, there is no need to manually tune, adjust or rearrange the delay lines when migrating NUMP TDC designs from one FPGA to another.

C. Some Practical Issues

The measured resolution of the TDC is affected by jitters, which can be grouped into two categories: 1) the external jitters, including the jitters in the external hit signal, the jitters along the external cables, the additive jitters introduced by the digital input ports of the FPGA and 2) the internal jitters, including the jitters introduced by the internal routes, gates, carry chains, and system clocks in the FPGA. The TDC resolution measured with internal pulses is only affected by the internal jitters and thus a good indicator of the intrinsic resolution. By contrast, the TDC resolution measured with external pulses is affected by both the internal jitters and the external jitters. Thus, the TDC resolution measured with external pulses (5.2 ps) is significantly larger than that measured with internal pulses (2.3 ps). Those results also indicated that the rms of external jitters in this paper was about 4.7 ps.

A clock may have short-term clock-to-clock phase jitters and long-term phase shifts. Oscillators with different accuracies (XO, TCXO, RbXO, and Cs oscillators) may be selected for different applications. The PLLs are commonly used to reduce the short-term clock-to-clock phase jitters. However, they are not able to reduce the long-term phase shifts of a clock. In addition, a PLL may generate a different phase delay between the input clock and output clock every time the system boots up. That is a common issue for many FPGA-based TDCs that use PLLs. However, that is not an issue for the NUMP TDC, because the clock sorting operation, performed every time the FPGA system boots up, is able to adapt the phase shift introduced by the PLL automatically.

The measurement range of presented TDC is depended on the number of the bits of the counter in coarse time module. We implemented a 16-bit counter in the coarse time module with a 500-MHz system clock. The measurement range is 0.13 ms (65536*2 ns).

VI. Conclusion

In this section, the NUMP TDC is a novel low-cost, high-performance FPGA-based TDC that is able to achieve an excellent (2.3 ps) timing resolution. It is a strong contender for a variety of applications, including TOF-LiDAR, ATE, medical TOF-PET, and HEP experiments. As a next step, we will implement multiple (>50) channels of NUMP TDCs in a low-cost Altera Cyclone V FPGA device so as to meet the needs of the next generation of sub-10-ps TOF-PET scanners [6], [7], [46], [47].

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 51627807, in part by the National Natural Science Foundation-Guangdong Joint Funds of China under Grant U1501256, and in part by the National Institutes of Health, National Institute of Biomedical Imaging and Bioengineering, under Grant R01EB006085. The Associate Editor coordinating the review process was Dario Petri.

Biographies

Yangze Xie was born in Shaoxing, Zhejiang, China, in 1995. He received the B.S. degree from the School of Mechanical Design Manufacturing and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is currently pursuing the master's degree with the School of Mechanical Science and Engineering.

His current research interests include data analysis and software development of high-performance positron emission tomography systems.

Tengjie Sui was born in Yantai, Shandong, China, in 1993. He received the B.S. degree from the School of Mechanical Design Manufacturing and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2016, where he is currently pursuing the master's degree with the School of Mechanical Science and Engineering.

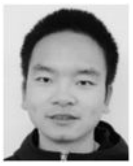
His current research interests include high-precision time measurement.

Yanyan Zhao was born in Hebi, Henan, China, in 1984. She received the M.Sc. degree from the School of Communication Engineering, Xidian University, Xi'an, China, in 2009.

She is currently a Circuit System Development Engineer with the State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. Her current research interests include design, development, and debugging of circuit systems.

Zhixiang Zhao was born in Zhumadian, Henan, China, in 1993. He received the B.S. degree in biomedical engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai.

He is currently a Visiting Student with the Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA. His current research interests include hardware and software designs of high-performance positron emission tomography systems.



Siwei Xie received the B.S. degree in mechanical engineering from the Wuhan University of Technology, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan.

He is currently a Visiting Student with the Department of Molecular Biophysics and Integrated Bioimaging, Lawrence Berkeley National Laboratory, Berkeley, CA, USA. His current research interests include nuclear medicine and functional imaging.

Qiu Huang received the Ph.D. degree in electrical engineering from the University of Utah, Salt Lake City, UT, USA.

She is currently a Research Professor with the School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is an Adjunct Research Professor with the Department of Nuclear Medicine, Ruijin Hospital, School of Medicine. Her current research interests include designs of instrumentation and algorithm in nuclear medicine imaging.

Jianfeng Xu received the B.S. degree from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2001, and the Ph.D. degree in mechanical engineering from the University of California at San Diego, San Diego, CA, USA, in 2008.

Since 2012, he has been a Professor with the Mechanical Science and Engineering, Huazhong University of Science and Technology, where he is currently involved in the development of precision engineering and instrumentation of nuclear medicine and functional imaging.

Qiyu Peng (M’04) received the Ph.D. degree in biomedical engineering from Tsinghua University, Beijing, China, in 2003.

He is currently a Career Scientist with the Department of Molecular Biophysics and Integrated Bioimaging, Lawrence Berkeley National Laboratory, Berkeley, CA, USA. His current research interests include nuclear medicine and functional imaging, biomedical ultrasound imaging, wireless biomonitoring, biorobotics, and studies on the mechanism and rehabilitation of neurological impairments including stroke, spinal cord injury, and urinary continence.

References

[1]. Lai J-C and Hsu T-Y, “Cost-effective time-to-digital converter using time-residue feedback,” IEEE Trans. Ind. Electron, vol. 64, no. 6, pp. 4690–4700, 6 2017.

[2]. Liu S, Feng C, An Q, Heng Y, and Sun S, “BES III time-of-flight readout system,” IEEE Trans. Nucl. Sei, vol. 57, no. 2, pp. 419–427, 4 2010.

[3]. Palojärvi R, Määttä K, and Kostamovaara J, “Integrated time-of-flight laser radar,” IEEE Trans. Instrum. Meas, vol. 46, no. 4, pp. 996–999, 8 1997.

[4]. Moses WW, “Time of flight in PET revisited,” IEEE Trans. Nucl. Sei, vol. 50, no. 5, pp. 1325–1330, 10 2003.

[5]. Hou L, Guo Y, Huang G, and Shu R, “A time-to-digital converter used in photon-counting LIDAR,” J. Infr. Millim. Waves, vol. 31, no. 3, pp. 243–247, 6 2012.

[6]. Lecoq P, “Pushing the limits in time-of-flight PET imaging,” IEEE Trans. Radiat. Plasma Med. Sei, vol. 1, no. 6, pp. 473–485, 11 2017.

[7]. Zhao JW et al., “Reaching time resolution of less than 10 ps with plastic scintillation detectors,” Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip, vol. 823, pp. 41–46, 7 2016.

[8]. Mauricio J, Gascón D, Ciaglia D, Gómez S, Fernandez G, and Sanuy A, “MATRIX: A 15 ps resistive interpolation TDC ASIC based on a novel regular structure,” J. Instrum, vol. 11, no. 12, p. Cl2047, 2016.

[9]. Perktold L and Christiansen J, “A multichannel time-to-digital converter ASIC with better than 3 ps RMS time resolution,” J. Instrum, vol. 9, no. 1, p. CO 1060, 2014.

[10]. Russo S, Petra N, De Caro D, Barbarino G, and Strollo AGM, “A 41 ps ASIC time-to-digital converter for physics experiments,” Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip, vol. 659, no. 1, pp. 422–427, 12 2011.

[11]. Meng XT, Levin DS, Chapman JW, Li DC, Yao ZE, and Zhou B, “Latency study of the high performance time to digital converter for the ATLAS Muon Spectrometer trigger upgrade,” J. Instrum, vol. 11, no. 9, p. P02008, 2016.

[12]. Cheng Z, Zheng X, Deen MJ, and Peng H, “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, 1 2016.

[13]. Marko vie B, Tamborini D, Villa F, Tisa S, Tosi A, and Zappa F, “10 ps resolution, 160 ns full scale range and less than 1.5% differential non-linearity time-to-digital converter module for high performance timing measurements,” Rev. Sei. Instrum, vol. 83, no. 7, p. 074703, 2012.

[14]. Levine PM and Roberts GW, “A high-resolution flash time-to-digital converter and calibration scheme,” in Proc. Int. Test Conf, 2004, pp. 1148–1157.

[15]. Szplet R, Kalisz J, and Szymanowski R, “Interpolating time counter with 100 ps resolution on a single FPGA device,” IEEE Trans. Instrum. Meas, vol. 49, no. 4, pp. 879–883, 8 2000.

[16]. Junnarkar SS, O’Connor P, and Fontaine R, “FPGA based self calibrating 40 picosecond resolution, wide range Time to digital converter,” in Proc. IEEE Nucl. Sei. Symp. Conf, 10 2008, pp. 3434–3439.

[17]. Wang J, Liu S, Shen Q, Li H, and An Q, “A fully fledged TDC implemented in field-programmable gate arrays,” IEEE Trans. Nucl. Sei, vol. 57, no. 2, pp. 446–450, 4 2010.

[18]. Zheng J, Cao P, Jiang D, and An Q, “Low-cost FPGA TDC with high resolution and density,” IEEE Trans. Nucl. Sei, vol. 64, no. 6, pp. 1401–1408, 6 2017.

[19]. Chen H, Zhang Y, and Li DD-U, “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, 6 2017.

[20]. Frankowski R, Gurski M, and Płóciennik P, “Optical methods of the delay cells characteristics measurements and their applications,” Opt. Quantum Electron, vol. 48, no. 3, p. 188, 2016.

