Analog Integrated Circuits and Signal Processing
https://doi.org/10.1007/s10470-019-01443-9

Digital background calibration algorithm and its FPGA implementation for timing mismatch correction of time-interleaved ADC

Asgar Abbaszadeh[1] · Esmaeil N. Aghdam[1] · Alfredo Rosado-Muñoz[2]

Received: 8 June 2018 / Revised: 6 February 2019 / Accepted: 14 March 2019

Abstract

Sample time error can degrade the performance of time-interleaved analog to digital converters (TIADCs). A fully digital background algorithm is presented in this paper to estimate and correct the timing mismatch errors between four interleaved channels, together with its hardware implementation. The proposed algorithm provides low computation burden and high performance. It is based on the simplified representation of the coefficients of the Lagrange interpolator. Simulation results show that it can suppress error tones in all of the Nyquist band. Results show that, for a four-channel TIADC with 10-bit resolution, the proposed algorithm improves the signal to noise and distortion ratio (SNDR) and spurious-free dynamic range (SFDR) by 19.27 dB and 35.2 dB, respectively. This analysis was done for an input signal frequency of $0.09f_s$. In the case of an input signal frequency of $0.45f_s$, an improvement by 33.06 dB and 43.14 dB is respectively achieved in SNDR and SFDR. In addition to the simulation, the algorithm was implemented in hardware for real-time evaluation. The low computational burden of the algorithm allowed an FPGA implementation with a low logic resource usage and a high system clock speed (926.95 MHz for four channel algorithm implementation). Thus, the proposed architecture can be used as a post-processing algorithm in host processors for data acquisition systems to improve the performance of TIADC.

Keywords TIADC · Background calibration · Lagrange interpolation · Timing mismatch

1 Introduction

Radio receiver designs tends to replace inaccurate and unreliable analog blocks by digital signal processing. The final goal of the radio receiver designers is to directly digitize the output signal of the antenna and implement all the receiver functions in digital hardware. This is achieved by moving the analog to digital converter (ADC) closer to the antenna for high-frequency and wide-bandwidth systems [1]. For this reason, the sampling rate and bit resolution of ADCs might become a bottleneck in modern communication systems [2].

Depending on the application, the required specifications for these ADCs can range from 1 GS/s and 11-bit to over 10 GS/s and 4-bit [3]. Usually, high-speed single channel ADC architectures cannot meet the high throughput requirements of these systems. However, parallel implementation of ADC channels in a time-interleaved structure can increase the sampling rate [4]. In a TIADC structure, each channel operates at its maximum possible speed. Therefore, time interleaving of M ADC channels increases the speed of the overall system, being M times faster than the single channel at the cost of a small increase in area and power dissipation [5]: here resides the main advantage of TIADC. Figure 1 shows a simplified block and timing diagram of an M-channel TIADC. The analog input signal $x(t)$ is fed into the ADC channels periodically, using an analog demultiplexer. In turn, digitized outputs of the ADC channels are sent to a multiplexer to perform

✉ Esmaeil N. Aghdam
najafiaghdam@sut.ac.ir

Asgar Abbaszadeh
a_abbaszadeh@sut.ac.ir

Alfredo Rosado-Muñoz
alfredo.rosado@uv.es

[1] Department of Electrical Engineering, Sahand University of Technology, Tabriz, Iran

[2] GPDD-DIE-ETSE, University of Valencia, 46100 Burjassot, Valencia, Spain

Published online: 22 March 2019

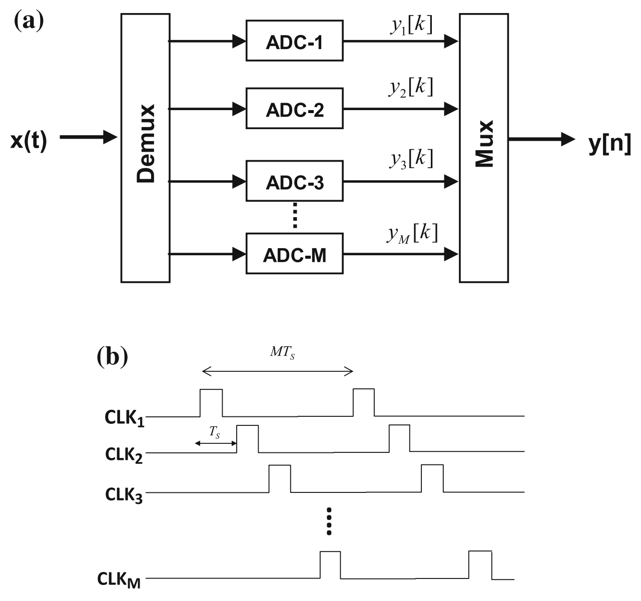


Fig. 1 M-channel time-interleaved analog to digital converter (TIADC) working principle: **a** simplified block diagram, **b** timing diagram

high-speed output. Typically, ADC channels operate in a $\frac{f_s}{M}$ sampling rate and the timing difference between the sampling instances of neighboring channels is T_S. By interleaving M channels, the sampling rate of the overall system increases to f_s. Additionally, the time-interleaving structure reduces the clock network power consumption, the probability of metastable states, and increases the flexibility to adjust the 'power versus speed' trade-off [6].

Despite the clear advantages of TIADC, several issues must be considered: offset, gain and sample time mismatches between channels can degrade the performance of TIADC. These mismatches are time varying and cannot be corrected by trimming or foreground calibration methods. They must be estimated and corrected during the normal operation of ADC, without interruption. Due to this, background calibration methods have been introduced in recent years to correct the mismatch effects. The effects from offset and gain mismatches can be easily removed by a simple least mean square (LMS) loop [7, 8]. On the contrary, the aperture time error introduces a significant effect on the output signal spectrum: it depends on the input signal frequency [9]. With an input signal frequency of f_{in}, the sampling time error causes image spurs to appear at frequencies f_k, given in (1), where f_s is the sampling frequency for the overall system and M is the number of interleaved channels.

$$f_k = \frac{kf_s}{M} \pm f_{in} \qquad k = 1, 2, 3, \ldots \tag{1}$$

The power of the image spurs increases as the input signal frequency increases. Therefore, the correction of thetiming mismatch becomes more difficult for frequencies close to $\frac{f_s}{2}$.

Timing mismatch can be corrected by two methods: VLSI circuit design and signal processing algorithms [10]. For circuit design, timing error detection is digitally performed and error correction is performed in the analog domain. In the case of signal processing algorithms, both detection and correction are performed in the digital domain. Most of the digital correction methods require large digital filters or high order interpolators requiring high power consumption and hardware resources, thus limiting the use of these methods. On the other hand, circuit designers try to avoid digital components in the circuit and solve this problem by analog methods such as variable delay lines [4]. However, this method provides a limited performance and can also increase the phase noise of the output signal [6].

A mismatch correction algorithm presented in [11] for two-channel TIADC mixes the output. its chopped version produces a DC output proportional to the timing error. Additionally, timing error correction is done by FIR filtering. A modified version of this method for calibration of four-channel TIADC has been reported in [12]. Similarly, an error correction using a variable delay line is proposed in [13]. In [14], authors assume the output signal to be the sum of an ideal signal and an error term. The error signal caused by timing skew is calculated as the product of the derivative of the signal and the estimated timing skew, it is then subtracted from the channel output to recover the ideal samples. The signal derivative is obtained by applying a differentiating FIR filter to the TIADC samples. In some cases, error detection is done by a cross-correlation function and error correction by using a voltage controlled sampling switch [15]. In [16], multiple filters with different delays are used to produce different delays. Based on the detected error, the calibration algorithm switches to the desired delay filter. This method has a limited accuracy as it depends on the existing number of delay filters. Blind calibration methods are also used, requiring a high computational cost [17–20]. For instance, the method proposed in [20] requires a 7×7 matrix inversion for online reconstruction filter design.

