

High-precision time measurement electronics using the bandpass sampling method

Cite as: Rev. Sci. Instrum. 95, 014705 (2024); doi: 10.1063/5.0178312

Submitted: 26 September 2023 • Accepted: 28 December 2023 •

Published Online: 19 January 2024



View Online



Export Citation



CrossMark

Yichun Fan,^{1,2} Lei Zhao,^{1,2} Jiajun Qin,^{1,2,a} Jiaming Li,^{1,2} Shiya Huang,^{1,2} and Zhe Cao^{1,2}

AFFILIATIONS

¹ State Key Laboratory of Particle Detection and Electronics, University of Science and Technology of China, Hefei 230026, China

² Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China

^aAuthor to whom correspondence should be addressed: jjqin@ustc.edu.cn

ABSTRACT

Waveform digitization at sampling rates up to several giga-samples per second is one of the approaches to achieve high-precision time measurements. In recent years, achieving high precision at lower sampling rates has emerged as a significant research topic. In this article, we focus on time measurement electronics, in which the bandpass sampling method is applied to obtain high precision at a sampling rate of roughly 100 MSps. An analog front-end circuit is designed, in which the input pulse signal is bandpass-filtered and amplified before sampling. A pipelined real-time time extraction algorithm is designed using the techniques of interpolation and cross correlation. A 1024-point fast Fourier transform algorithm is adopted to implement the cross correlation operation. Four time measurement channels are implemented in a mid-range field-programmable gate array, and the measurement rate of 116 kHz is achieved. Tests are conducted to evaluate the performance of the timing system. The typical RMS precision is better than 0.4 ps.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0178312>

I. INTRODUCTION

Waveform digitization can be utilized to achieve high-precision time measurements.^{1–6} To attain high precision, sampling rates of several giga-samples per second are commonly employed, which bring high requirements to the core components and the peripheral circuits in the electronics system and increase the complexity of the system. In recent years, research studies have been conducted to achieve a precision of around 2 ps at sampling rates of several hundreds of mega-samples per second.^{7,8} Moreover, it is still expected to achieve a higher precision at lower sampling rates. The work in this paper is actually invoked by Pánek.

Pánek proposed a time interval measurement method based on the Surface Acoustic Wave (SAW) filter excitation.^{9–12} In his research, an exciter circuit would produce a uniform-shaped pulse with a fast rise time and a short pulse width in response to an input pulse.¹³ A SAW filter converts the excited pulse into a bandpass signal. After being amplified, the bandpass signal is sampled and digitized. After the above process is finished, the exciter generates a second fast pulse that is synchronized to the reference clock, which will also be processed by the SAW filter and then digitized. The time interval between these two signals is obtained by a time

extraction algorithm. A high time measurement precision of around 0.7 ps is achieved.¹⁴ However, the exciter circuits in the design only retain the time information of the input pulses, and moreover, the measurement rate is only 10 kHz.¹⁴

In this work, we follow the basic idea of waveform digitization but make modifications. First, we apply the analog bandpass filtering directly to the input pulse, and therefore, amplitude information is also kept in the digitized signal. Pulse signals with a fixed shape are concerned in this paper. Second, we design a pipelined time extraction algorithm using the techniques of interpolation and cross correlation, and the measurement rate is improved up to 116 kHz. A time precision better than 0.4 ps is successfully achieved in this work.

The structure of this article is as follows: In Sec. II, the structure of the time measurement electronics will be covered. The design of the analog front-end circuit and the design of the time extraction algorithm will be discussed in Secs. III and IV correspondingly. In Sec. V, we will build a simulation model and conduct a simulation of the dominant parameter of time precision. In Sec. VI, tests will be conducted to evaluate the performance of the time measurement system.

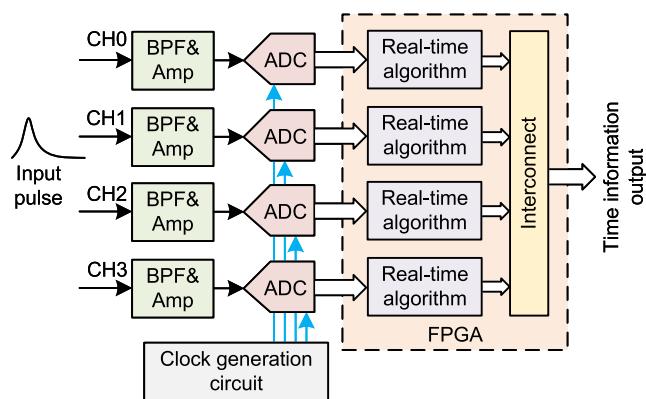


FIG. 1. Block diagram of the time measurement electronics.

II. STRUCTURE OF THE TIME MEASUREMENT ELECTRONICS

The structure of the time measurement electronics is shown in Fig. 1. In each time measurement channel, the input pulse is directly sent to the bandpass filtering and amplifying circuit, of which the output bandpass signal is an oscillation in the time domain. The bandpass signal is sampled by an analog-to-digital converter (ADC), where the undersampling technique is used. The sampling clock of ADCs is generated by a clock generation circuit and simultaneously distributed to each channel. The samples are received by a single field-programmable gate array (FPGA), which runs the real-time time extraction algorithm. The time information of the pulses is output via a single data transfer interface. Four time measurement channels are contained in one electronics module.

Undersampling, also known as bandpass sampling, is a technique to digitize the analog signal at a sampling rate below its Nyquist rate, i.e., twice the maximum frequency, which is commonly used in the field of radio frequency (RF) communications¹⁵ and the field of accelerators.¹⁶ There are several constraints while using the technique. First, the signal to be measured needs to have a narrow frequency band, which is typically conditioned by an analog bandpass filter. Second, the frequency band of the analog signal needs to lie entirely within one of the Nyquist zones of the ADC to ensure that there is no overlapping of frequency components. In other words, the sampling rate f_s needs to satisfy the following inequalities:

$$(k - 1)\frac{f_s}{2} < f_{c,low} < f_{c,high} < k\frac{f_s}{2}$$

for

$$k = 1, 2, \dots, \left\lfloor \frac{f_{c,high}}{f_{c,high} - f_{c,low}} \right\rfloor, \quad (1)$$

where k is an integer that represents the k th Nyquist zone of the ADC and $f_{c,low}$ and $f_{c,high}$ are the cutoff frequencies of the bandpass filter. Third, the bandwidth of amplifiers and ADCs needs to be higher than the maximum frequency $f_{c,high}$.

In this work, the input pulses are bandpass-filtered. Therefore, undersampling can be used when selecting an appropriate

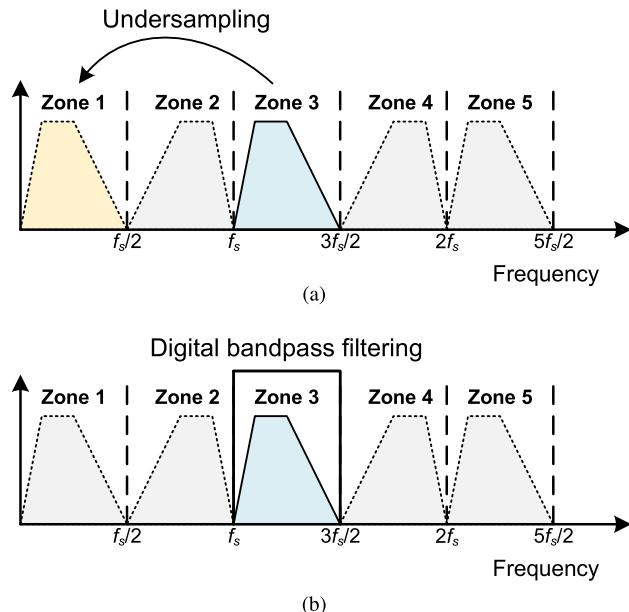


FIG. 2. An example of the frequency spectrum while undersampling and interpolation filtering. The bandpass signal lies in the third Nyquist zone. The frequency is shown in the analog domain. (a) Undersampling. (b) Digital bandpass filtering.

sampling rate and designing an analog front-end circuit with a high bandwidth.