[21]. Szplet R, Kalisz J, and Jachna Z, “A 45 ps time digitizer with a two-phase clock and dual-edge two-stage interpolation in a field programmable gate array device,” Meas. Sci. Technol, vol. 20, no. 2, p. 025108, 2009.

[22]. Cui K, Ren Z, Li X, Liu Z, and Zhu R, "A high-linearity, ring-oscillator-based, Vernier time-to-digital converter utilizing carry chains in FPGAs," IEEE Trans. Nucl. Sei, vol. 64, no. 1, pp. 697–704, 1 2017.

[23]. Lu P, Wu Y, and Andreani P, "A 2.2-ps two-dimensional gated-Vernier time-to-digital converter with digital calibration," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 63, no. 11, pp. 1019–1023, 11 2016.

[24]. Cheng Z, Deen MJ, and Peng H, "A low-power gateable Vernier ring oscillator time-to-digital converter for biomedical imaging applications," IEEE Trans. Biomed. Circuits Syst, vol. 10, no. 2, pp. 445–454, 4 2016. [PubMed: 26168446]

[25]. Wu J, Shi Z, and Wang IY, "Firmware-only implementation of time-to-digital converter (TDC) in field-programmable gate array (FPGA)," in Proc. IEEE Conf. Rec. NSS, vol. 1, 10 2003, pp. 177–181.

[26]. Fishbum MW, Menninga LH, Favi C, and Charbon E, "A 9.6 ps, FPGA-based TDC with multiple channels for open source applications," IEEE Trans. Nucl. Sei, vol. 60, no. 3, pp. 2203–2208, 6 2013.

[27]. Song J, An Q, and Liu S, "A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays," IEEE Trans. Nucl. Sei, vol. 53, no. 1, pp. 236–241, 2 2006.

[28]. Wu J and Shi Z, "The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay," in Proc. IEEE Conf. Rec. NSS, 10 2008, pp. 3440–3446.

[29]. Shen Q 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. Sei, vol. 62, no. 3, pp. 947–954, 6 2015.

[30]. Pan W, Gong G, and Li J, "A 20-ps time-to-digital converter (TDC) implemented in field-programmable gate array (FPGA) with automatic temperature correction," IEEE Trans. Nucl. Sei, vol. 61, no. 3, pp. 1468–1473, 6 2014.

[31]. Qi J, Gong H, and Liu Y, "On-chip real-time correction for a 20-ps wave union time-to-digital converter (TDC) in a field-programmable gate array (FPGA)," IEEE Trans. Nucl. Sei, vol. 59, no. 4, pp. 1605–1610, 8 2014.

[32]. Liu C and Wang Y, "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. Sei, vol. 62, no. 3, pp. 773–783, 6 2015.

[33]. Szplet R, Sondej D, and Grzçda G, "High-precision time digitizer based on multiedge coding in independent coding lines," IEEE Trans. Instrum. Meas, vol. 65, no. 8, pp. 1884–1894, 8 2016.

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

[35]. Chaberski D, "Time-to-digital-converter based on multiple-tapped-delay-line," Measurement, vol. 89, pp. 87–96, 7 2016.

[36]. Szplet R, Jachna Z, Kwiatkowski P, and Rozyc K, "A 2.9 ps equivalent resolution interpolating time counter based on multiple independent coding lines," Meas. Sei. Technol, vol. 24, no. 3, p. 35904, 2013.

[37]. Wang Y and Liu C, "A 3.9 ps time-interval RMS precision time-to-digital converter using a dual-sampling method in an UltraScale FPGA," IEEE Trans. Nucl. Sei, vol. 63, no. 5, pp. 2617–2621, 10 2016.

[38]. Wang Y and Liu C, "A 4.2 ps time-interval RMS resolution time-to-digital converter using a bin decimation method in an UltraScale FPGA," IEEE Trans. Nucl. Sei, vol. 63, no. 5, pp. 2632–2638, 10 2016.

[39]. Wang Y, Kuang P, and Liu C, "A 256-channel multi-phase clock sampling-based time-to-digital converter implemented in a Kintex-7 FPGA," in Proc. IEEE Conf. Rec. IMT, 5 2016, pp. 1–5.

[40]. Chen K, Liu S, and An Q, "A high precision time-to-digital converter based on multi-phase clock implemented within field-programmable-gate-array," Nucl. Sei. Techn, vol. 21, no. 2, pp. 123–128, 2010.

[41]. Fries MD and Williams JJ, "High-precision TDC in an FPGA using a 192 MHz quadrature clock," in Proc. IEEE Conf. Rec. NSS, vol. 1, 11 2002, pp. 580–584.

[42]. Zhang M, Wang H, and Liu Y, "A 7.4 ps FPGA-based TDC with a 1024-unit measurement matrix," Sensors, vol. 17, no. 4, p. 865, 4 2017.

[43]. Wang H, Zhang M, and Yao Q, "A new realization of time-to-digital converters based on FPGA internal routing resources," IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 60, no. 9, pp. 1787–1795, 9 2013. [PubMed: 24658711]

[44]. Wu J, "Several key issues on implementing delay line based TDCs using FPGAs," IEEE Trans. Nucl. Sei, vol. 57, no. 3, pp. 1543–1548, 6 2010.

[45]. Joost R and Salomon R, "Bounce, a new approach to measure subnanosecond time intervals," in Proc. IEEE Int. Conf. Field Program. Logic Appl, 9 2008, pp. 511–514.

[46]. Gundacker S, Auffray E, Pauwels K, and Lecoq P, "Measurement of intrinsic rise times for various L(Y)SO and LuAG scintillators with a general study of prompt photons to achieve 10 ps in TOF-PET," Phys. Med. Biol, vol. 61, no. 7, pp. 2802–2837, 2016. [PubMed: 26982798]

[47]. Surti S, Shore AR, and Karp JS, "Design study of a whole-body PET scanner with improved spatial and timing resolution," IEEE Trans. Nucl. Sei, vol. 60, no. 5, pp. 3220–3226, 10 2013.

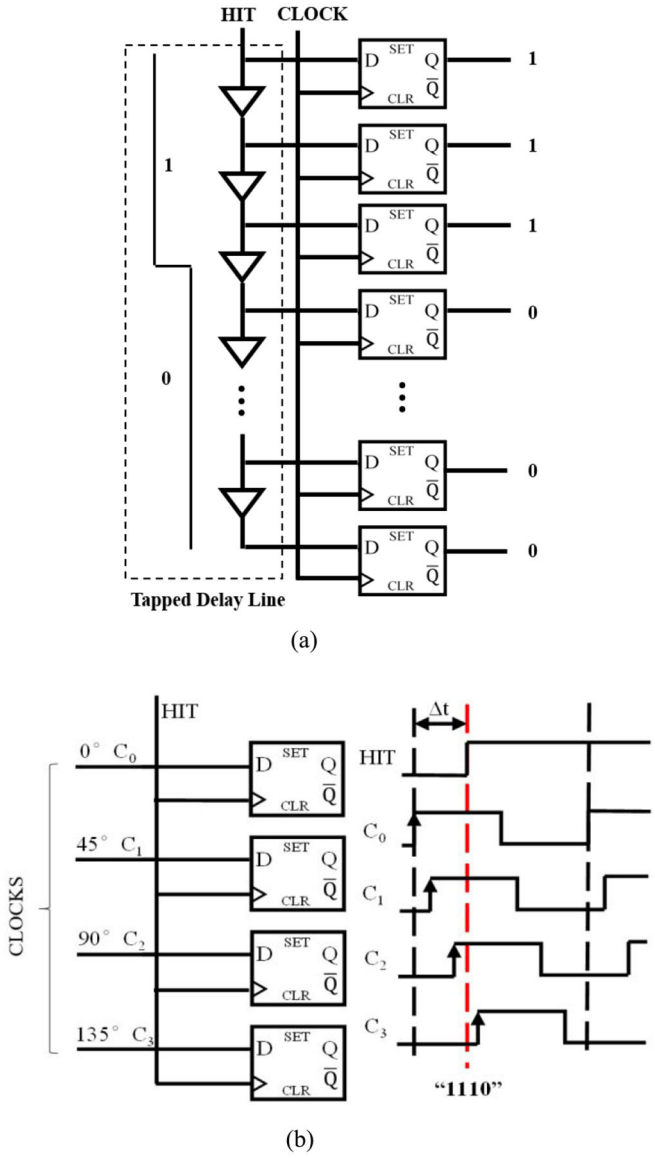


Fig. 1.
Architectures of the (a) type I CSSEs TDC and (b) type II ELSCs TDC.