Channel sampling randomization method [21], split ADC architecture [22, 23] and reference channel approaches [24–27] are using extra channels to correct the timing mismatch error. Cubic spline interpolation [9] and adaptive calibration proposed in [28] and [29] detect the sampling time error by applying a reference signal to the system. The algorithm proposed in [30] estimates timing error by comparing the mean value of the multiplication of signals in two adjacent channels; a digitally controlled delay line, capacitor array based, is used to manage sampling time of channels. Some fractional delay [31, 32], filter bank [33] and interpolation techniques [34, 35] are used for sample time error correction. However, they are not able to work in all of the Nyquist band and need signal oversampling. An interesting simple algorithm using variable delay buffers used to manage the clock signal is proposed in [36].

A simple digital background calibration algorithm to correct sample time error using low logic resources hardware, based on a modification in the Lagrange interpolation technique is presented in this paper. Both detection and compensation for timing error is done in the digital domain. Section 2 describes the proposed algorithm. Hardware implementation and simulation results are introduced in Sects. 3 and 4 , respectively. Sections 5 and 6 present the discussion and conclusions of this work.

2 Sampling timing calibration algorithm

A sampling time error correction algorithm for a 4-channel TIADC is proposed. For an M-channel TIADC with overall sampling period of T_s, sampling period for each channel is MT_s. The nth sampling instance of the ith channel $T_i[n]$ is described in (2) [17], ΔT_i being the timing error of the ith channel.

$$T_i[n] = nMT_s + (i-1)T_s + r_iT_s \tag{2}$$

For simplicity, we use normalized timing values with respect to T_s instead of absolute values. Thus, (2) can be represented as (3) where $t_i[n]$ is the nomalized nth sampling instance of the ith channel and r_i is the relative timing error of the ith channel ($r_i = \frac{\Delta T_i}{T_s}$).

$$t_i[n] = nM + (i-1) + r_i \tag{3}$$

Let us to consider a two-channel TIADC. The first channel is considered as the reference channel ($r_1 = 0$). As shown in Fig. 2, the first channel samples in $t_1[n]$ and $t_1[n+1]$ generate output values $y_1[n]$ and $y_1[n+1]$, and the second channel samples in $t_2[n]$ generate the output value $y_2[n]$. The sampling time of the second channel has an offset from the ideal value $\hat{t}_2[n]$, equal to $\Delta T_2 = r_2T_s$. The ideal value for the second channel output is $\hat{y}_2[n]$ under an ideal sampling instance. Then, timing error correction requires estimating the offset value from the actual samples. In general, for an M-channel TIADC, we need to estimate the output values of $M-1$ channels. In order to do this estimation, two steps are considered: error detection and correction. In the error detection step, r_i is obtained, and, in the correction step, $\hat{y}_i$ is calculated by using the r_i error previously detected.

2.1 Sampling time error detection

As defined in (4), let us define x_1 and x_2 as the difference between a two sequential outputs in two-channel TIADC where |.| denotes the absolute value.

$$\begin{aligned} x_1 &= |y_2[n] - y_1[n]| \\ x_2 &= |y_1[n+1] - y_2[n]| \end{aligned} \tag{4}$$

For a system without timing error, the average values for x_1 and x_2 are equal. However, in case of an offset r_2 in the sampling time of the second channel, the timing difference between sampling instances of $y_2[n]$ and $y_1[n]$ is $2r_2$ greater than the timing difference between sampling instances of $y_1[n+1]$ and $y_2[n]$. Therefore, the average of x_1 is supposed to be greater than the average of x_2.

In [36], it is proved that the difference between the average values of x_1^2 and x_2^2 can be written as in (5) where $R[t]$ is the autocorrelation function of the input signal.

$$E[x_1^2] - E[x_2^2] = -4T_sr_2\frac{dR}{d\tau} \tag{5}$$

Equation (5) shows that the timing error is proportional to the difference of the power of x_1 and x_2. Figure 3 shows a block diagram for the timing error detection in a two-channel TIADC. In this structure, error detection requires three steps:

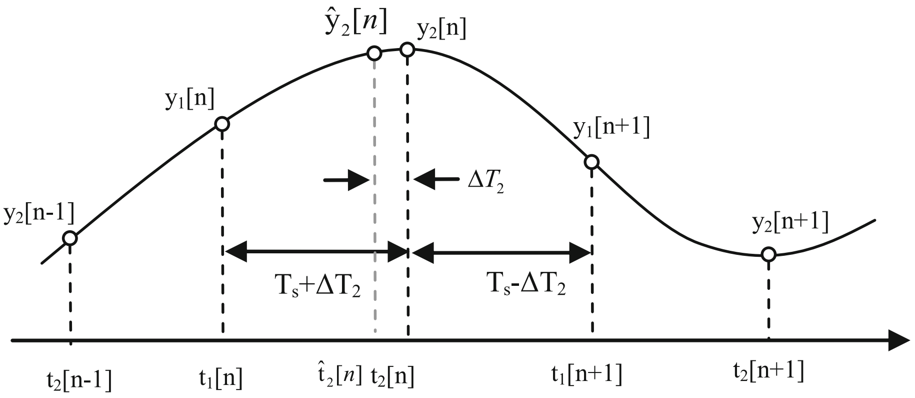


Fig. 2 Waveform illustration of timing mismatch error in a two-channel TIADC

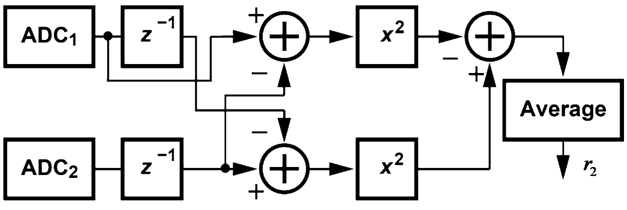


Fig. 3 Timing mismatch detection block diagram

- compute x_1 and x_2 by using two subtractors.
- calculate x_1^2 and x_2^2.
- take the difference between x_1^2 and x_2^2.
- accumulate $x_1^2 - x_2^2$

2.2 Timing error correction

The goal is to estimate the desired outputs at the ideal sampling instances. In this work, we propose the simplified Lagrange interpolation method. Consider a signal sampled at $N+1$ distinct instances t_i, with corresponding values y_i, where $i = 0, 1, \ldots, N$. The interpolated signal value $\hat{y}(t)$, under an ideal sampling instance t, and based on the Lagrange interpolation defined in (6) with h_i in (7).

$$\hat{y}(t) = \sum_{i=-N/2}^{i=N/2} h_i y_i \tag{6}$$

$$h_i = \prod_{\substack{j=-N/2 \\ j \neq i}}^{j=N/2} \frac{t - t_j}{t_i - t_j} \tag{7}$$

In order to calculate h_i, we need to compute $2N-2$ multiplications, $2N$ additions and one division. Thus, $2N^2 - 2$ multiplications, $2N^2 + 2N$ additions and $N+1$ divisions are required for computing $N+1$ coefficients. Additionally, in each sampling clock cycle, $N+1$ multiplications and N additions are required to obtain (6). All these required arithmetic operations make it difficult to implement high order interpolators for high-speed TIADCs in a real-time hardware system since it requires a large number of logic resources and power. In order to reduce the required arithmetic operations, let us consider the fourth order Lagrange interpolator, i.e. five coefficients must be calculated as shown in (8).

$$\boldsymbol{H}_2 = (h_{-2}, h_{-1}, h_0, h_1, h_2) \tag{8}$$

For a two-channel TIADC, samples corresponding to the second channel must be corrected. We need to estimate $\hat{y}_2[n]$ under the ideal sampling instance $\hat{t}_2[n] = 2n+1$. The timing window to calculate the $\hat{y}_2[n]$ value shown in Fig. 2 can be obtained as in (9).

$$\begin{aligned} \boldsymbol{\tau}_2 &= (t_{-2}, t_{-1}, t_0, t_1, t_2) \\ &= (t_2[n-1], t_1[n], t_2[n], t_1[n+1], t_2[n+1]) \end{aligned} \tag{9}$$