After the bandpass signal is undersampled, the frequency spectrum will be aliased to the first Nyquist zone, as shown in Fig. 2(a). It is converted into a low-frequency signal. An interpolation filter¹⁷ can be used to recover the bandpass signal, including two steps of zero-stuffing and digital bandpass filtering. For an interpolation factor of M_1 , the zero-stuffing step involves inserting $(M_1 - 1)$ zeros between each pair of adjacent samples, which equivalently improves the sampling rate. The digital bandpass filtering step involves filtering out unnecessary low-frequency and high-frequency components, aiming to restore the analog signal's spectrum, as shown in Fig. 2(b). The passband of the digital filter should match the analog signal's frequency band. It should be noted that interpolation filtering can restore the bandpass signal only when Eq. (1) is satisfied.

III. DESIGN OF THE ANALOG FRONT-END CIRCUIT

The block diagram of the analog front-end circuit is shown in Fig. 3. The bandpass filtering and amplifying circuit consists of two bandpass filters with the same frequency response (BPF1 and BPF2) and an amplifier (Amp1). When the input pulse is partially capacitive coupling to the output of BPF1, the amplitude of the feedthrough pulse signal can be higher than the bandpass signal. To reduce the feedthrough pulse and avoid the overlapping of frequency components while using the undersampling technique, BPF2 is needed. In addition, BPF2 is applied to reduce the noise and harmonics of the amplifier circuit.

Since the input to the ADC is differential, a balun is used to convert the bandpass signal to a differential form. The reference clock of the phase-locked loop (PLL) can be selected from an external input

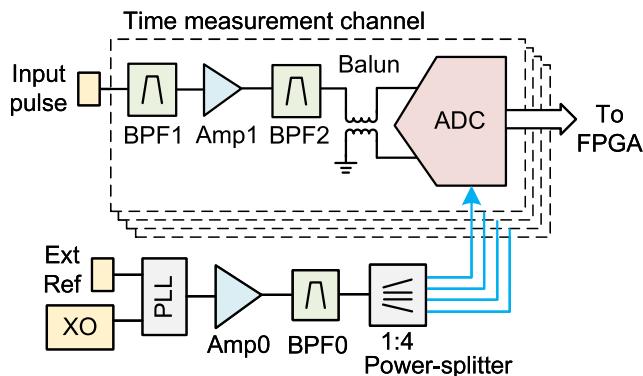


FIG. 3. Block diagram of the analog front-end circuit.

or generated by an onboard crystal oscillator. To minimize clock jitter, a bandpass filter (BPF0) is employed with a center frequency that closely aligns with the sampling rate.

In the design of the analog front-end circuit, several parameters should be considered, including the center frequency and bandwidth of the bandpass filter, the gain of the amplifier, and the sampling rate of the ADC. According to Pánek's analysis,¹¹ a higher center frequency for the bandpass filter has the potential to yield improved precision. However, it is essential to note that increasing the passband frequency is accompanied by a reduction in the signal-to-noise ratio. Therefore, the bandpass filter should be chosen according to the frequency spectrum of the input pulse signal. In addition, the gain of the amplifier should be high enough to match the amplitude of the bandpass signal and the input range of the ADC, and the sampling rate needs to be chosen to satisfy Eq. (1).

In this work, commercial off-the-shelf components are employed to simplify the system design. The RF360 B4233 SAW bandpass filter with a center frequency of 390 MHz and a bandwidth of 20 MHz is utilized. For a pulse signal with a rise time of around 1.2 ns, the time-domain waveform, the frequency spectrum, and the position of the filter passband within the spectrum are shown in Fig. 4. Two stages of low noise amplifiers, each being Mini-Circuits PMA-5452+ with a gain of 23 dB and a passband ranging from 50 MHz to 6 GHz, are cascaded. The ADC in the design has a 16-bit resolution and an analog bandwidth higher than 600 MHz. The 122.88 Msps sampling rate is chosen according to the frequency of commercial crystal oscillators, and the passband of the filter lies in the seventh Nyquist zone of the ADC. In addition, Crystek CCPD-575 is used for the onboard crystal oscillator, and Mini-Circuits BPF-A122+ is used for BPF0.

A simulation is conducted to evaluate the functionality of the analog front-end circuit. The pulse signal as shown in Fig. 4(a) is generated, and the output of BPF1 is shown in Fig. 5, where the feedthrough pulse has a high amplitude and a wide frequency band, and the bandpass signal lasts for hundreds of nanoseconds with a narrow frequency band. After amplification and BPF2, the shape of the output waveform is shown in Fig. 6. It shows that the feedthrough pulse is effectively reduced in comparison with the bandpass signal.

Compared to directly digitizing the fast pulse, converting it into a bandpass signal with an extended duration provides more

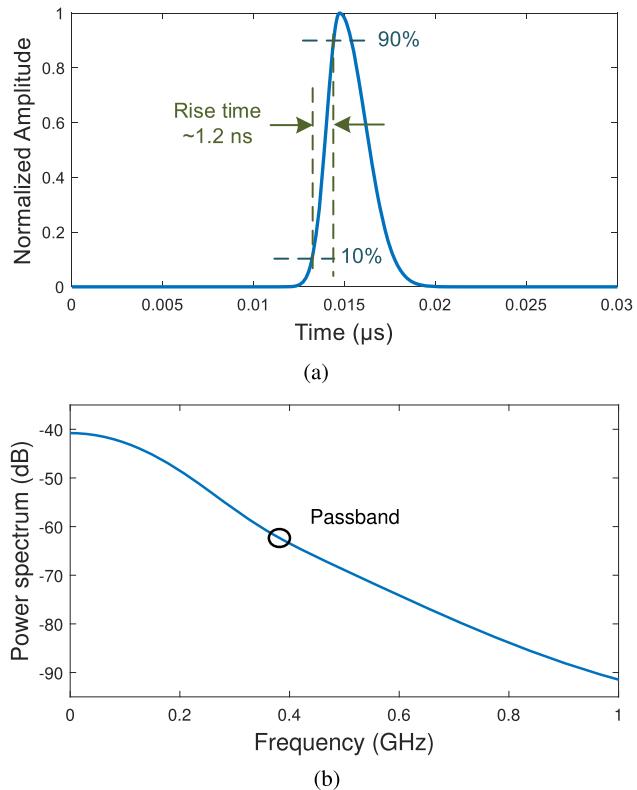


FIG. 4. A pulse signal with 1.2 ns rise time and the corresponding frequency spectrum. (a) Time domain. (b) Frequency domain.

sampling points. In combination with a suitable time extraction algorithm, it could be more feasible to achieve higher time precision.