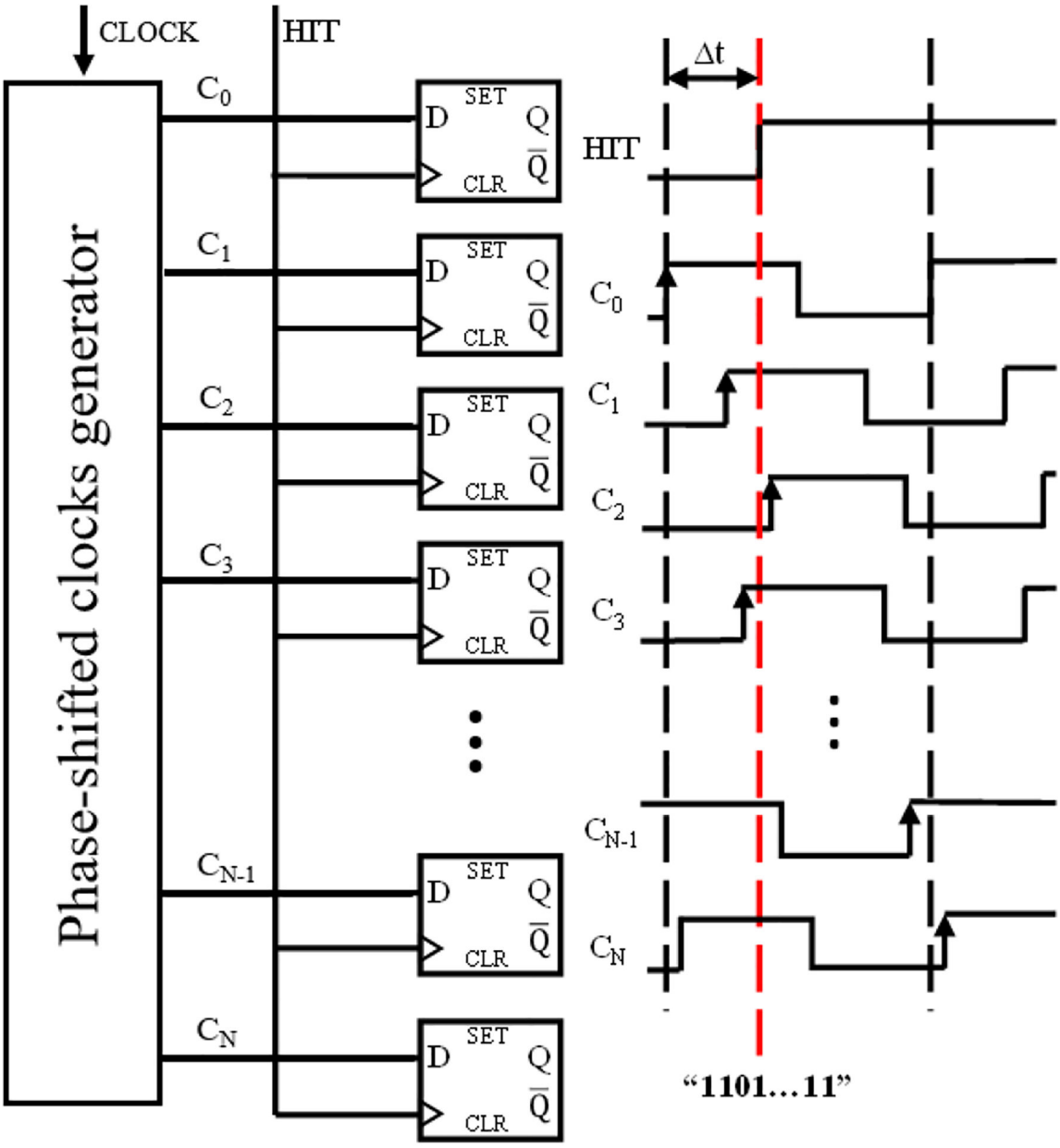


Fig. 2.
Architecture of the NUMP TDC.

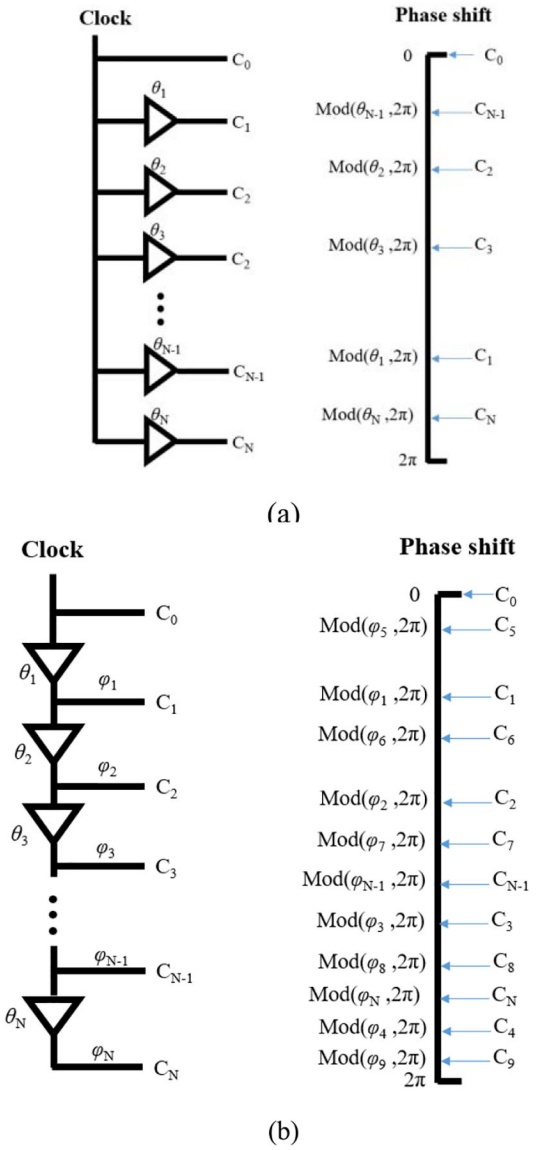


Fig. 3.

Two typical ways to construct the PSCG. Mod (φ,2π) represents the module operation over 2π. (a) Parallel delay architecture. (b) Series delay architecture.

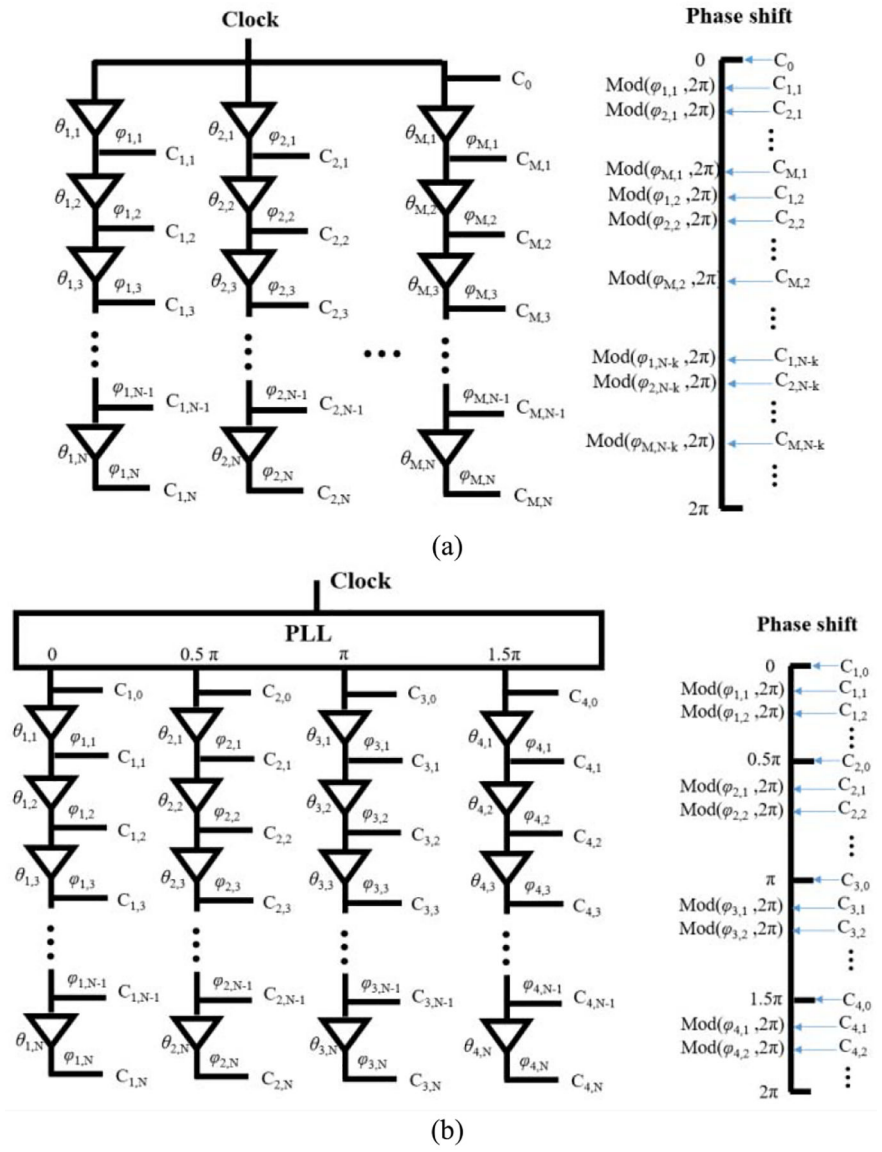


Fig. 4.

PSCGs constructed with multiple trains of delay units. (a) PSCG consists of M trains of N delay units. (b) PSCG consists of four trains of N delay units.