Using (3), the elements of $\boldsymbol{\tau}_2$ can be computed as in (10).

$$\begin{aligned} t_{-2} &= 2n - 1 + r_2 \\ t_{-1} &= 2n \\ t_0 &= 2n + 1 + r_2 \\ t_1 &= 2n + 2 \\ t_2 &= 2n + 3 + r_2 \end{aligned} \tag{10}$$

Thus, the coefficients of the interpolator are obtained as in (11).

$$\begin{aligned} h_{-2} &= \frac{-r_2(2+r_2)}{8(1-r_2)(3-r_2)} \\ h_{-1} &= \frac{r_2(2-r_2)(2+r_2)}{2(1-r_2)(1+r_2)(3+r_2)} \\ h_0 &= \frac{(2-r_2)(2+r_2)}{4(1+r_2)(1-r_2)} \\ h_1 &= \frac{-r_2(2-r_2)(2+r_2)}{2(3-r_2)(1-r_2)(1+r_2)} \\ h_2 &= \frac{r_2(2-r_2)}{8(3+r_2)(1+r_2)} \end{aligned} \tag{11}$$

In addition, it is possible to represent (11) as a polynomial of r_2 and, as the timing error is very small in comparison to T_s ($r_2 << 1$), (11) can be approximated by the first order Maclaurin expansion (12).

$$\begin{aligned} h_{-2} &\approx -0.0833r_2 - 0.1527r_2^2 - 0.0277r_2^3 + 0.01388r_2^4 \\ h_{-1} &\approx 0.6666r_2 - 0.2222r_2^2 + 0.5r_2^3 + 0.2777r_2^4 + \cdots \\ h_0 &\approx 1 + 0.75r_2^2 - 0.25r_2^4 \\ h_1 &\approx -0.6666r_2 - 0.2222r_2^2 - 0.5r_2^3 + 0.1666r_2^4 + \cdots \\ h_2 &\approx 0.0833r_2 - 0.1527r_2^2 + 0.0277r_2^3 + 0.01388r_2^4 \end{aligned} \tag{12}$$

Neglecting orders higher than two for r_2, which are obviously small, (12) can be rewritten as in (13).

$$\begin{aligned} h_{-2} &\approx -0.0833r_2 - 0.1527r_2^2 \\ h_{-1} &\approx 0.6666r_2 - 0.2222r_2^2 \\ h_0 &\approx 1 + 0.75r_2^2 \\ h_1 &\approx -0.6666r_2 - 0.2222r_2^2 \\ h_2 &\approx 0.0833r_2 - 0.1527r_2^2 \end{aligned} \tag{13}$$

In order to prove the assumptions made, Table 1 shows a comparison for the approximated coefficient values and the exact values for an error value $r_2 = 0.01$. As shown in the table, the approximated coefficients by the second order polynomial of r_2 result in an accuracy with an error lower than 2^{-19}.

The same analysis can be done to calculate coefficients for higher order interpolations and (7) can be rewritten as (14) where $h'_{1,i}$ and $h'_{2,i}$ are fixed coefficients.

$$h_i = \begin{cases} h'_{1,i} r_2 + h'_{2,i} r_2^2 & i \neq 0 \\ 1 + h'_{2,0} r_2^2 & i = 0 \end{cases} \tag{14}$$

As in (13), $h'_{1,i}$ has the antisymmetric property and $h'_{2,i}$ has the symmetric property (15).

$$\begin{aligned} h'_{1,i} &= -h'_{1,-i} \\ h'_{2,i} &= h'_{2,-i} \end{aligned} \tag{15}$$

Then, we can write the output equation for $\hat{\imath}_2[n]$ as (16).

$$\begin{aligned} \hat{y}_2[n] = \hat{y}[2n+1] &= r_2 \sum_{i=1}^{N/2} h'_{1,i}(y[2n+i] - y[2n-i]) \\ &+ r_2^2 \left(\sum_{i=1}^{N/2} h'_{2,i}(y[2n+i] + y[2n-i]) + h'_{2,0} y[2n] \right) + y[2n] \end{aligned} \tag{16}$$

This form of calculation reduces the computation and, additionally, there is no need to recalculate the coefficients in each clock cycle. N + 1 multiplications are required to perform $h'_{1,i}(y[2n+i] - y[2n-i])$, $h'_{2,i}(y[2n+i] + y[2n-i])$ and $h'_{2,0}y[2n]$. In addition, three multiplications are required to multiply the first term by r_2 and the second term by r_2^2 and computing r_2^2. On the other hand, N additions are required to compute $(y[2n+i] \pm y[2n-i])$ and $N-2$ additions are required to compute the summations in the first and the second terms. Considering the three additions in (16), $2N+1$ additions are required in total. Therefore, $N+4$ multiplications, $2N+1$ additions and no divisions are required to compute the algorithm, which provides a light burden and therefore ease the hardware implementation. Figure 4 shows the architecture of the reconstruction filter for $N = 5$. We need to estimate the ideal values of the samples corresponding to the second channel. For this reason, the reconstruction filter operates in $\frac{f_s}{2}$ frequency and the samples corresponding to the first channel are delayed for a duration equal to the latency of the reconstruction filter so that they are fed to the output.

The output of the error detection algorithm is proportional to the sampling time error and then, it does not correspond to the exact value of the error. We propose the error detection and correction in an LMS loop to minimize $x_1^2 - x_2^2$. The timing error detection algorithm is applied after the interpolator. The calibrated data are applied to the timing error detection algorithm to detect the remaining error values. Figure 5 illustrates the complete proposed architecture for estimating and correcting the sampling time error for two-channel TIADC. Delayed samples of the first channel and corrected samples of the second channel are fed to a multiplexer to perform the output. Note that the inputs to the MUX unit have a $\frac{f_s}{2}$ sampling rate and the output of the MUX unit has a f_s sampling rate. The loop gain, μ_t, controls the speed of convergence.

It is important to note that the Lagrange interpolator has a flat response in low frequencies but imposes additional signal phase error in frequencies higher than $\frac{f_s}{4}$. This fact can limit the performance of the algorithm [37, 38]. Using an LMS loop compensates for the additional phase error. In this structure, the timing error detection algorithm can detect the phase error imposed by the interpolator as a timing error and correct it in the LMS loop. Therefore, the estimated error will converge to values higher than the actual value of the error in frequencies higher than $\frac{f_s}{4}$ to compensate for the effect of the non-flat phase response of the interpolator.

Figure 6 shows the proposed architecture for sampling time error correction of a four-channel TIADC. A four-channel TIADC is considered as two separate two-channel TIADCs. The first TIADC includes ADC_1 and ADC_3 and the second, ADC_2 and ADC_4. In the first step, the timing error of ADC_3 with respect to ADC_1, and ADC_4 with respect to ADC_2, are detected and compensated. The outputs of these blocks are fed to another calibration block to correct the timing error between ADC_1 and ADC_2.