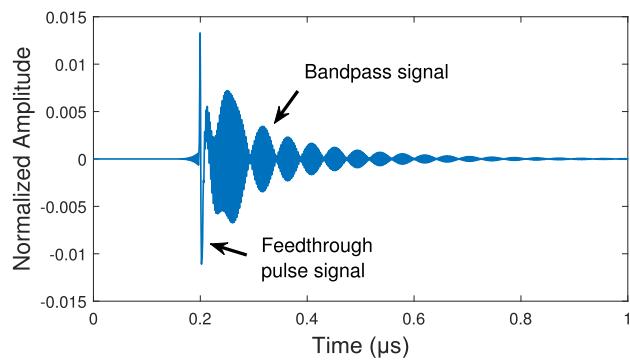
IV. DESIGN OF THE TIME EXTRACTION ALGORITHM

A time extraction algorithm is used to calculate the arrival time of each test pulse. Since the bandpass signal oscillates in the time domain, conventional time pickoff methods, such as digital leading edge discrimination and digital constant fraction discrimination, are not applicable in this context. This is where the cross correlation timing method becomes particularly useful, as it can effectively handle signals that oscillate in the time domain. The method calculates the time interval between two pulses by identifying the position of the maximum value within the cross correlation function. During the timing process, a large number of sampling points are used, which can achieve an equivalent effect of averaging.¹⁸

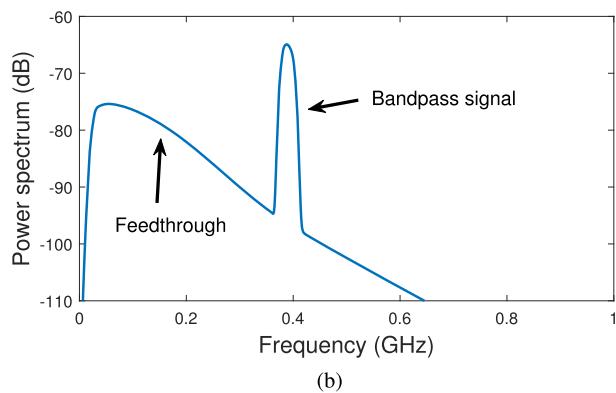
The cross correlation function is defined as follows. At each position indexed by n , the function $x_{test}[i+n]$ is subjected to pointwise multiplication with $x_{ref}[i]$ and the results are then accumulated using the following equation:¹⁹

$$x_{CORR}[n] = \sum_{i=-\infty}^{+\infty} x_{test}[i+n]x_{ref}[i]. \quad (2)$$

In this work, the test signal $x_{test}[n]$ and the reference signal $x_{ref}[n]$ share the same pulse shape. The reference signal denotes the



(a)



(b)

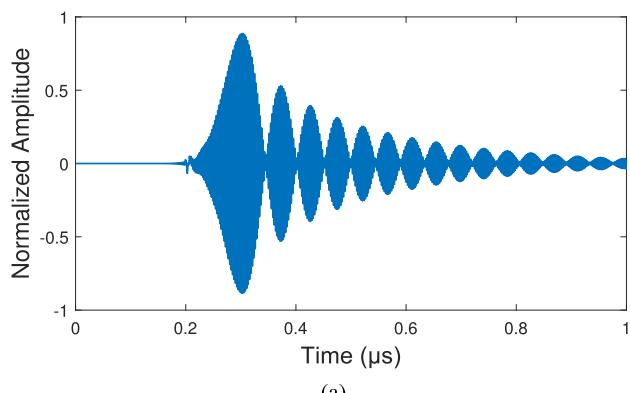
FIG. 5. The feedthrough pulse signal and the bandpass signal after the first bandpass filter (BPF1). The amplitude of the feedthrough is set to 2.7% of the input pulse. (a) Time domain. (b) Frequency domain.

beginning of time, which is equivalent to the start signal in a time-to-digital converter (TDC). All test signals could share the same reference signal. When the position n corresponds to the time delay between the two signals, the shifted test signal aligns with the reference signal, and it is at this point that the cross correlation function attains its maximum value. Since the arrival time of the reference signal referring to the sampling clock is fixed and could be calibrated, the arrival time of each test pulse could then be obtained.

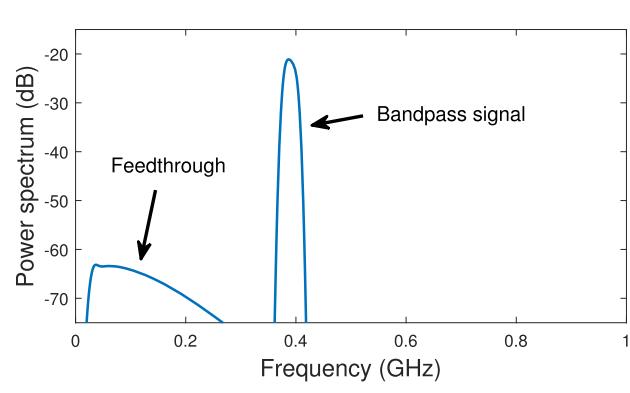
When computing the cross correlation function directly using Eq. (2), for a test signal comprising N effective data points, the number of required multiplication and addition operations scales with the order of $O(N^2)$. By using the Fast Fourier Transform (FFT) algorithm,¹⁷ the computational complexity can be reduced to the order of $O(N \log_2 N)$.

The cross correlation of two signals can be represented by the convolution of the first signal and the time reversal of the second signal, as follows:

$$\begin{aligned} x_{\text{CORR}}[n] &= \sum_{i=-\infty}^{+\infty} x_{\text{test}}[i+n] x'_{\text{ref}}[-i], \\ &= \sum_{i=-\infty}^{+\infty} x_{\text{test}}[n-i] x'_{\text{ref}}[i], \\ &= x_{\text{test}}[n] * x'_{\text{ref}}[n], \end{aligned} \quad (3)$$



(a)



(b)

FIG. 6. The feedthrough signal and the bandpass signal after the second bandpass filter (BPF2). (a) Time domain. (b) Frequency domain.

where $x'_{\text{ref}}[n]$ is the time reversal of the reference signal and the “*” operator represents the convolution of two signals. Therefore, the cross correlation can be calculated using the FFT algorithm as

$$\begin{aligned} X_{\text{test}}[f] &= \text{FFT}(x_{\text{test}}[n]), \\ X_{\text{ref}}[f] &= \text{FFT}(x_{\text{ref}}[n]), \\ X_{\text{CORR}}[f] &= X_{\text{test}}[f] \cdot X_{\text{ref}}^*[f], \\ x_{\text{CORR}}[n] &= \text{IFFT}(X_{\text{CORR}}[f]), \end{aligned} \quad (4)$$

where $X_{\text{ref}}^*[f]$ is the complex conjugate of the frequency spectrum of the reference signal. It should be noted that although the FFT algorithm is used in the calculation process, the cross correlation operation still represents an analysis of the signal's time-domain characteristics.

The interpolation technique is employed in two reasons. First, the bandpass signal is digitized using the undersampling technique so that it is equivalently converted into a low-frequency signal. The interpolation technique is used in this work to recover the high-frequency bandpass signal. Second, the time resolution of the cross correlation timing method is determined by the least time interval between the samples. The interpolation is used to improve the time resolution.

The relationship between the least significant bit (*LSB*) and the overall interpolation factor *M* follows

$$LSB = \frac{T_{clk}}{M}, \quad (5)$$

where T_{clk} is the sampling period, which equals 8.138 ns in this work.

To accurately determine the necessary value of *M*, it is important to consider the impact of time interval quantization on time precision (*RMS*), which is represented by²⁰

$$RMS = LSB \cdot \sqrt{\frac{\Delta T_0}{LSB} \left(1 - \frac{\Delta T_0}{LSB}\right)}. \quad (6)$$

Here, $\Delta T_0 = T_0\%T_{clk}$ represents the remainder of the time interval T_0 relative to the sampling period T_{clk} , where T_0 is the time interval intended for measurement. The maximum potential value of *RMS* equals $LSB/2$.