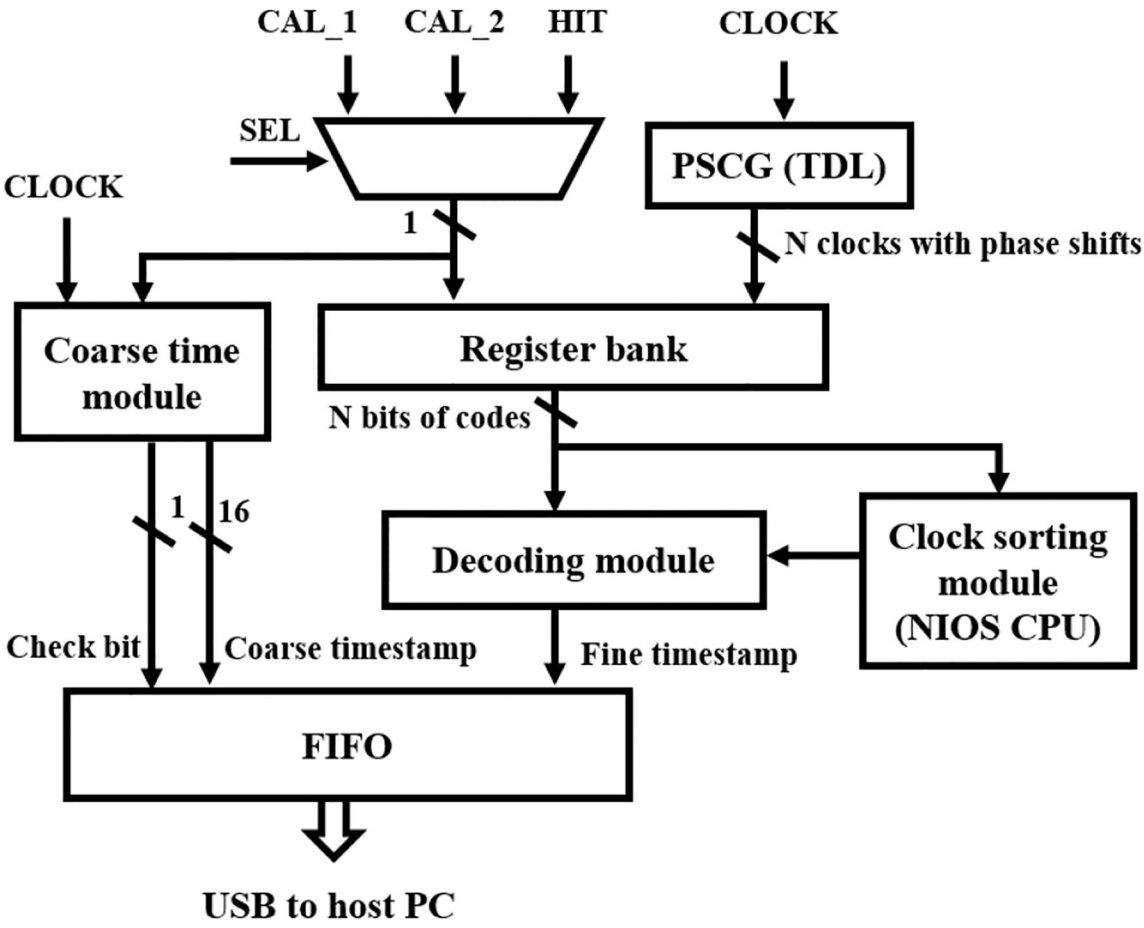


Fig. 5.
Diagram of the NUMP TDC.

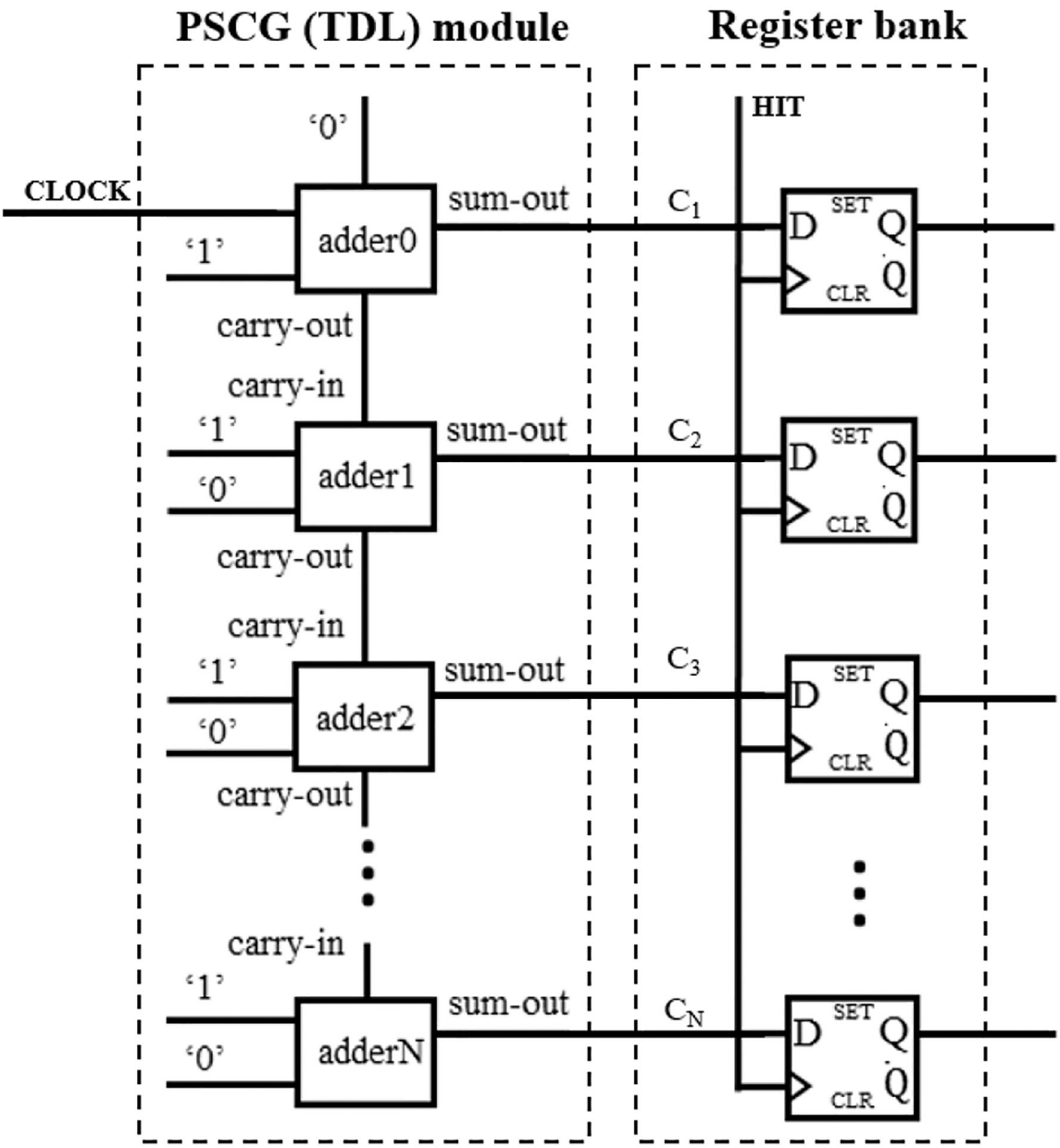


Fig. 6.
PSCG was constructed with a train of carry chains.

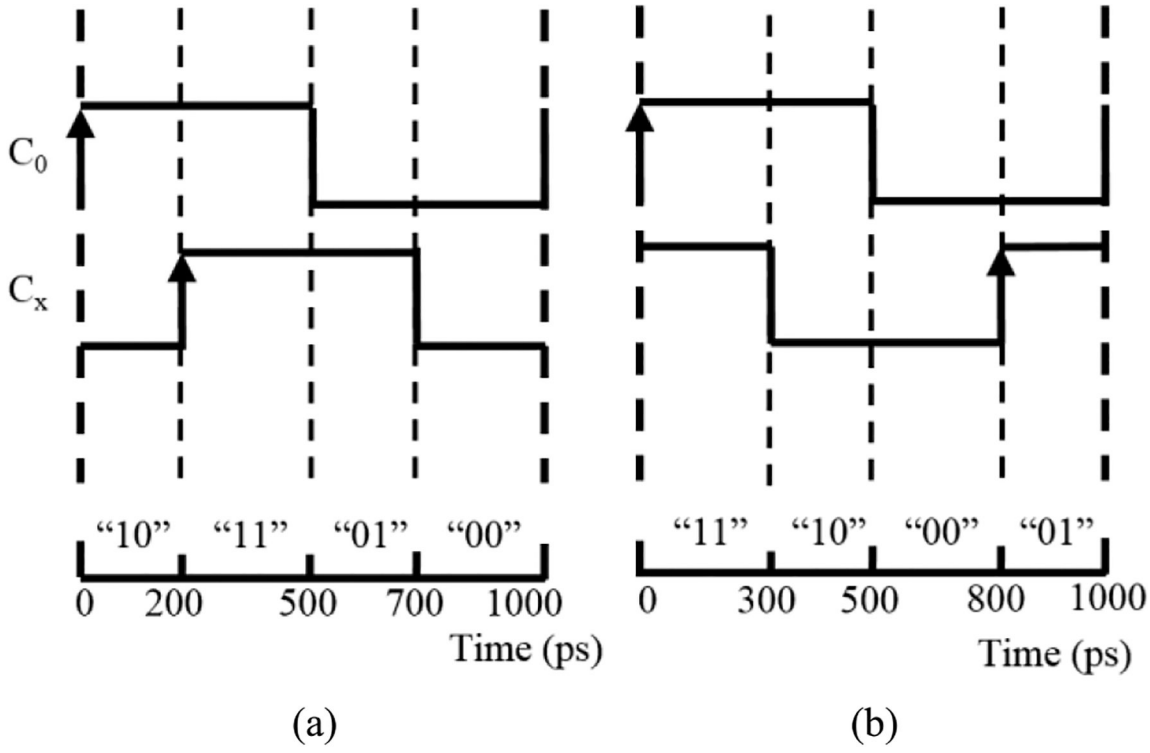


Fig. 7.

Two possible relationships between a reference clock C_0 and any other clock C_X generated by PSCG. (a) Situation where the rising edge of C_X arrives ahead of its falling edge in a clock cycle of C_0. (b) Inverse situation. The period of the clock is 1000 ps. The rising edge position of C_0 is set to 0 ps.