Increasing the order of the interpolator increases the SFDR and SNDR as well as the computational cost. For

Table 1 Comparison for interpolator coefficients for fourth order Lagrange interpolator in exact values and the approximation proposed ($r_2 = 0.01$)

	Exact values	Approximated values	Approximation error
h_{-2}	-8.4879×10^{-4}	-8.4857×10^{-4}	2.2×10^{-7}
h_{-1}	66.450×10^{-4}	66.4378×10^{-4}	-12.2×10^{-7}
h_0	1.0000750	1.0000750	0
h_1	-66.8946×10^{-4}	-66.8822×10^{-4}	12.4×10^{-7}
h_2	8.1823×10^{-4}	8.1773×10^{-4}	-5×10^{-7}

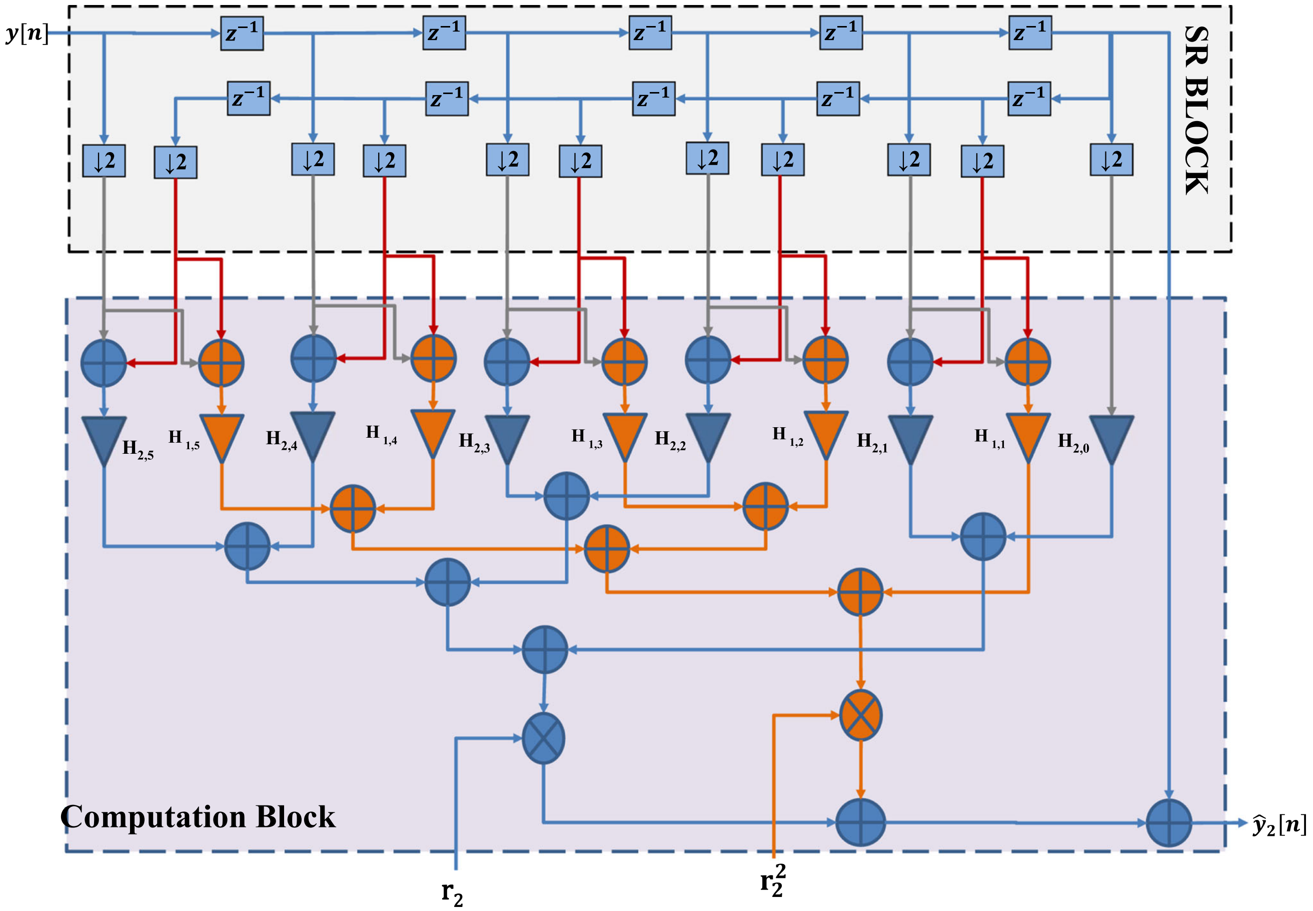


Fig. 4 Proposed reconstruction filter architecture for two-channel TIADC

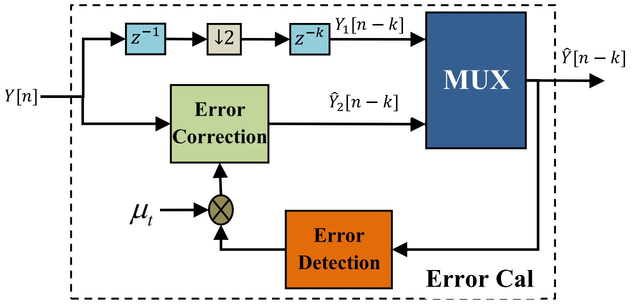


Fig. 5 Proposed timing mismatch calibration architecture for two-channel TIADC

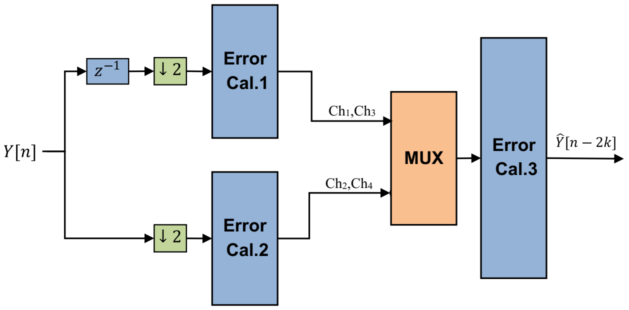


Fig. 6 Proposed timing mismatch calibration architecture for four-channel TIADC

this reason, it is necessary to select the proper order of interpolator. To make a decision about the order of the interpolator, multiple Matlab simulations with different orders of the interpolator have been done for a four channel 10-bit TIADC. Relative timing errors are assumed as $r_1 = 0$, $r_2 = 0.01$, $r_3 = 0.025$ and $r_4 = -0.015$, with a loop gain of $\mu_t = 2^{-20}$.

Figure 7 shows SFDR and SNDR of the calibrated signal compared with the order of the interpolator for an input signal frequency of $0.4f_s$. As shown in the figure, increasing the order of interpolator more than 18 has no improvement in SFDR and SNDR. Thus, $N = 18$ has been selected as the most adequate order of the interpolator.

3 Hardware implementation

In order to show the real-time capability of the proposed algorithms, a sampling time calibration core is defined. The core can be used as a standalone device placed in the signal

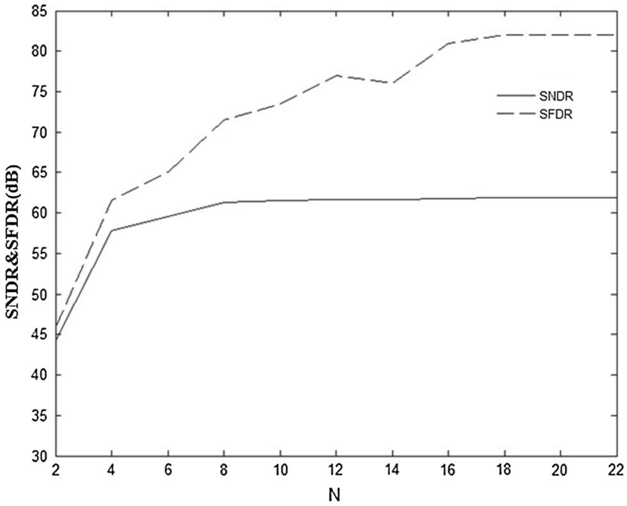


Fig. 7 SFDR and SNDR of the output signal versus order of the interpolator for input signal frequency of $0.4f_s$

processing sequence, or used as part of a single chip with additional processing stages. In this case, Xilinx Virtex 7 family is used to correct sampling time error of a four-channel 10-bit TIADC, taking advantage of its internal DSP48E arithmetic blocks and the LUT logic resources. Synthesis was done using Xilinx ISE 14.7 software.

Figure 8 represents the hardware implementation of the reconstruction filter for a fifth order interpolator. It consists of a shift registers block (SRB) and a computation block (CB). The SRB consists of $N+1$ registers to store the current and previous input data. All mathematical computations are done in the CB. The primary FPGA building block for the CB is the DSP48E (FPGA embedded arithmetic unit), which consists of a 25×18 two's complement multiplier/accumulator with 48-bit output, 18-bit preadder and 48-bit postadder (Fig. 9). $ALU1$ and $ALU2$ blocks in CB are DSP48E blocks which are set to calculate $(D-A) \times B$ and $(D+A) \times B$, respectively. These computations are used to obtain $h_{1,i}(y[n+i]-y[n-i])$ and $h_{2,i}(y[n+i]+y[n-i])$ in Eq. 16. Finally, $ALU3$ blocks are set to compute $A \times B$.