To ensure that the contribution of time interval quantization to time precision is much smaller than the contribution of electronic noise and other factors, an overall interpolation factor *M* of 2^{16} is chosen, and the corresponding *LSB* is 124 fs.

A basic idea is to complete the interpolation process before the cross correlation operation. For a signal with a length of t_{win} , if the required time resolution unit is *LSB*, the number of data points *N* for cross correlation operation will be t_{win}/LSB . For example, when t_{win} is 260 ns and *LSB* is 124 fs, *N* will be 2×10^6 , which will consume extremely high computing resources even if the FFT algorithm is employed.

To reduce the resource consumption and improve the measurement rate based on the FPGA, in this work, the interpolation process is carried out in two steps. The first-step interpolation is done before the cross correlation calculation, applying directly to the samples. Then, the second-step interpolation is applied to the output of the cross correlation calculation. The interpolation factors corresponding to the two steps are M_1 and M_2 , respectively, where M_1 is smaller and M_2 will be implemented in multiple stages. The overall interpolation factor *M* is the product of M_1 and M_2 , as shown in the following equation:

$$M = M_1 \cdot M_2. \quad (7)$$

The structure of the time extraction algorithm is shown in Fig. 7. Since the reference signal and the test signal would not appear at the same time, computing resources for the windowing process, the first-step interpolation, and a part of the cross correlation process could be shared.

A. First-step windowing

While receiving the data stream from the ADC, a window with finite length is applied for each bandpass-filtered pulse. A predefined threshold is set to detect the arrival of each pulse. When the absolute value of the sample exceeds the threshold, the value n_0 of the clock counter is recorded, which is the coarse time of the pulse signal, and adjacent N_{win0} samples around n_0 are written into a first in, first out (FIFO). The write clock of the FIFO is synchronized to the sampling clock of ADCs, and the read clock is the clock of back-end time extraction logic.

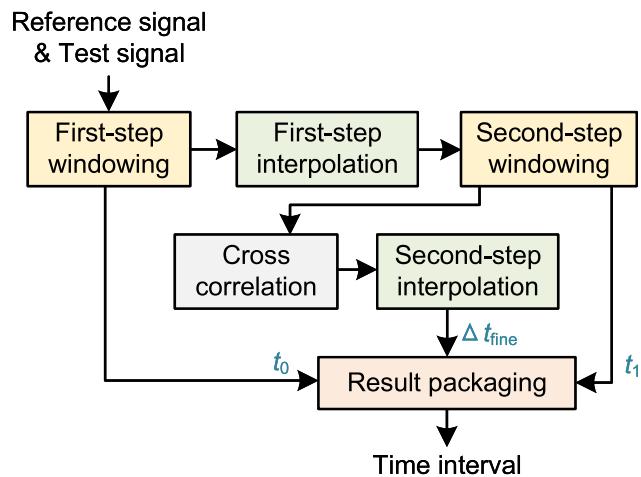


FIG. 7. Block diagram of the time extraction algorithm.

B. First-step interpolation

After the first-step windowing, a first-step interpolation is implemented by zero-stuffing followed by digital bandpass filtering.

To recover high-frequency bandpass signals in the time domain, the interpolation factor M_1 in the first-step interpolation needs to be higher than the order of the Nyquist zone where the bandpass signal is situated, meaning that it should be greater than 7. For the convenience of encoding the time measurement results, the interpolation factors are chosen to be integer powers of 2. Specifically, to achieve a sufficient time precision in this paper, M_1 is set to 16.

In order to adapt to the pipelined algorithm implementation structure and achieve a high throughput, the bandpass filter is chosen to be the finite impulse response (FIR) filter. To design the FIR filter, the Kaiser window¹⁷ is employed to specifically adjust and optimize the filter's sidelobe attenuation and transition bandwidth. The expression for the Kaiser window is given by

$$w_1[n] = \begin{cases} \frac{I_0\left(\beta_1 \sqrt{1 - \left(\frac{n-N_{filt1}/2}{N_{filt1}/2}\right)^2}\right)}{I_0(\beta_1)}, & \text{for } n = 0, 1, 2, \dots, N_{filt1}, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $I_0(*)$ is the zeroth-order modified Bessel function of the first kind, β_1 is the shape parameter of the Kaiser window, and N_{filt1} is the order of the filter.

The impulse response of an ideal digital bandpass FIR filter is given by the following expression:

$$h_{dl}[n] = \frac{\sin\left(2\pi f_H\left(n - \frac{N_{filt1}}{2}\right)\right) - \sin\left(2\pi f_L\left(n - \frac{N_{filt1}}{2}\right)\right)}{\pi\left(n - \frac{N_{filt1}}{2}\right)} \quad \text{for all integer } n, \quad (9)$$

where f_H and f_L are the upper and lower cutoff frequencies of the bandpass filter, respectively.

The core of the Kaiser window method in designing FIR filters lies in multiplying the ideal filter's impulse response with the Kaiser window function. This process is represented as

$$h_1[n] = h_{d1}[n] \cdot w_1[n] \quad \text{for } n = 0, 1, 2, \dots, N_{filt1}. \quad (10)$$

The order N_{filt1} determines the length of the filter's unit impulse response. A higher order corresponds to a narrower transition band but also requires more hardware resources.

To recover the bandpass signal, the passband of the FIR filter is set to be in the seventh Nyquist zone of the ADC. In particular, the center frequency of the FIR filter is set to 390 MHz, the same as the bandpass filter in the front-end circuit. The response of the filter is determined by the bandwidth f_{BW} , the order N_{filt1} , and the shape parameter β_1 of the Kaiser window, which will be optimized by the genetic algorithm (GA)^{21–23} together with the parameters in the second-step interpolation.

The samples after the first-step windowing and the signal after the first-step interpolation are shown in Fig. 8. The optimized parameter f_{BW} is slightly smaller than the bandwidth of the analog bandpass filter; hence, the interpolated signal does not strictly pass through each sampling point.

C. Second-step windowing

A second-step windowing is adopted to reduce the number of FFT points in the calculation of cross correlation. In order to ensure the consistency of the pulse shape after windowing, the starting position of the window is set at a fixed number of data points prior to the maximum absolute value. The length of the window is denoted as N_{win1} , and the index of the first data point is denoted as n_1 . The number of FFT points equals $2N_{win1}$ to avoid aliasing in the time domain. In this work, N_{win1} is set to 512, and the start and end points of the second-step windowing are shown in Fig. 8.

D. Cross correlation

The computation structure of the cross correlation is shown in Fig. 9. The frequency spectrum of samples after the second-step windowing is calculated in the FFT module. The frequency spectrum of the reference signal $X_{ref}[f]$ is calculated first, and the result is conjugated and stored in the block RAM. For each test signal, after

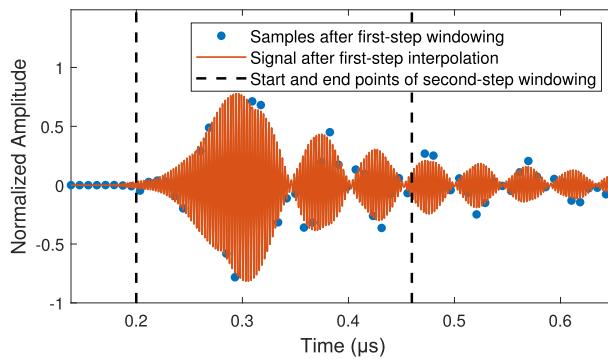


FIG. 8. The samples after the first-step windowing, the signal after the first-step interpolation, and the start and end points of the second-step windowing.