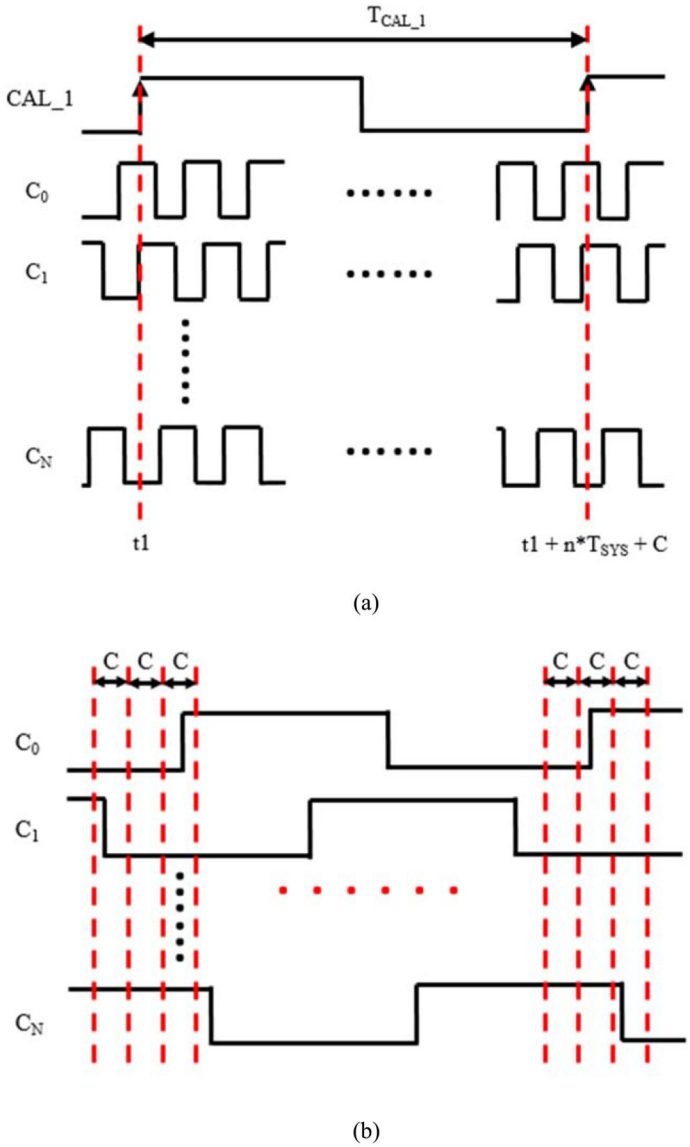


Fig. 8.

(a) Low-frequency signal CAL_1 is employed to latch the TDL registers. (b) CAL_1 is equivalent to a high-frequency signal whose period is C.

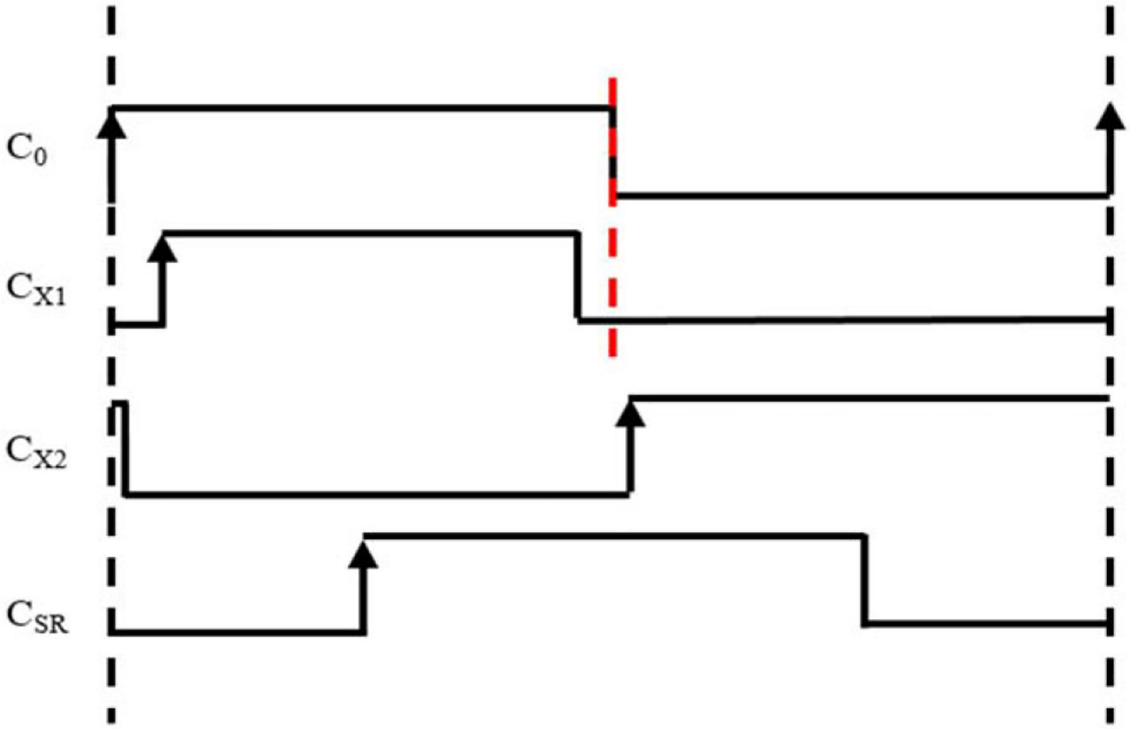


Fig. 9.

C_{X1} and C_{X2} represent clocks whose edge positions cannot be determined with C_0 as the reference. C_{SR}, therefore, serves as the new reference.

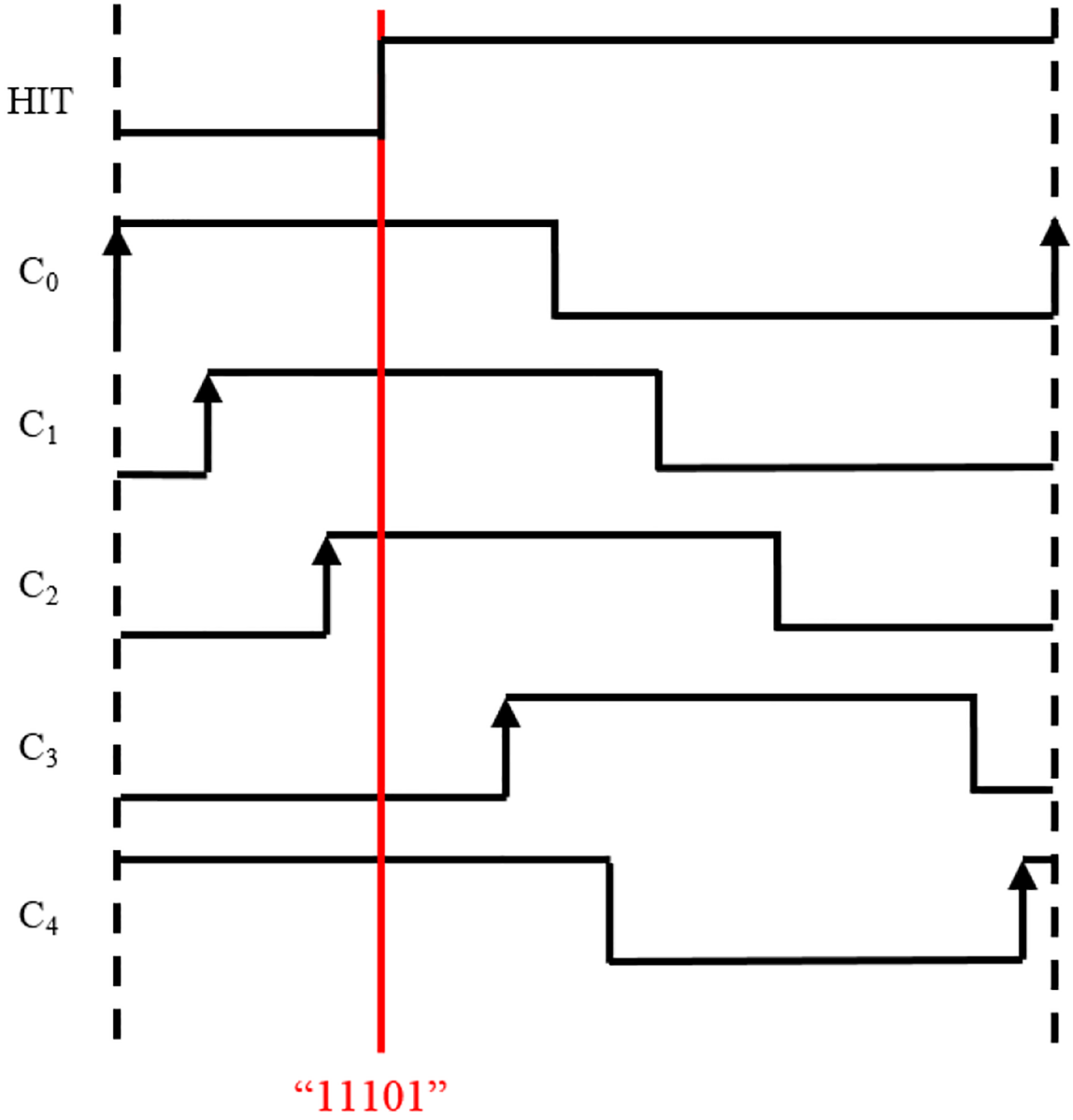


Fig. 10.
Five typical clocks and the hit signal are shown here to illustrate the decoding method.