The multipliers and adders in the structure of the first step calibration blocks are operating at a $\frac{f_s}{4}$ frequency. In order to save hardware resources, the same CB is shared between $ErrorCal1$ and $ErrorCal2$ blocks which operate at a $\frac{f_s}{2}$ frequency (Fig. 10). Multipliers and adders in $ErrorCal3$ are also operating at $\frac{f_s}{2}$. In traditional architectures, all multipliers and adders are working at f_s frequency and the maximum possible frequency of the system is limited by the maximum frequency of the multipliers. In the proposed architecture, multipliers and adders operate at $\frac{f_s}{2}$. This property increases the maximum possible system frequency(f_{max}).

Table 2 shows the synthesis results for the proposed architecture. As shown in the table, the proposed calibration structure can operate at a clock frequency of 926.95 MHz and only a few percentage of the total resources in the device are used. As explained before, the maximum possible frequency is twice the maximum possible frequency of the DSP48E blocks.

4 Simulation results

In order to evaluate the hardware accuracy of the proposed algorithms, simulations in the hardware domain were carried out. For the constant coefficients of the interpolator, they are converted to Q2.16 fixed point format. Input data were generated in Matlab and saved in a text file in Q1.9 fixed point format so that they can be directly applied to the hardware interpolator inputs using a VHDL testbench. The hardware output values, corresponding to calibrated data, were exported back to Matlab to evaluate the performance of the system.

As the timing skew severely impacts the SNDR and SFDR in high frequencies, the efficiency of the algorithm was tested in frequencies close to $\frac{f_s}{2}$ as well as low frequencies. Figure 11 shows the output spectrum before and after calibration for $f_{in} = 0.09f_s$. In this case, undesired tones appear at $0.16f_s$, $0.34f_s$ and $0.41f_s$. For given error values, the amplitude of the largest error tone is about -48.6 dB before calibration and SNDR is equal to 42.61 dB. Figure 11(b) shows the FPGA-verified result of the proposed architecture for the spectrum of the corrected output signal. After applying the proposed calibration algorithm, the amplitude of the largest error tone is equal to -85.41 dB. Moreover, results show 19.27 dB improvement in SNDR, i.e. from 42.61 dB to 61.88 dB.

Figure 12 shows the output spectrum before and after calibration for $f_{in} = 0.45f_s$ demonstrating a 33.07 dB and 43.14 dB improvement in SNDR and SFDR, respectively. Figure 13 shows the estimated timing error convergence. Note that in the first step, timing error of the third and fourth channels are measured with respect to the first and second channel, respectively. Also, these error values are estimated in a two-channel TIADCs with a $2T_s$ sampling period. For this reason, the estimated timing error of the third and fourth channels are normalized with respect to $2T_s$ instead of T_s. Consequently, the estimated timing error of the third channel should converge to $\frac{T_s r_3 - T_s r_1}{2T_s} = \frac{r_3 - r_1}{2} = 0.0125$ instead of $r_3 - r_1$. Additionally, the estimated timing error of the fourth channel should converge to $\frac{r_4 - r_2}{2} = -0.0125$ instead of $r_4 - r_2$. In the second step, the timing error of the second channel is measured with respect

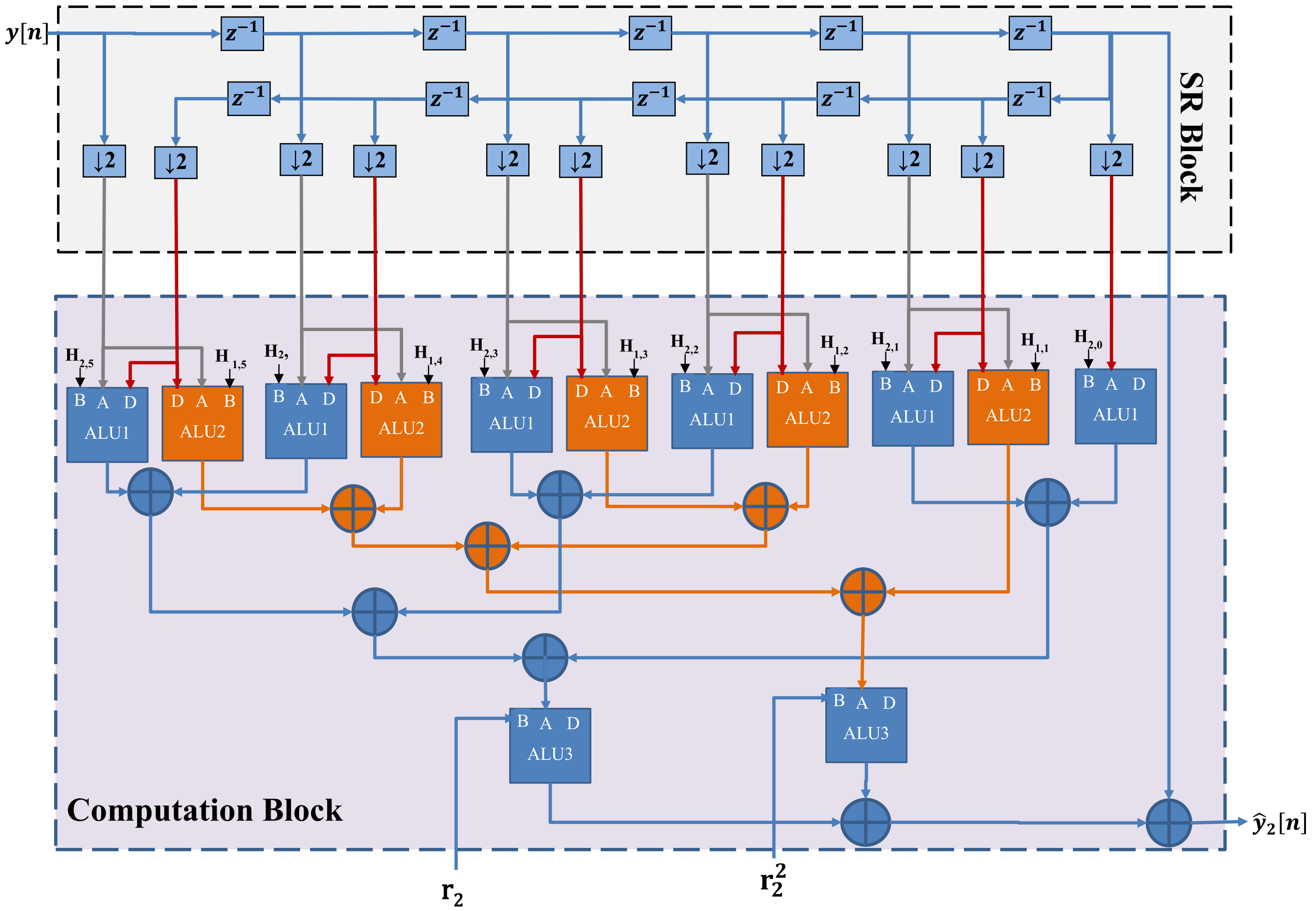
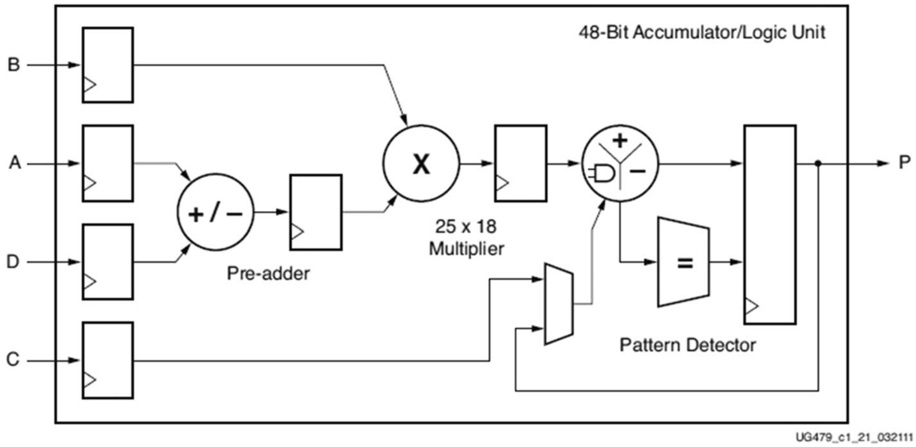


Fig. 8 Hardware implementation of the reconstruction filter for order $N = 5$

Fig. 9 Xilinx DSP48E block logical structure. Image source: [39]



to the first channel in a four-channel TIADC with a T_s sampling period and the estimated timing error of the second channel should converge to $r_2 - r_1 = 0.01$.