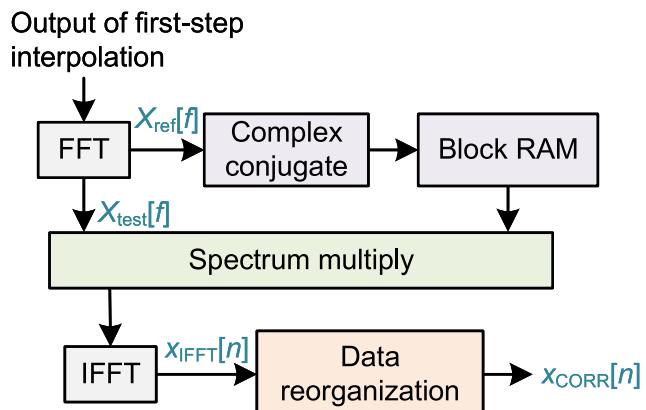


FIG. 9. Block diagram of the cross correlation calculation.

computing the frequency spectrum $X_{test}[f]$, the spectra of two signals are multiplied. The product is sent to the inverse FFT (IFFT) module, and the cross correlation is obtained, as shown in Fig. 10(a). In order to further locate the maximum value, the output of the cross correlation is shifted as follows:

$$x_{CORR}[n] = \begin{cases} x_{IFFT}[n + N_{win1} + 1] & \text{for } n = 0, 1, \dots, N_{win1} - 2, \\ x_{IFFT}[n - N_{win1} + 1] & \text{for } n = N_{win1} - 1, N_{win1}, \dots, 2N_{win1} - 2. \end{cases} \quad (11)$$

The shifted cross correlation function $x_{CORR}[n]$ is shown in Fig. 10(b).

The time interval between adjacent data points in the signal output from the cross correlation operation equals T_{clk}/M_1 . To achieve a high time resolution, a second-step interpolation will be conducted subsequently.

E. Second-step interpolation

By applying the second-step interpolation, the maximum value of the cross correlation is precisely located, which marks the fine time of the time interval to be measured.

Since the first-step interpolation factor M_1 is selected, the second-step interpolation factor M_2 could be determined by Eq. (7). For instance, when M_1 is set to 16, M_2 should be 2^{12} . Typically, when employing FIR interpolation filters, the order of the filter is chosen as an integer multiple of the interpolation factor, such as 4–10 times. Direct implementation of M_2 -times interpolation would still require extremely high computational resources and prove challenging to implement in an FPGA. Therefore, it is necessary to design the structure of the second-step interpolation process.

In this work, the second-step interpolation is divided into K stages, as shown in Fig. 11. To reduce resource consumption, the data are truncated to 201 points before each interpolation stage, where the maximum value is centered. In order to prevent shape jumps at the edges of the data after interpolation filtering, a bias is added to make the first data equal 0. In each stage, an interpolation factor $M_{2,i}$ ($i = 1, 2, \dots, K$) is employed, and the peak index D_i is found and recorded. To simplify the implementation structure of

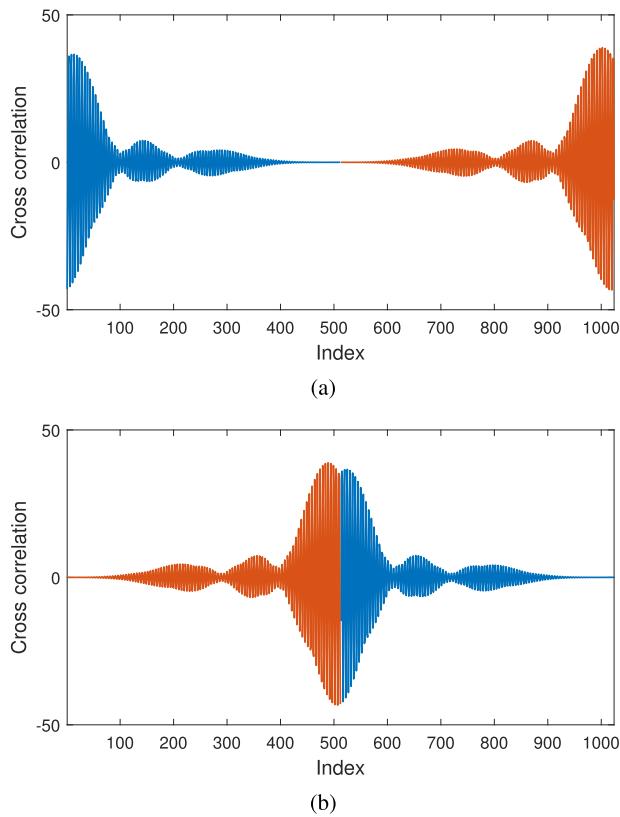


FIG. 10. The cross correlation function of two bandpass signals. (a) The output of the IFFT module. (b) The shifted cross correlation function.

the algorithm, the same interpolation filter is used in each stage. The needed number of stages K can be calculated by

$$K = \frac{\log_2 M_2}{\log_2 M_{2,i}}. \quad (12)$$

To minimize the number of required Digital Signal Processor (DSP) resources in FPGA, it is essential to determine the interpolation factor $M_{2,i}$ for each interpolation stage. To maintain a consistent transition bandwidth in the analog domain, the filter order N_{filt2} for each stage is set to be proportional to the interpolation factor $M_{2,i}$ under varying conditions. The number of required DSPs n_{DSP} is proportional to N_{filt2} and the number of stages K . Combining with Eq. (12), n_{DSP} satisfies the following formula:

$$n_{DSP} \propto \frac{M_{2,i}}{\log_2 M_{2,i}}. \quad (13)$$

In light of this, to minimize n_{DSP} , the ideal values for $M_{2,i}$ should be either 2 or 4. Furthermore, the interpolation stage achieves a lower processing time with an interpolation factor of 2, enabling the sequential implementation of each of the two interpolation stages. By reusing the hardware resources, the overall resource consumption can be further reduced. Therefore, $M_{2,i}$ is chosen to be 2 in our design, and 12 interpolation stages are required.

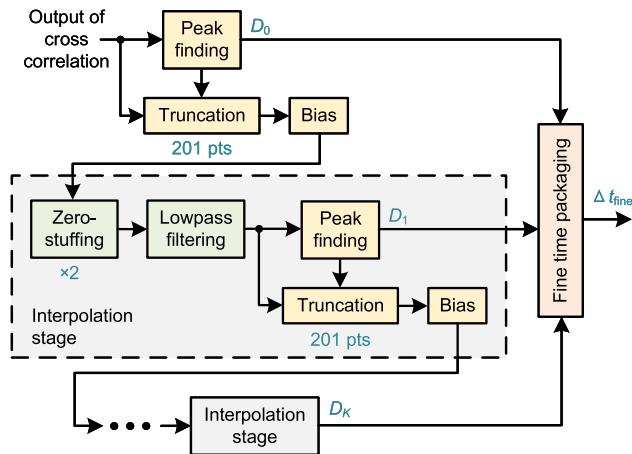


FIG. 11. Block diagram of the second-step interpolation.