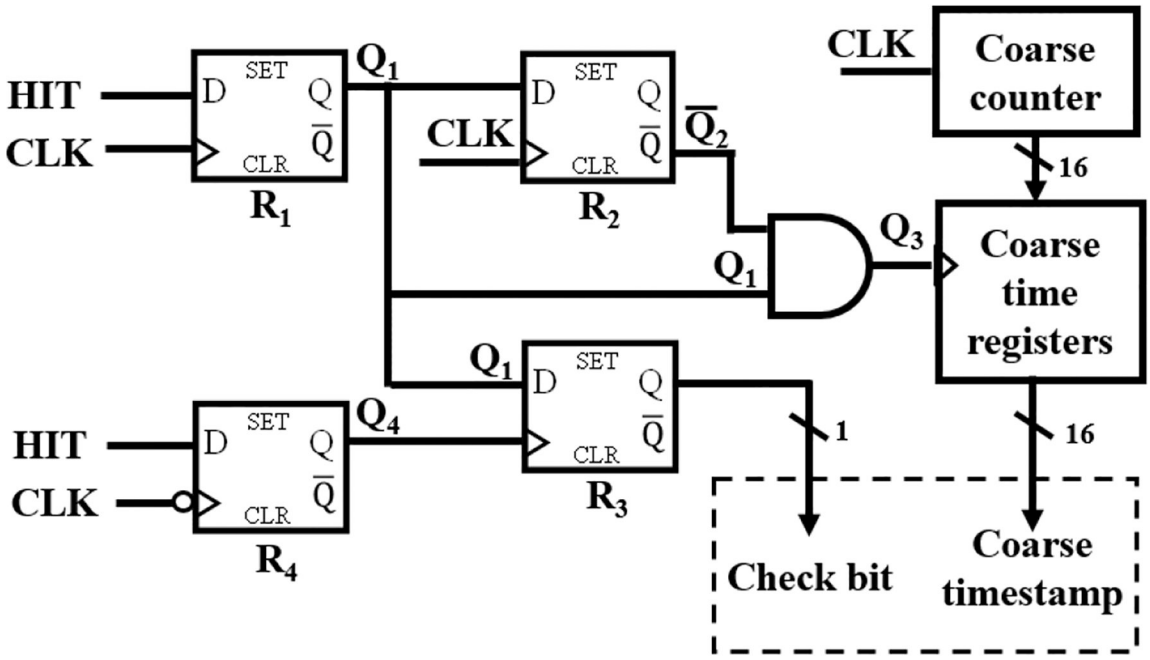


Fig. 11.
Diagram of the coarse time module.

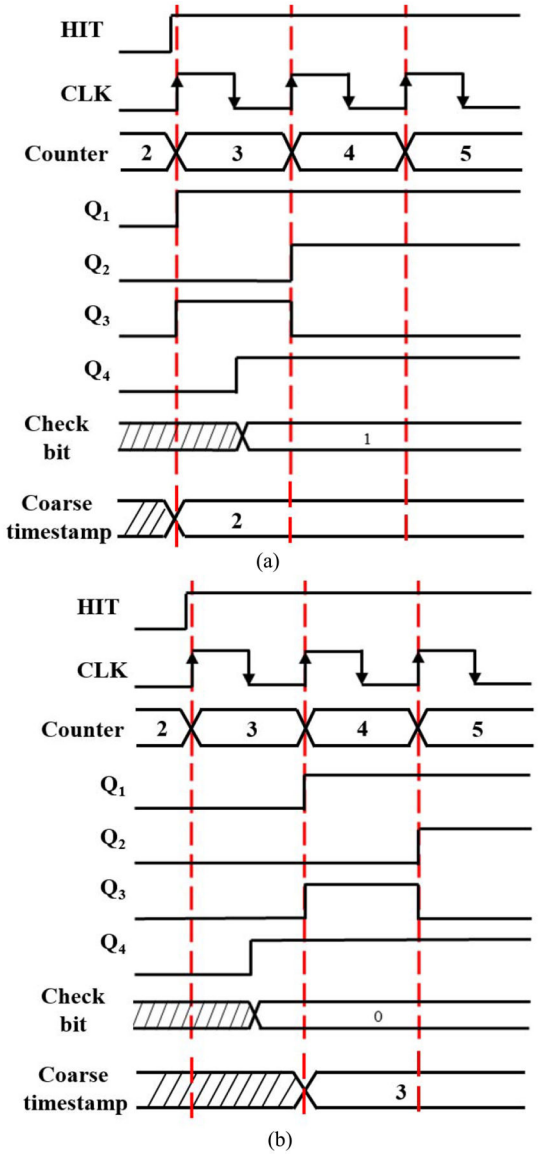


Fig. 12.

Two possible timing diagrams when the metastable state happens in register R_1. (a) Possible timing diagram I. (b) Possible timing diagram II.

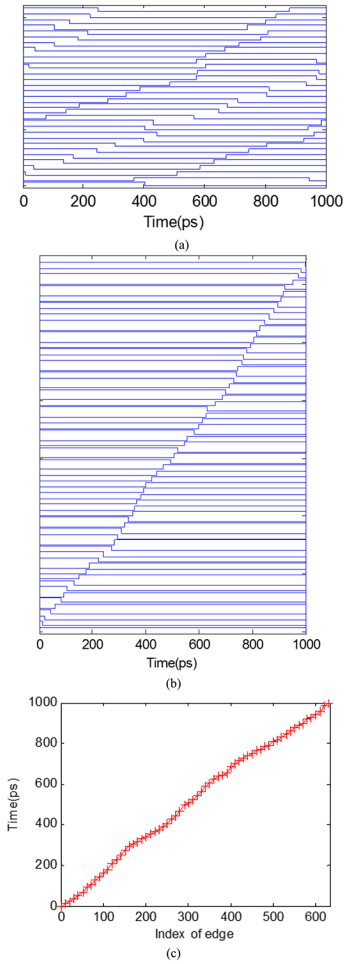


Fig. 13.

32 channels of clocks (a) before and (b) after the clock sorting operation. Note that (b) shows both the rising and falling edges. (c) Time delays for the 64 sorted edges.

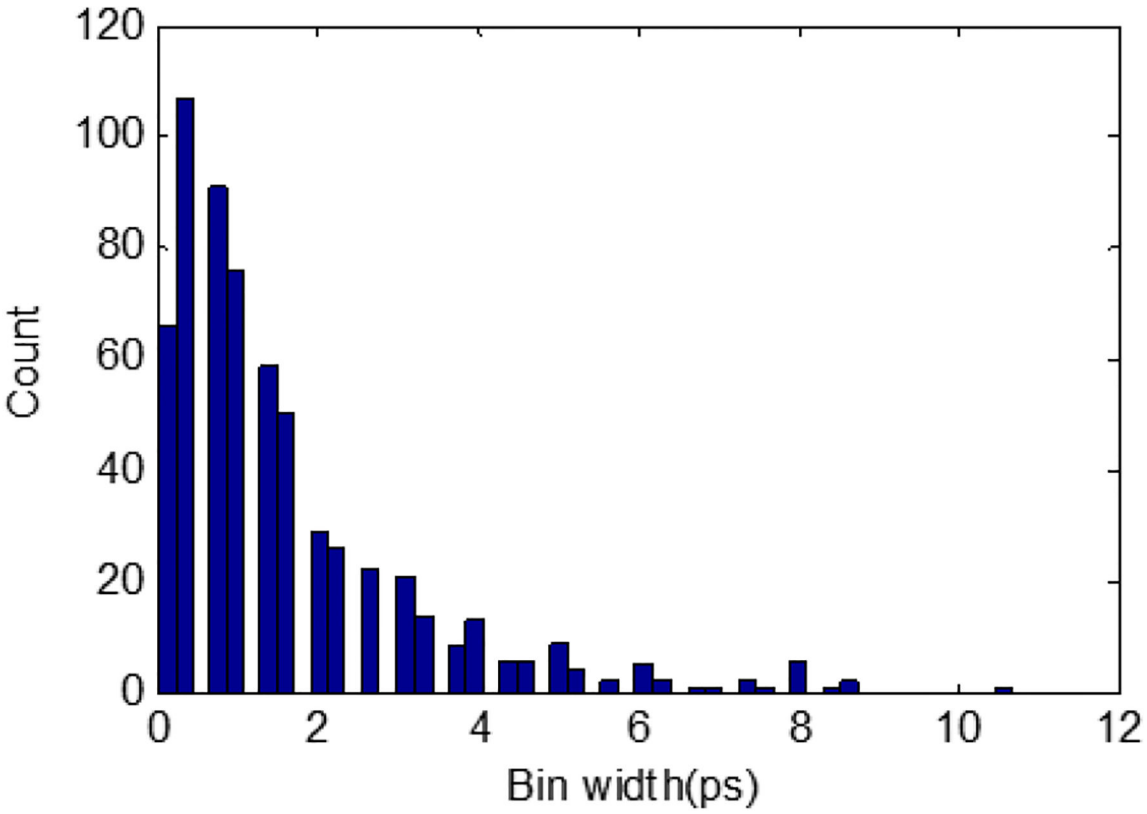


Fig. 14.
Histogram for bin width measured by the code density test method.