Figure 14 shows SNDR values depending on the input signal frequency. As seen in the figure, more than 60 dB SNDR is achievable in more than 80% of Nyquist band for a 10-bit TIADC. The performance of the system decreases when the frequency of the input signal is very close to one of the error tones. These frequency bands are $\frac{f_s}{8} \pm 0.005 f_s$, $\frac{f_s}{4} \pm 0.025 f_s$, $\frac{3f_s}{8} \pm 0.005 f_s$ and $\frac{f_s}{2} \pm 0.025 f_s$. In these cases, the error detection algorithm cannot detect the error values precisely.

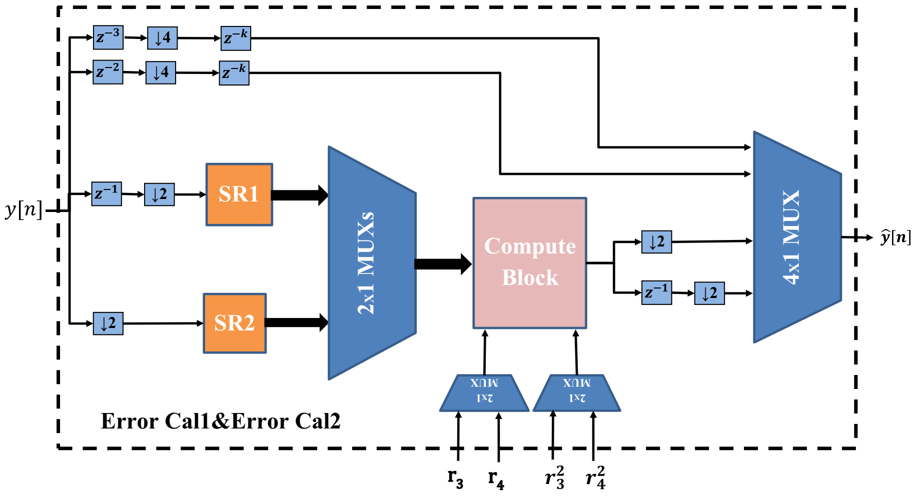


Fig. 10 Implementation of *ErrorCal*1 and *ErrorCal*2 using a single computation block

Table 2 FPGA resources used by the proposed architecture, synthesized for Virtex7 FPGA (7VX485TFFG1157)

Logic utilization	Used	Available in device
Number of slice registers	4287	607,200
Number of NuLUTs	2583	303,600
Number of DSP48	38	2800
f_{max} (MHz)	926.95	

5 Discussion

The proposed algorithm and associated hardware implementation greatly reduce the logic resources while maintaining a high SNDR value. Table 3 provides the main values comparing the proposed calibration structure with other recently reported digital background algorithms [4, 14, 19]. As discussed in Sect. 2.2, an order of $N = 18$ has been selected as the order of the interpolator in this paper. The error estimation algorithms used in [4] and [14] are similar to the algorithms we used in this article. However, in [4], a 5th order improved farrow structure is used for error correction and, in [14], an 8th order Lagrange differentiator filter is used for error correction. In [19], a 160th order Lagrange interpolator is used for error correction. Also, error values are obtained by a training algorithm before starting the correction algorithm. As shown in Table 3, all the compared algorithms show a maximum possible SNDR based on the ADCs resolution, but the performance of the other algorithms decreases in high frequencies while the performance of the proposed architecture does not degrade with a high input signal frequency: by increasing the input signal frequency from $0.1f_s$ to $0.45f_s$, the SNDR of the output signal decreases only by 0.22 dB while other algorithms result in a larger decrease than 8 dB. Moreover, the proposed system can achieve the fastest convergence rate in comparison to other algorithms.

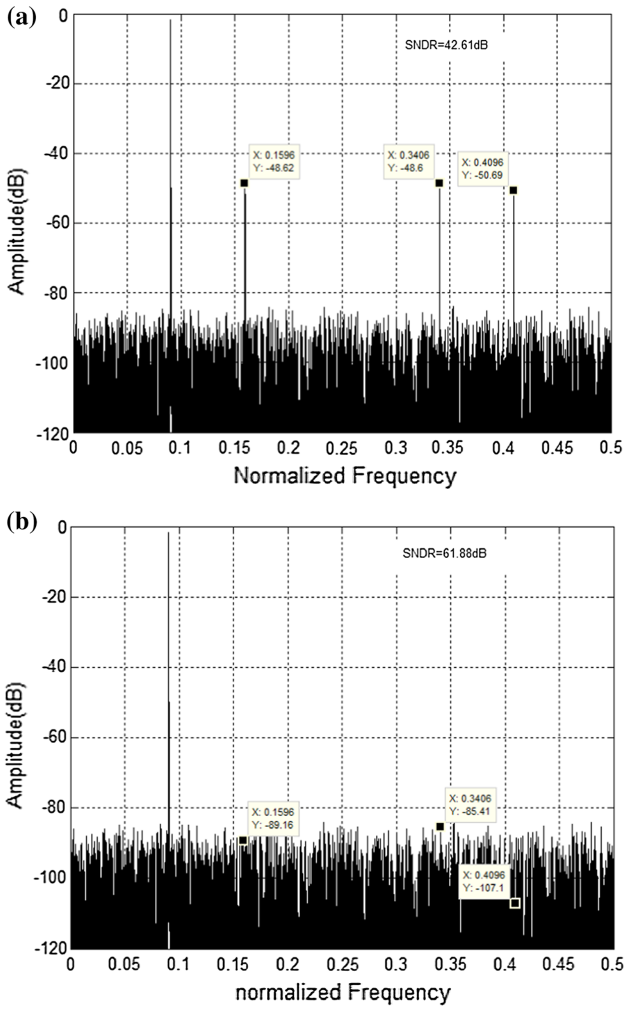


Fig. 11 Output spectrum at $f_{in} = 0.09f_s$: **a** before calibration, and **b** after calibration