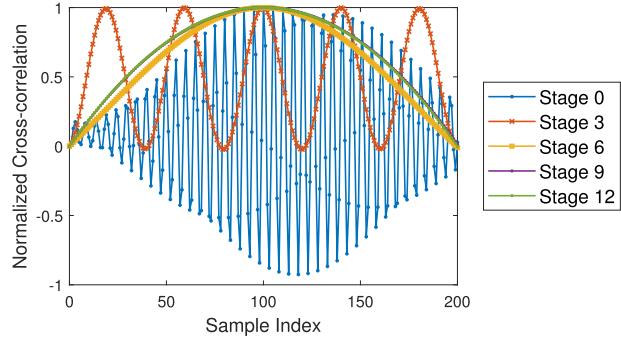


FIG. 12. Waveforms' output from some of the interpolation stages. The amplitude of each waveform is normalized.

Figure 12 displays the example waveforms' output from some of the stages. From this figure, it is evident that with each interpolation stage, there is a zoom-in on the position of the maximum value of the cross correlation function, ultimately enabling precise localization. It is important to note that when calculating the fine time intervals, it is necessary to divide the peak index output from each interpolation stage by the interpolation factor corresponding to that stage and then accumulate these values. After the processing of all the stages, the fine time is packaged by

$$\Delta t_{fine} = T_{clk} \cdot \left(D_0 + \sum_{i=1}^K \frac{D_i}{\prod_{j=1}^i M_{2,j}} \right) / M_1. \quad (14)$$

F. Result packaging

After the fine time is determined, it is packaged with the coarse time count. The delay of the test signal relative to the reference signal can be calculated by

$$\Delta T = T_{clk} \cdot \left((n_{0,test} - n_{0,ref}) + \frac{n_{1,test} - n_{1,ref}}{M_1} \right) + \Delta t_{fine}. \quad (15)$$

TABLE I. Resource utilization.

Pipeline stage	LUT	FF	DSP	BRAM
First-step interpolation	1851	4197	57	2.5
FFT ^a	8460	9685	33	9.5
Spectrum multiplication	364	794	16	2
Every two stages in the second-step interpolation ^b	1323	2330	25	1.5
Total	29 218	39 942	289	42.5
Ratio	12.0%	8.2%	15.0%	7.1%

^aFFT and IFFT call the same IP Core, and the resource consumption is the same.

^bTwelve interpolation stages in total are needed in the second-step interpolation.

G. FPGA implementation

The time extraction algorithm introduced in this section is implemented in an FPGA using a Xilinx High-level Synthesis (HLS) design methodology²⁴ and packed into an IP Core. The FFT and FIR operations are implemented by calling the Xilinx IP Core. The parameters that affect the resource consumption are selected as follows: M_1 is set to 16, N_{win1} is set to 512, N_{filt1} is set to 112, and N_{filt2} is set to 22.

The algorithm is implemented in the FPGA device, Xilinx KU040. The resource utilization of one time measurement channel is shown in Table I. The number of DSPs used in the first-step interpolation is half the filter order, since the response of the FIR filter is symmetrical. The data width in the second-step interpolation is larger than the DSP48, so two DSP slice columns are needed,²⁵ and the number of DSPs basically equals the filter order. Four time measurement channels can be implemented in one mid-range FPGA.

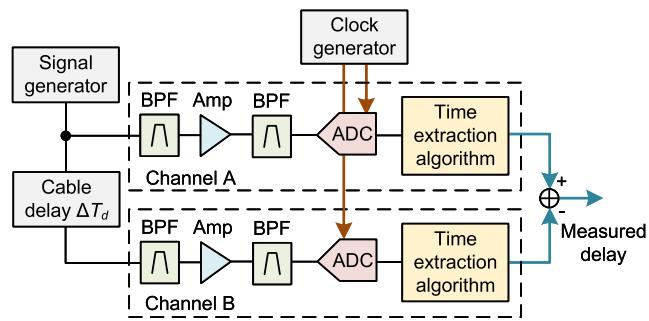
The time extraction algorithm is implemented in a pipeline structure. C/RTL Co-simulation is conducted to estimate the initiation interval (II) of each pipeline stage. The simulation results show that the highest II is 2150 clock periods. When the processing logic runs at a clock frequency of 250 MHz, a measurement rate of 116 kHz can be achieved.

V. SIMULATIONS OF TIME PRECISION

There are parameters influencing the time precision of the design. A simulation model is built, and the dominant parameter of time precision is analyzed.

A. Simulation model

The block diagram of the simulation model is shown in Fig. 13. First, a reference pulse signal is generated and processed by two time measurement channels. Then, referring to the standard “cable delay test” method,^{8,26} a series of test pulses are generated, split, and fed to each channel. One of the pulses is delayed by ΔT_d , which is set to a fixed value. In each channel, the pulses are bandpass-filtered, amplified, and sampled as in the front-end circuit. The noise of the amplifier, the ADC, and the sampling clock, and the feedthrough of the bandpass filter are considered. The time differences between each test pulse and the reference pulse are measured in each channel. The differences in time results between two channels are the measured cable delay. The phase of the sampling clock is random with

**FIG. 13.** Block diagram of the simulation model.

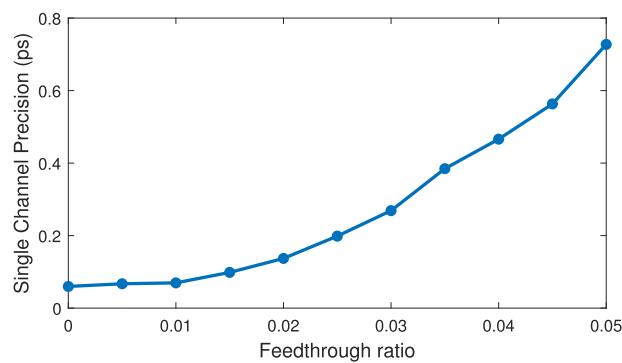
respect to the test pulses, so the measurement error satisfies a certain distribution. The time precision of a single channel is concerned in this work. Since the two channels are identical, the precision can be obtained by dividing the RMS value by $\sqrt{2}$.

It is noted that there are factors that are not considered in the simulation model, including the reflection of signals, the triple transit response of the bandpass filters, and the crosstalk of the analog circuits. Therefore, the simulation results are typically better than the test results.

B. Discussion on the feedthrough effect

Based on the simulation model, analyses of the factors affecting the time precision are conducted. The ratio of the amplitude of the feedthrough signal output from the bandpass filter to the amplitude of the input pulse signal is defined as the feedthrough ratio. While varying the feedthrough ratio in the simulation model, the corresponding time precision performance is obtained. While the cable delay ΔT_d is set to 0.4 ns, the simulation result is shown in Fig. 14.

From this figure, it is evident that when the feedthrough signal can be neglected, the time precision can reach very high levels. This indicates that the contribution of factors such as electronic noise and clock jitter to time precision is relatively small. However, as the amplitude of the feedthrough signal increases, the RMS precision degrades. In practical applications involving electronics that utilize

**FIG. 14.** Single channel precision vs feedthrough ratio. The noise of the amplifier, the ADC, and the sampling clock are set according to the analog front-end circuits. The cable delay is set to 0.4 ns.

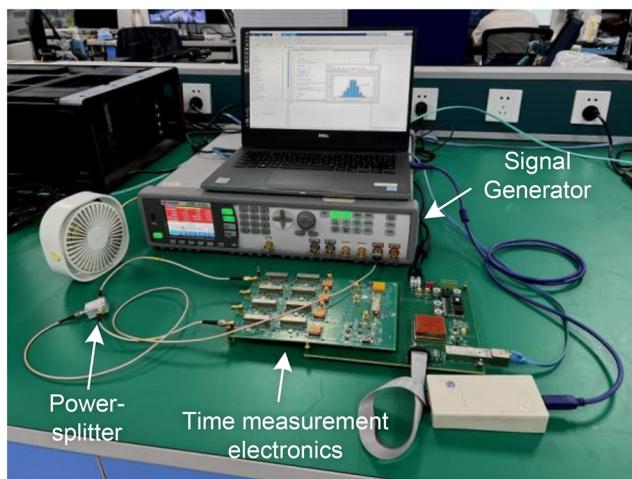


FIG. 15. The system under test.

bandpass filters, the feedthrough ratio typically falls within a range of 2.5%–3%. This corresponds to an RMS precision of ~ 0.25 ps. The results indicate that the feedthrough effect is the dominant contributing factor to time precision.