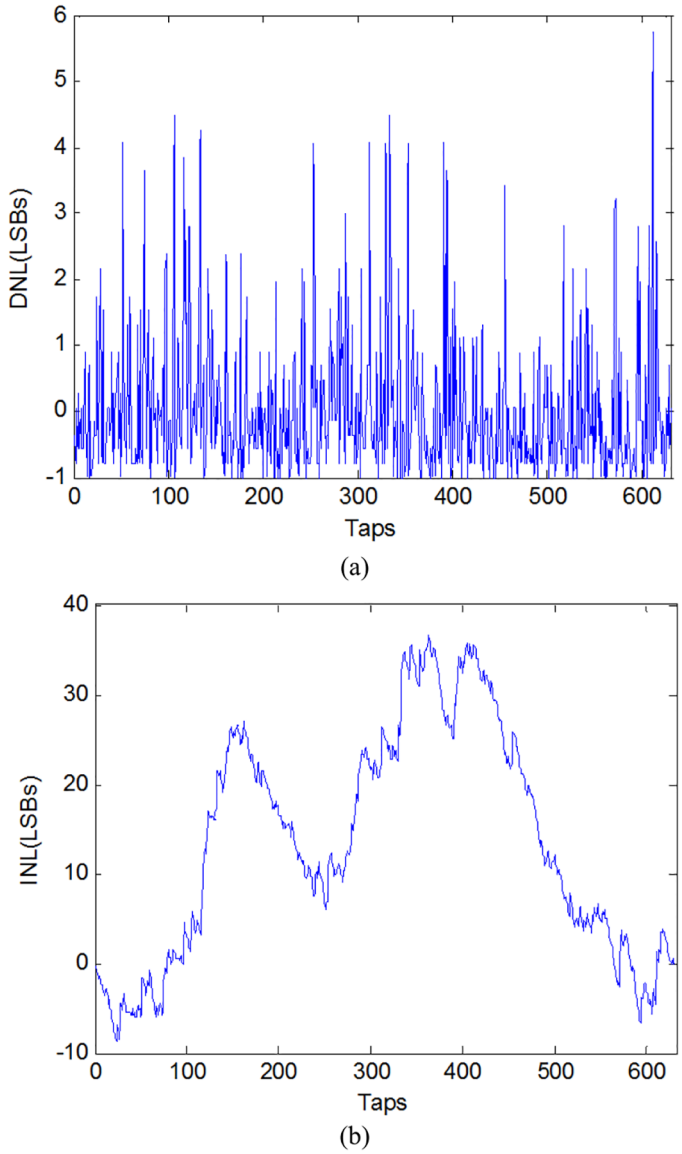


Fig. 15.
(a) DNL and (b) INL of the NUMP TDC.

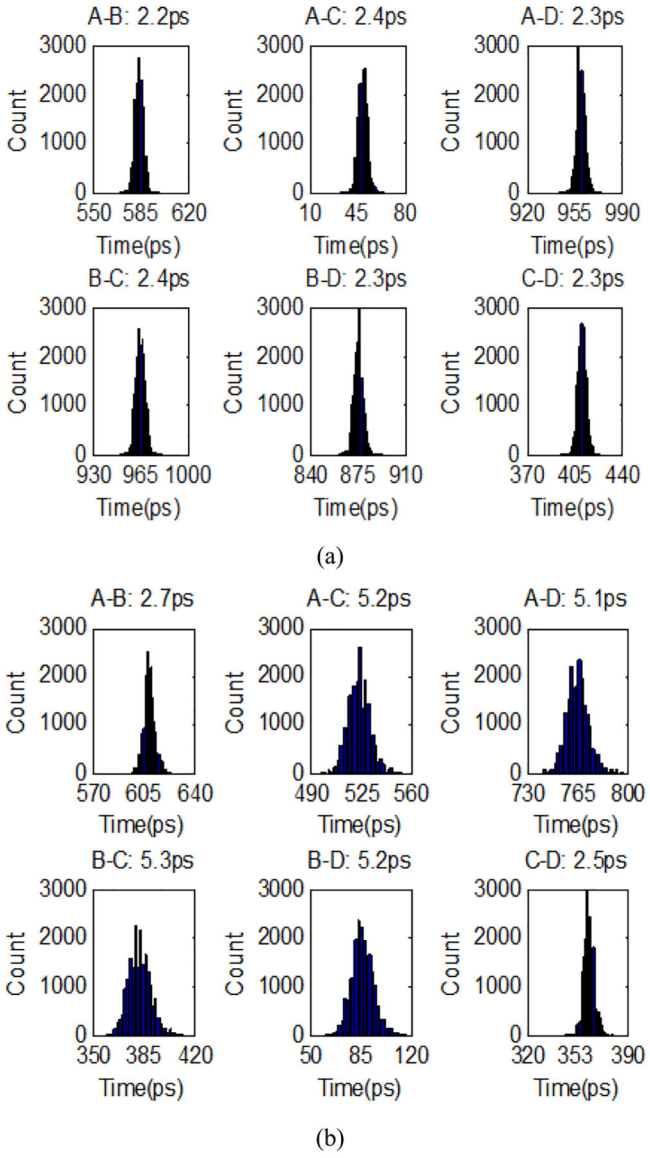


Fig. 16.

(a) Test results using internal pulses, (b) Test results using external pulses. Note that the rms values in the images are the average resolution for a single TDC channel (single-shot resolution), which are calculated by dividing the dual-channel results by √2.

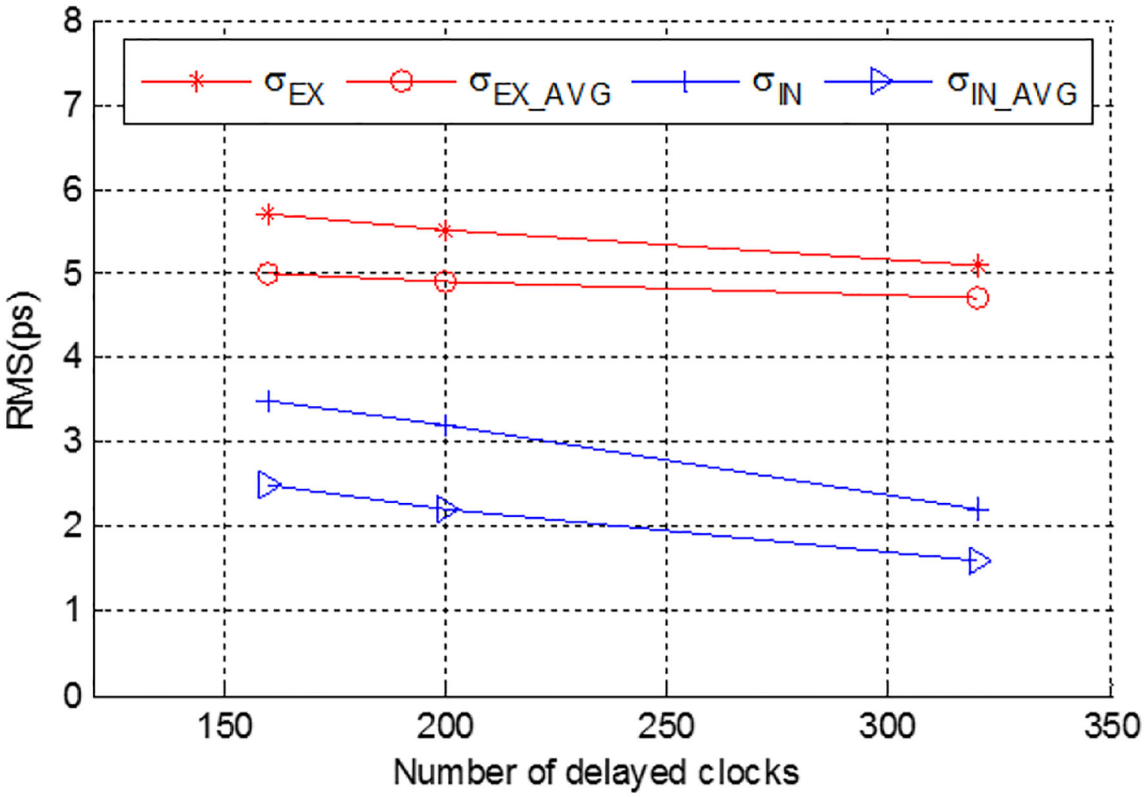


Fig. 17.

Relation between the timing resolutions of the NUMP TDCs and the number of delayed clocks. NUMP TDCs constructed with 160, 200, and 320 delayed clocks were tested using both internal and external pulses. We also measured the timing resolution constructed by averaging two NUMP TDCs. σ_{IN} and represent the measurements with internal and external pulses, respectively. Thus, σ_{IN_AVG} and σ_{EX_AVG} represent the results for the TDCs constructed by averaging two NUMP TDCs.