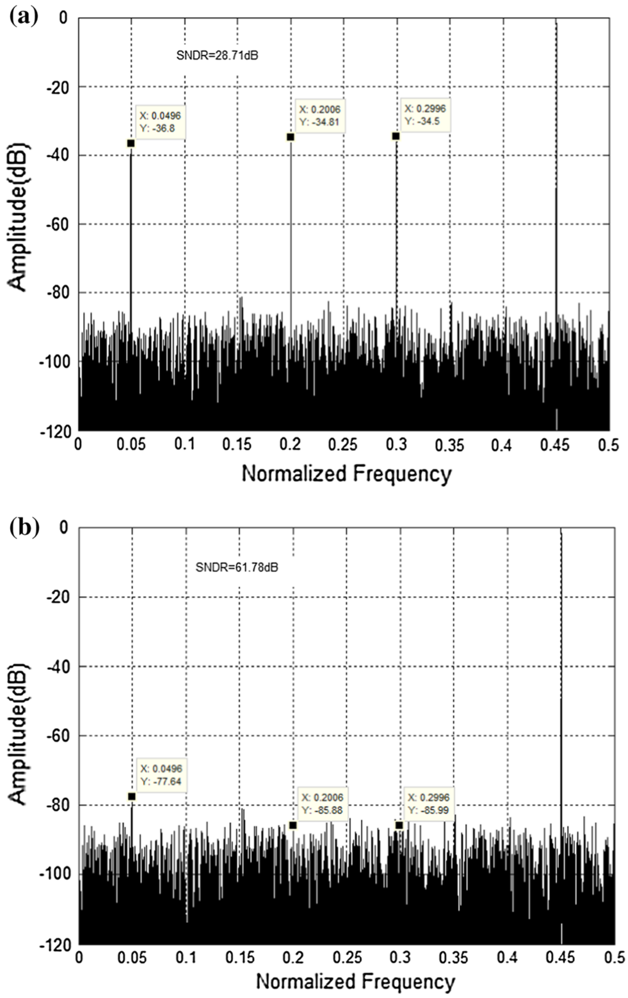


Fig. 12 Output spectrum at $f_{in} = 0.45f_s$: **a** before calibration, and **b** after calibration

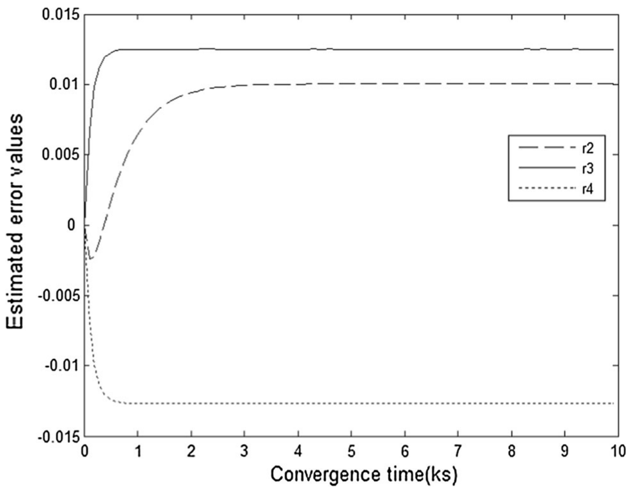


Fig. 13 Convergence of the estimated error values

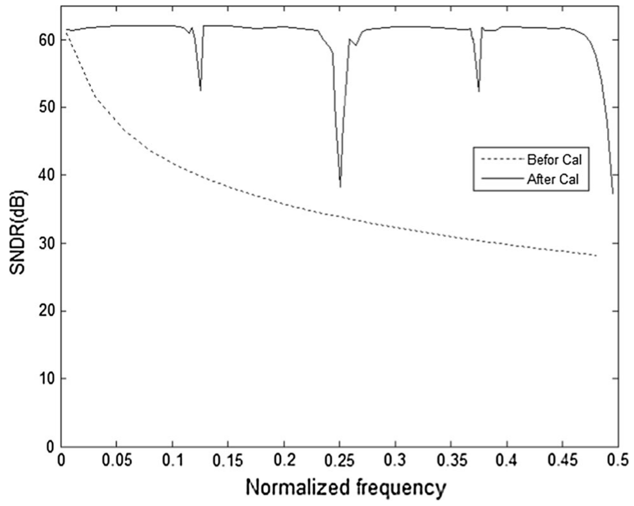


Fig. 14 SNDR performance before and after calibration

Table 3 Comparision with state-of-the-art algorithms

	This work	[14]	[4]	[19]
Resolution (bit)	10	8	8	12
Channel number	4	8	4	8
Multipliers	40	59	45	160
Convergence time (k samples)	3	150	10	–
SNDR at $0.1f_s$	62	48	50	68
SNDR at $0.45f_s$	61.78	35	42	50
SFDR at $0.1f_s$	82.55	62	67	–
SFDR at $0.45f_s$	77.64	40	50	–

It is also important to note that there is no need for online coefficient computation in the proposed architecture while other algorithms require complicated computations for the calculation of coefficients. For instance, [40] needs to compute the inverse matrix of three 7 × 7 matrices for updating the coefficients. In the proposed architecture, multipliers and adders are operating in a $\frac{f_s}{2}$ frequency while in traditional methods, multipliers and adders operate at a f_s frequency. Moreover, there is no need for the input signal to be in the Nyquist band.

6 Conclusion

Sample time error degrades the performance of TIADCs. A low computation yet effective digital background algorithm based on Lagrange interpolator is proposed in this work to correct the timing error of a four-channel TIADC. A simplified representation of the coefficients of the interpolator decreases the computation cost dramatically as well as the convergence time. The proposed algorithm can suppress error tones in high frequencies as well as low frequencies.

The proposed hardware architecture can be used as a post-processing algorithm in the host processors of a data acquisition systems or may be used as a standalone block.