VI. TEST RESULTS

Tests are conducted to evaluate the performance of the time measurement system. The system under test is shown in Fig. 15. We use a Keysight 81160A pulse generator to generate a series of pulses of 1 ns rise time, 2 ns fall time, 5 V amplitude, and 2 ns pulse width in the internal trigger mode. The pulses are divided into two ways by a power-splitter, Mini-Circuits ZFRSC-123-S+, of which the insertion loss is 9.5 dB and the isolation loss is 19.5 dB. The amplitude of pulses is around 1.6 V after being split into two ways. Cables of different lengths are used to transfer each output of the power-splitter to the input channel of the time measurement electronics. After the power-splitter and the cable, the rise time of the pulses is around 1.2 ns. The source of the sampling clock is an onboard crystal oscillator. The first input pulse to each channel is regarded as the reference signal, and the following pulses are test signals. The time interval between each test signal and the reference signal is measured.

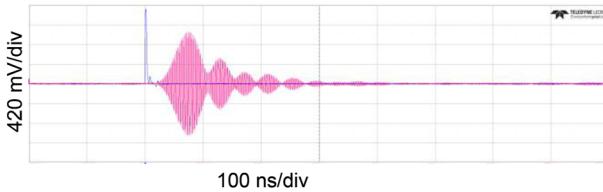


FIG. 16. Pulse shapes of the input pulse and bandpass signal in the functional test of the front-end circuit. The blue signal is the input fast pulse, and the pink signal is the amplified bandpass signal.

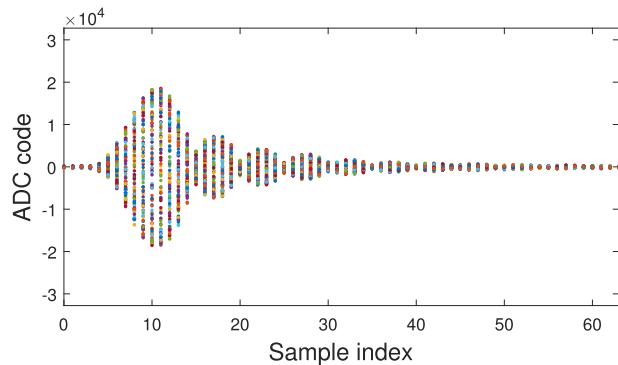
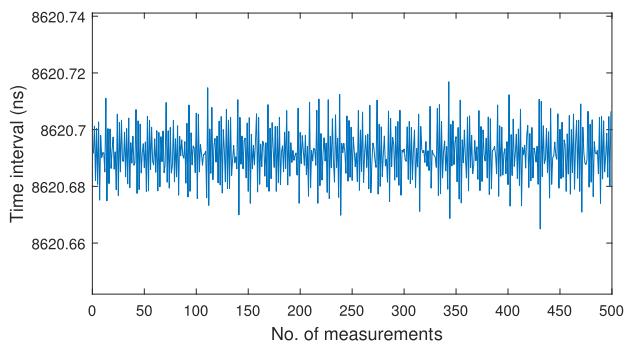


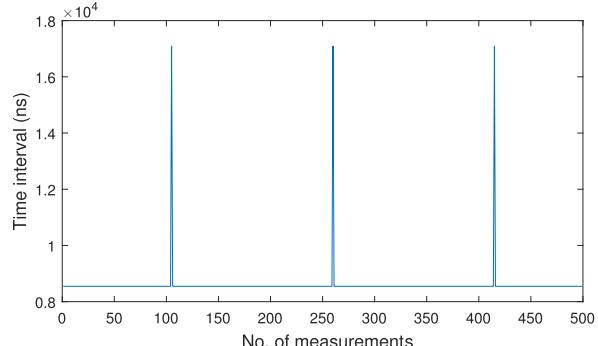
FIG. 17. Samples received by the FPGA. 100 trigger events are displayed in different colors.

A. Tests of the pulse shape

The pulse shapes of the input pulse and the bandpass signal are recorded using a digital oscilloscope and shown in Fig. 16. Similar to the simulation result, the bandpass signal lasts for hundreds of nanoseconds, and the amplitude is 1.1 V after being amplified 100 times. The samples that the FPGA received are shown in Fig. 17 with 100 trigger events.



(a)



(b)

FIG. 18. Test results of the maximum measurement rate. (a) 116 kHz. (b) 117 kHz.

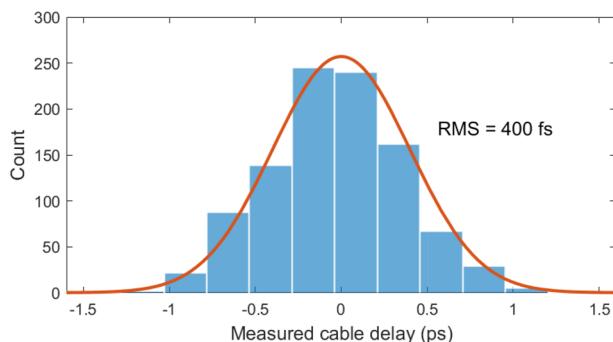


FIG. 19. Histogram of the measurement error with the cable delay set to 0.4 ns.

B. Tests of the maximum measurement rate

Set the frequency of the input pulse signal to 116 kHz, and the time interval between every two adjacent pulses is calculated. The results are shown in Fig. 18(a). By increasing the frequency of the pulse signal, measurement results may be lost due to the overflow of a FIFO between the pipeline stages, as shown in Fig. 18(b). Therefore, the maximum measurement rate is 116 kHz.

C. Tests of the time precision

When the cable delay ΔT_d is set to 0.4 ns, the histogram of the measurement error is shown in Fig. 19. The precision is presented as $1/\sqrt{2}$ of the RMS value of the measurement error, which is around 0.3 ps. The test results are in good agreement with the simulation results.

Furthermore, tests between different channels with different cable delay values are conducted. Figure 20 shows the test results. The precision is typically better than 0.4 ps.

D. Discussion

The sampling rate and literature time precision performance of this work are compared with those of other waveform digitization systems in Table II. The precision is compared under a short cable delay. If the precision performance is given as time interval precision, it is converted to time precision for a single channel. This table

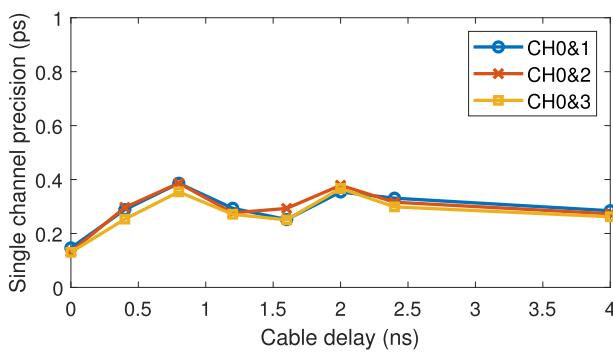


FIG. 20. Precision vs cable delay. Test results between different channels are presented.

TABLE II. Comparison between the work of this paper and the other waveform digitization systems.