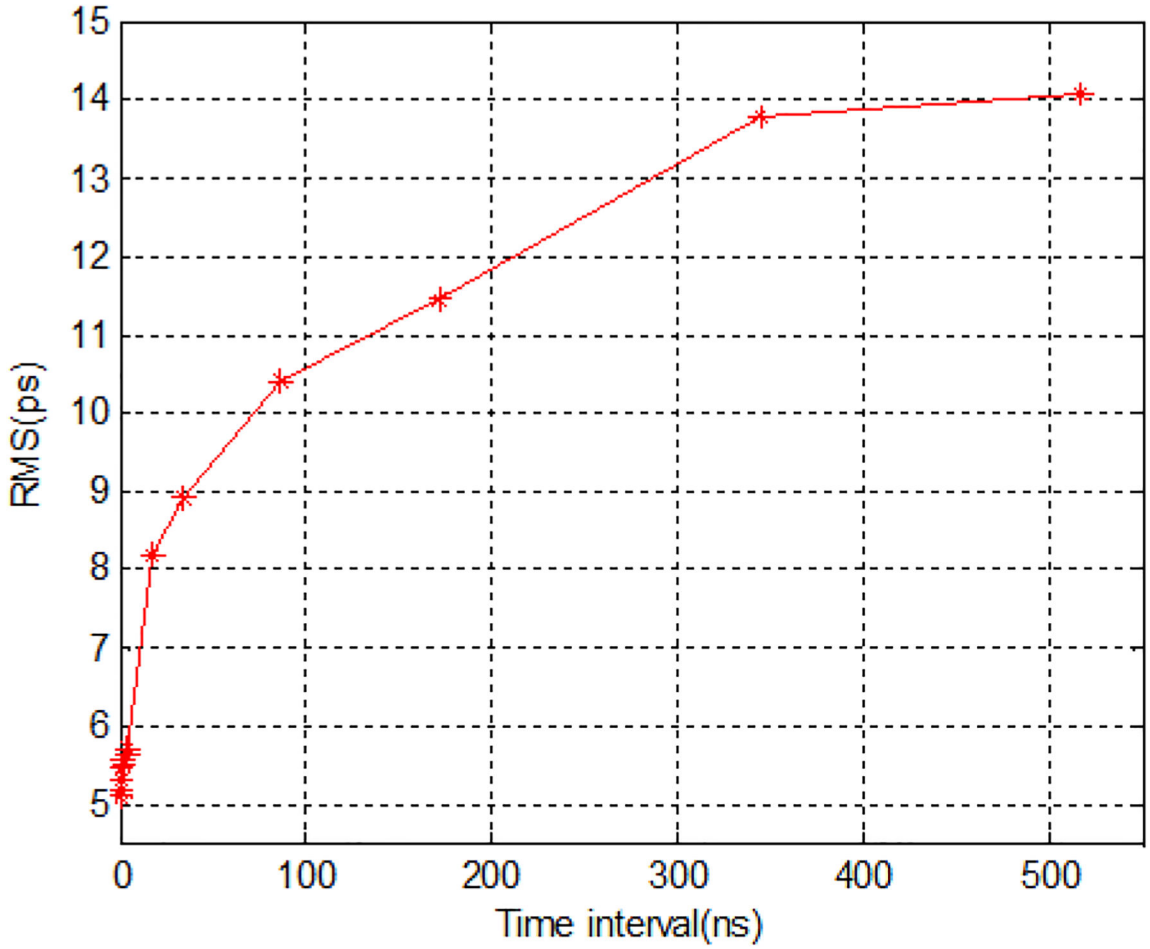


Fig. 18.
External pulse test results with different time intervals. The NUMP TDC was constructed using 400 clocks. The clock frequency is 500 MHz.

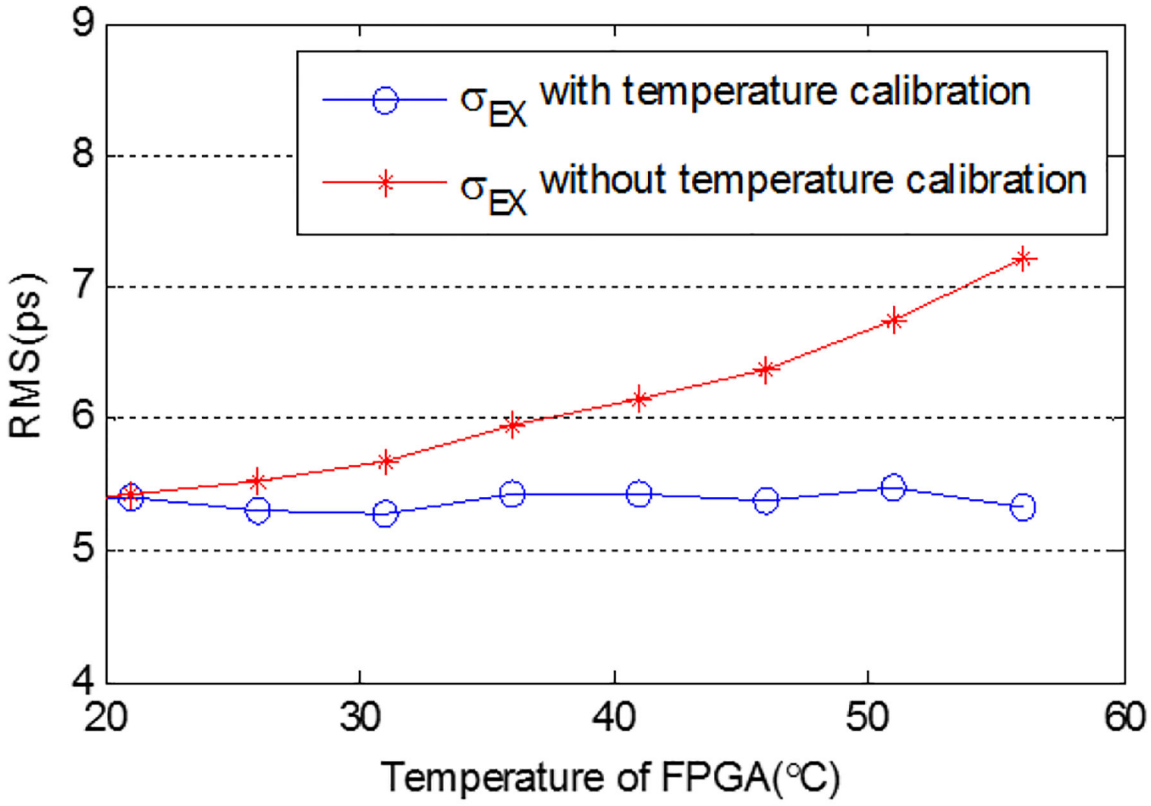


Fig. 19.

Test results of two TDC solutions at different ambient temperatures, one with temperature calibration and the other one without.

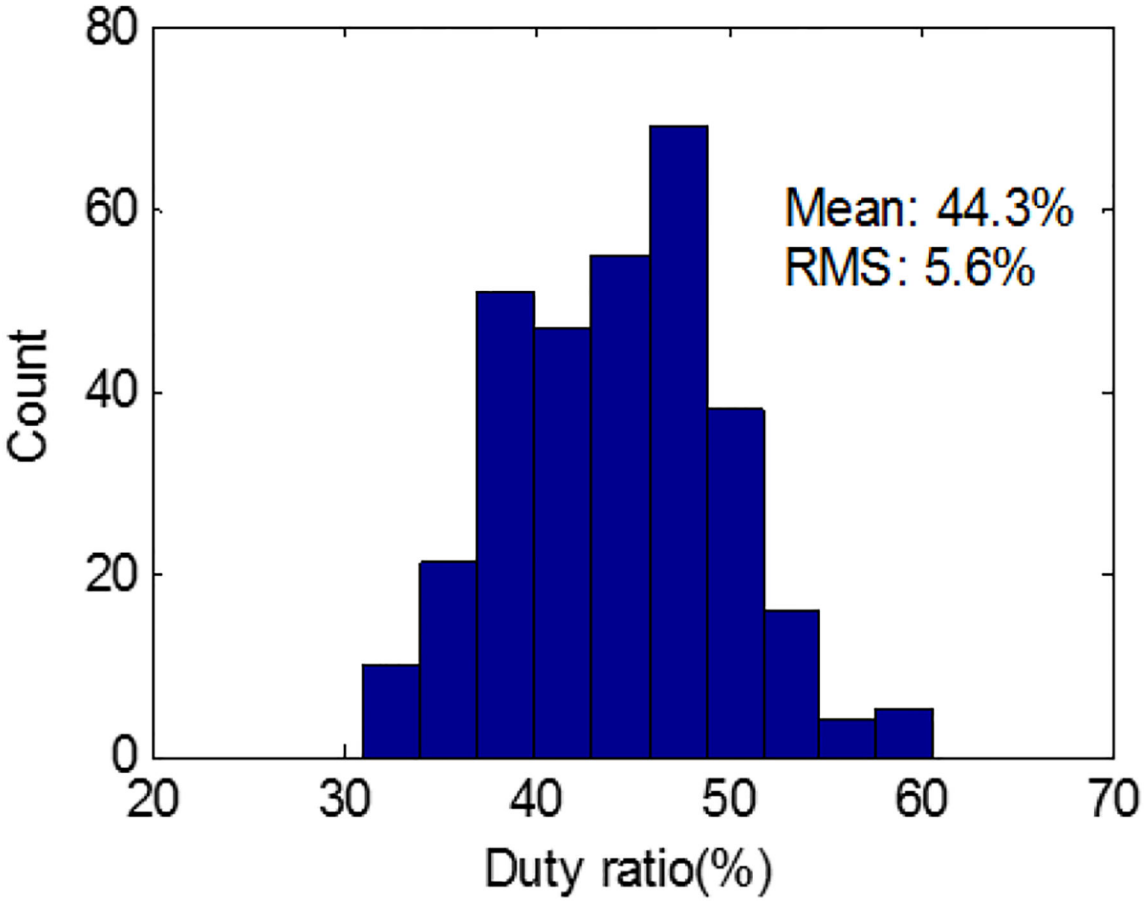


Fig. 20.

Histogram of the duty cycles of 320 delayed clocks. The duty cycles of the clocks can affect the performance of the conventional multiphase TDCs but does not affect NUMP TDCs at all.

TABLE I

LUT of the Clock Timing

Clock label	Time of the rising edge (ps)	Time of the falling edge (ps)
C_0	0	468
C_1	45	544
C_2	203	656
C_3	461	912
C_4	505	966

TABLE II

LUT of the Sorted Clocks

Index	Clock label	Rising edge or falling edge	Time (ps)
1	C_0	Rising	0
2	*C_1*	Rising	45
3	*C_2*	Rising	203
4	*C_3*	Rising	461
5	*C_0*	Falling	468
6	*C_4*	Falling	505
7	*C_1*	Falling	544
8	*C_2*	Falling	656
9	*C_3*	Falling	912
10	*C_4*	Rising	966