References

1. Wepman, J. A., & Hoffman, J. R. (1996). *RF and IF digitization in radio receivers: Theory, concepts, and examples*. US Department of Commerce, National Telecommunications and Information Administration.
2. Vernhes, J.-A., Chabert, M., Lacaze, B., Lesthievent, G., Baudin, R., & Boucheret, M.-L. (2016). Blind estimation of unknown time delay in periodic non-uniform sampling: Application to desynchronized time interleaved-ADCs. In *IEEE international conference on acoustics, speech and signal processing (ICASSP), 2016* (pp. 4478–4482). IEEE.
3. El-Chammas, M., & Murmann, B. (2011). A 12-GS/s 81-mW 5-bit time-interleaved flash ADC with background timing skew calibration. *IEEE Journal of Solid-State Circuits*, *46*(4), 838–847.
4. Chen, H., Pan, Y., Yin, Y., & Lin, F. (2017). All-digital background calibration technique for timing mismatch of time-interleaved ADCs. *Integration, the VLSI Journal*, *57*, 45–51.
5. Jamal, S. M., Fu, D., Singh, M. P., Hurst, P. J., & Lewis, S. H. (2004). Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter. *IEEE Transactions on Circuits and Systems-I-Regular Papers*, *51*(1), 130–139.
6. Razavi, B. (2013). Design considerations for interleaved ADCs. *IEEE Journal of Solid-State Circuits*, *48*(8), 1806–1817.
7. Abbaszadeh, A., & Dabbagh-Sadeghipour, K. (2009). A new FPGA-based postprocessor architecture for channel mismatch correction of time interleaved ADCS. In *IEEE workshop on signal processing systems, 2009. SiPS 2009* (pp. 202–207). IEEE.
8. Abbaszadeh, A., & Dabbagh-Sadeghipour, K. (2010). An efficient postprocessor architecture for channel mismatch correction of time interleaved ADCs. In *18th Iranian conference on electrical engineering (ICEE), 2010* (pp. 382–385). IEEE.
9. Qin, G.-J., Liu, G.-M., Gao, M.-G., Fu, X.-J., & Xu, P. (2014). Correction of sample-time error for time-interleaved sampling system using cubic spline interpolation. *Metrology and Measurement Systems*, *21*(3), 485–496.
10. Reyes, B. T., Sanchez, R. M., Pola, A. L., & Hueda, M. R. (2017). Design and experimental evaluation of a time-interleaved ADC calibration algorithm for application in high-speed communication systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *64*(5), 1019–1030.
11. Jamal, S. M., Fu, D., Chang, N. C.-J., Hurst, P. J., & Lewis, S. H. (2002). A 10-b 120-Msample/s time-interleaved analog-to-digital converter with digital background calibration. *IEEE Journal of Solid-State Circuits*, *37*(12), 1618–1627.
12. Law, C. H., Hurst, P. J., & Lewis, S. H. (2010). A four-channel time-interleaved ADC with digital calibration of interchannel timing and memory errors. *IEEE Journal of Solid-State Circuits*, *45*(10), 2091–2103.
13. Razavi, B. (2012). Problem of timing mismatch in interleaved ADCs. In *CICC* (pp. 1–8).
14. Li, D., Zhu, Z., Zhang, L., & Yang, Y. (2016). A background fast convergence algorithm for timing skew in time-interleaved ADCs. *Microelectronics Journal*, *47*, 45–52.
15. Yu, B., Chen, C., Ye, F., & Ren, J. (2013). A mixed sample-time error calibration technique in time-interleaved ADCs. *IEICE Electronics Express*, *10*(24), 20130882.
16. Yi, R., Wu, M., Asami, K., Kobayashi, H., Khatami, R., Katayama, A., Shimizu, I., & Katoh, K. (2013). Digital compensation for timing mismatches in interleaved ADCs. In *2013 22nd Asian test symposium* (pp. 134–139). IEEE.
17. Wang, Z., Guo, L., Tian, S., & Liu, T. (2014). Estiamtion and correction of mismatch errors in time-interleaved ADCs. *Journal of Electronic Testing*, *30*(5), 629–635.
18. Shahmansoori, A. (2015). Adaptive blind calibration of timing offsets in a two-channel time-interleaved analog-to-digital converter through lagrange interpolation. *Signal, Image and Video Processing*, *9*(5), 1047–1054.
19. Schmidt, C. A., Cousseau, J. E., Figueroa, J. L., Reyes, B. T., & Hueda, M. R. (2016). Efficient estimation and correction of mismatch errors in time-interleaved ADCs. *IEEE Transactions on Instrumentation and Measurement*, *65*(2), 243–254.
20. Pillai, A. K. M., & Johansson, H. (2013). Efficient signal reconstruction scheme for m-channel time-interleaved ADCs. *Analog Integrated Circuits and Signal Processing*, *77*(2), 113–122.
21. Vogel, C., Pammer, V., & Kubin, G. (2005). A novel channel randomization method for time-interleaved adcs. In *2005 IEEE instrumentation and measurement technology conference proceedings* (Vol. 1, pp. 150–155). IEEE.
22. McNeill, J., Coln, M. C., & Larivee, B. J. (2005). "Split ADC" architecture for deterministic digital background calibration of a 16-bit 1-MS/s ADC. *IEEE Journal of Solid-State Circuits*, *40*(12), 2437–2445.
23. McNeill, J. A., David, C., Coln, M., & Croughwell, R. (2009). "Split ADC" calibration for all-digital correction of time-interleaved ADC errors. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *56*(5), 344–348.
24. Kumar, R., & Gupta, R. (2014). Design and considerations of ADC0808 as interleaved ADCs, arXiv preprint. arXiv:1404.6040.
25. Chen, V. H.-C., & Pileggi, L. (2014). A 69.5 Mw 20 GS/s 6b time-interleaved ADC with embedded time-to-digital calibration in 32 nm CMOS SOI. *IEEE Journal of Solid-State Circuits*, *49*(12), 2891–2901.
26. Nakamura, Y., & Oshima, T. (2014). A 1-GS/s 11.5-ENOB time-interleaved ADC with fully digital background calibration. In *IEEE international symposium on circuits and systems (ISCAS), 2014* (pp. 1332–1335). IEEE.
27. Bazrafshan, A., Taherzadeh-Sani, M., & Nabki, F. (2013). A low-complexity digital background calibration of sample-time error in time-interleaved A/D converters. *Analog Integrated Circuits and Signal Processing*, *76*(2), 245–249.
28. Liu, S. J., Qi, P. P., Wang, J. S., Zhang, M. H., & Jiang, W. S. (2014). Adaptive calibration of channel mismatches in time-interleaved ADCs based on equivalent signal recombination. *IEEE Transactions on Instrumentation and Measurement*, *63*(2), 277–286.
29. Liu, S., Lv, N., Ma, H., & Zhu, A. (2017). Adaptive semiblind background calibration of timing mismatches in a two-channel time-interleaved analog-to-digital converter. *Analog Integrated Circuits and Signal Processing*, *90*(1), 1–7.
30. Lei, Q., Zheng, Y., Zhu, D., & Siek, L. (2014). A statistic based time skew calibration method for time-interleaved ADCs. In *IEEE international symposium on circuits and systems (ISCAS), 2014* (pp. 2373–2376). IEEE.
31. Elbornsson, J., Gustafsson, F., & Eklund, J.-E. (2005). Blind equalization of time errors in a time-interleaved ADC system. *IEEE Transactions on Signal Processing*, *53*(4), 1413–1424.
32. Johansson, H., & Löwenborg, P. (2002). Reconstruction of nonuniformly sampled bandlimited signals by means of digital

fractional delay filters. *IEEE Transactions on Signal Processing*, *50*(11), 2757–2767.

33. Namgoong, W. (2002). Finite-length synthesis filters for non-uniformly time-interleaved analog-to-digital converter. In *IEEE international symposium on circuits and systems, 2002. ISCAS 2002* (Vol. 4, pp. IV–815). IEEE.
34. Jin, H., & Lee, E. K. (2000). A digital-background calibration technique for minimizing timing-error effects in time-interleaved ADCs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *47*(7), 603–613.
35. Tsui, K. M., & Chan, S. (2014). A novel iterative structure for online calibration of M-channel time-interleaved ADCs. *IEEE Transactions on Instrumentation and Measurement*, *63*(2), 312–325.
36. Wei, H., Zhang, P., Sahoo, B. D., & Razavi, B. (2014). An 8 bit 4 GS/s 120 Mw CMOS ADC. *Journal of Solid-State Circuits*, *49*(8), 1751–1761.
37. Diaz-Carmona, J., & Dolecek, G. J. (2011). *Fractional delay digital filters*. INTECH Open Access Publisher.
38. Majdinasab, E., & Farshidi, E. (2014). A new approach for digital calibration of timing-mismatch in four-channels time-interleaved analog-to-digital converters. *Journal of Instrumentation*, *9*(09), T09001.
39. 7 series dsp48e1 slice user guide (v1.8), Xilinx, Inc, technical report UG479, November 2014. (Online). http://www.xilinx.com/support/documentation/usern guides/ug479n 7Series DSP48E1.pdf.
40. Xu, S., Lim, Y. C., & Lee, J. W. (2016). Recursive filters for time-interleaved ADC mismatch compensation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *63*(6), 848–858.

Asgar Abbaszadeh received the B.Sc. degree from the Malek-Ashtar University of Technology, Iran, in 2002 and the M.Sc. degree from University of Semnan, Semnan, Iran in 2005, both in electronic engineering. He is a Ph.D. candidate at the Sahand University of Technology. Currently, He is a lecturer at the Technical and Vocational University. His work is related to digital hardware design (embedded systems) for digital signal processing.

Esmaeil N. Aghdam was born in Zonouz, Iran, in 1964. He received the B.E. degree from University of Sistan and Baluchestan, Zahedan, Iran, in 1990, and the M.S. degree from Amir-Kabir University of Technology, Tehran, Iran, in 1994, both in electronic engineering. In 1995, he joined the Department of Electric Engineering at Sahand University of Technology, Tabriz, Iran, as a Lecturer. In 2002, he started his Ph.D. program dealing with a high-performance bandpass Delta Sigma ADC. The research program is directed by Prof. P. Benabes at SUPELEC, France. Esmaeil NAJAFI AGHDAM is now an Associate Professor at Sahand University of Technology. His current research interests include mixed mode electronic circuits, Delta Sigma Converters, RFIC design, Ultrasonic circuits and Electronic measurement.

Alfredo Rosado-Muñoz received the M.Sc. and Ph.D. degrees in physics from the University of Valencia, Spain, in 1994 and 2000, respectively. He is a member of International Federation of Automatic Control (IFAC). Currently, he is professor at the Department of Electronic Engineering, University of Valencia. His work is related to automation systems, digital hardware design (embedded systems) for digital signal processing and control algorithms, especially targeted for biomedical engineering, and bioinspired and neuromorphic systems.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.