First author	Rise time (ns)	Timing method	Sampling rate (Gps)	Precision (ps)
Stricker-Shaver ¹⁸	3	Digital leading edge discrimination	5	1.7
Stricker-Shaver ¹⁸	3	Cross correlation	5	0.6
Delagnes ²⁷	0.3	Digital constant fraction discrimination	6.4	3.5
Oberla ³	0.9	Fitting	10	4
Warburton ⁷	6	Interpolated timing	0.5	2.1
This work	1.2	Bandpass sampling and cross correlation	0.12	0.4

shows that the work of this paper can achieve an improved precision at a lower sampling rate in the research on high-precision time measurements of pulse signals based on waveform digitization.

In comparison with Pánek's work,¹⁴ the work of this paper can achieve a higher time precision while also greatly increasing the electronic system's measurement rate.

VII. CONCLUSION

In this article, we designed the high-precision time measurement electronics using the bandpass sampling method. The input pulse was sent to a bandpass filter, and the undersampling technique was used to digitize the bandpass signal. A pipelined time extraction algorithm using cross correlation and interpolation was designed, achieving a measurement rate of 116 kHz. Simulations and tests were conducted to evaluate the precision of the timing system, and the test results showed that the typical RMS precision of a single channel is better than 0.4 ps.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 12035014) and the Youth Innovation Promotion Association, CAS.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Yichun Fan: Conceptualization (equal); Data curation (lead); Investigation (lead); Methodology (equal); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead).

Lei Zhao: Conceptualization (equal); Funding acquisition (lead); Methodology (equal); Project administration (equal); Supervision (equal); Writing – review & editing (equal). **Jiajun Qin:** Conceptualization (equal); Methodology (equal); Project administration (equal); Supervision (equal); Writing – review & editing (equal). **Jiaming Li:** Investigation (supporting); Validation (supporting). **Shiya Huang:** Investigation (supporting); Validation (supporting). **Zhe Cao:** Conceptualization (supporting); Methodology (supporting).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹S. Ritt, R. Dinapoli, and U. Hartmann, “Application of the DRS chip for fast waveform digitizing,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **623**, 486–488 (2010).
- ²A. Ronzhin, M. G. Albrow, S. Los, E. Ramberg, Y. Guo, H. Kim, A. Zatserklyanyi, M. Mazzillo, B. Carbone, G. Condorelli *et al.*, “Waveform digitization for high resolution timing detectors with silicon photomultipliers,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **668**, 94–97 (2012).
- ³E. Oberla, J.-F. Genat, H. Grabs, H. Frisch, K. Nishimura, and G. Varner, “A 15 GSa/s, 1.5 GHz bandwidth waveform digitizing ASIC,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **735**, 452–461 (2014).
- ⁴C.-M. Du, J.-D. Chen, X.-L. Zhang, H.-B. Yang, K. Cheng, J. Kong, Z.-G. Hu, Z.-Y. Sun, H. Su, and H.-S. Xu, “Study of time resolution by digital methods with a DRS4 module,” *Chin. Phys. C* **40**, 046101 (2016).
- ⁵J. Y. Yeom, R. Vinke, and C. S. Levin, “Optimizing timing performance of silicon photomultiplier-based scintillation detectors,” *Phys. Med. Biol.* **58**, 1207 (2013).
- ⁶G. Simpson, S. Curtoni, D. Dauvergne, M.-L. Gallin-Martel, S. Marcatilli, and G. Thiamova, “Performance of four digital algorithms for γ - γ timing with LaBr₃ (Ce) scintillators,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **940**, 50–55 (2019).
- ⁷W. K. Warburton and W. Hennig, “New algorithms for improved digital pulse arrival timing with sub-GSpCs ADCs,” *IEEE Trans. Nucl. Sci.* **64**, 2938–2950 (2017).
- ⁸Y. Fan, L. Zhao, J. Qin, Z. Jiang, Z. Cao, S. Liu, and Q. An, “Research and verification on real-time interpolated timing algorithm based on waveform digitization,” *IEEE Trans. Nucl. Sci.* **67**, 2246–2254 (2020).
- ⁹P. Panek and I. Prochazka, “Time interval measurement device based on surface acoustic wave filter excitation, providing 1 ps precision and stability,” *Rev. Sci. Instrum.* **78**, 094701 (2007).
- ¹⁰P. Panek, “Time-interval measurement based on SAW filter excitation,” *IEEE Trans. Instrum. Meas.* **57**, 2582–2588 (2008).
- ¹¹P. Panek, “Random errors in time interval measurement based on SAW filter excitation,” *IEEE Trans. Instrum. Meas.* **57**, 1244–1250 (2008).
- ¹²I. Prochazka and P. Panek, “Nonlinear effects in the time measurement device based on surface acoustic wave filter excitation,” *Rev. Sci. Instrum.* **80**, 076102 (2009).
- ¹³P. Panek, I. Prochazka, and J. Kodet, “Time measurement device with four femtosecond stability,” *Metrologia* **47**, L13 (2010).
- ¹⁴P. Panek, J. Kodet, and I. Prochazka, “Event timing device providing sub-picosecond precision,” in *2013 Joint European Frequency and Time Forum & International Frequency Control Symposium (EFTF/IFC)* (IEEE, 2013), pp. 167–170.
- ¹⁵W. Kester and J. Bryant, “Section 2—Sampled data systems,” in *Mixed-Signal and DSP Design Techniques*, edited by W. Kester (Newnes, Burlington, 2003), pp. 15–58.
- ¹⁶M. Shafiee, S. A. F. Feghhi, and J. Rahighi, “Experimental performance evaluation of ILSF BPM data acquisition system,” *Measurement* **100**, 205–212 (2017).
- ¹⁷A. V. Oppenheim, *Discrete-Time Signal Processing* (Pearson Education India, 1999).
- ¹⁸D. Stricker-Shaver, S. Ritt, and B. J. Pichler, “Novel calibration method for switched capacitor arrays enables time measurements with sub-picosecond resolution,” *IEEE Trans. Nucl. Sci.* **61**, 3607–3617 (2014).
- ¹⁹L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, 1975).
- ²⁰J. Kalisz, “Review of methods for time interval measurements with picosecond resolution,” *Metrologia* **41**, 17 (2003).
- ²¹C. Fernández-Ramírez, E. M. de Guerra, A. Udias, and J. M. Udias, “Properties of nucleon resonances by means of a genetic algorithm,” *Phys. Rev. C* **77**, 065212 (2008).
- ²²V. Sanchez-Tembleque, V. Vedia, L. Fraile, S. Ritt, and J. M. Udias, “Optimizing time-pickup algorithms in radiation detectors with a genetic algorithm,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **927**, 54–62 (2019).
- ²³S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: Past, present, and future,” *Multimedia Tools Appl.* **80**, 8091–8126 (2021).
- ²⁴Xilinx, Vitis High-Level Synthesis User Guide (UG1399), 2023.
- ²⁵Xilinx, FIR Compiler v7.2 LogiCORE IP Product Guide, 2022.
- ²⁶L. Zhao, L. Kang, J. Zhou, S. Liu *et al.*, “A 16-channel high-resolution time and charge measurement module for the external target experiment in the CSR of HIRFL,” *Nucl. Sci. Tech.* **25**, 29 (2014).
- ²⁷E. Delagnes, D. Breton, H. Grabs, J. Maalmi, and P. Rusquart, “Reaching a few picosecond timing precision with the 16-channel digitizer and timestamper SAMPIC ASIC,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **787**, 245–249 (2015).