"Split ADC" Calibration for All-Digital Correction of Time-Interleaved ADC Errors

John A. McNeill, *Member, IEEE*, Christopher David, *Student Member, IEEE*, Michael Coln, *Member, IEEE*, and Rosa Croughwell, *Member, IEEE*

***Abstract*—The "split analog-to-digital converter (ADC)" architecture enables fully digital calibration and correction of offset, gain, and aperture-delay mismatch errors in time-interleaved ADCs. The calibration of M interleaved ADCs requires $2M + 1$ half-sized ADCs, a minimal increase in analog complexity. Each conversion is performed by a pair of half-sized ADCs, generating two independent outputs that are digitally corrected using estimates of offset, gain, and aperture-delay errors. The ADC outputs are averaged to produce the ADC output code. The difference of the outputs is used in a calibration algorithm that estimates the error in the correction parameters. Any nonzero difference drives a least-mean-square feedback loop toward zero difference, which can only occur when the average error in each correction parameter is zero. A simulation of a 4 : 1-time-interleaved 16-bit 12-MSps successive-approximation-register ADC shows calibration convergence within 400 000 samples.**

***Index Terms*—Adaptive systems, analog-digital conversion, calibration, digital background calibration, mixed analog-digital integrated circuits, self-calibrating, time-interleaving.**

I. Introduction

SCALING of CMOS device dimensions offers clear advantages for digital circuitry in terms of density, speed, and integration. For mixed-signal design, scaling offers the possibility of not only increased speed but also difficulties associated with reduced supply voltage, degraded transistor characteristics, and increased variability. For Nyquist-rate analog-to-digital converters (ADCs), the tradeoffs among the sampling rate, power, dynamic range, and die area are complex and interrelated. An additional dimension in the designer's tradeoff space is the ADC architecture: depending on the required speed and dynamic range, approaches such as flash, pipeline, or successive approximation register (SAR) may be considered.

In very high speed applications, interleaving converters is an effective way to substantially increase the sampling rate f_S in the presence of technology-imposed speed limitations. However, the performance of interleaved ADCs is limited by offset, gain, and aperture-delay mismatch between channels. As CMOS devices enter the nanoscale region, issues such as matching in the presence of increased variability become more critical [1], necessitating some form of calibration.

Manuscript received October 29, 2008; revised February 3, 2009. Current version published May 15, 2009. This work was supported by in part by the National Science Foundation under Award 0523996 and in part by Analog Devices, Inc. This paper was recommended by Associate Editor K.-P. Pun.

J. McNeill and C. David are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: cdavid@ece.wpi.edu).

M. Coln and R. Croughwell are with Analog Devices, Inc., Wilmington, MA 01887 USA.

Digital Object Identifier 10.1109/TCSII.2009.2019161

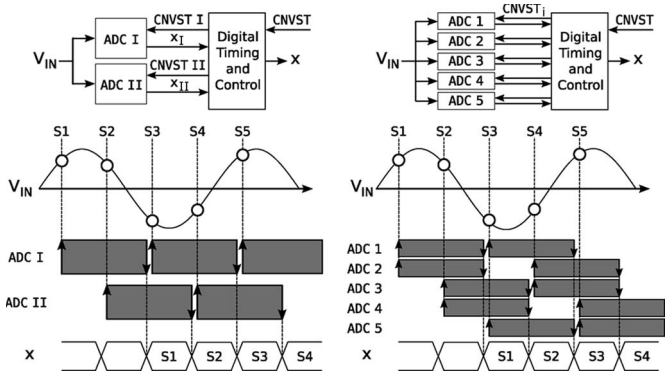

Fig. 1. (a) Time-interleaved converter. (b) "Split ADC" interleaved converter.

This brief presents an application of the "split ADC" [2], [3] architecture to the problem of calibrating offset, gain, and aperture-delay mismatch errors in interleaved ADCs. Compared with the work in [3], the novel content in this brief is the calibration of more error coefficients rather than the two described in the cyclic converter. A comparison with other calibration techniques, as described in [2], shows how the split ADC approach exploits the advantages of CMOS scaling by performing all calibration and correction in the digital domain; requirements on analog circuitry are relaxed so that degraded accuracy can be tolerated. The calibration algorithm operates in the background with no undue requirements on input signal behavior, and the speed of calibration adaptation is sufficient to track parameter variations due to environmental change.

Section II reviews issues associated with errors in interleaved ADCs, as well as analog and digital methods of correcting errors. Section III describes the error calibration process using the split ADC architecture. Section IV describes the design of a 16-bit 12-MSps 4 : 1-interleaved SAR ADC in a 250-nm CMOS process, with simulation results at both the behavioral and extracted silicon levels.

II. Time-Interleaved ADC Review

A. Overview

In a time-interleaved ADC, input v_{IN} is applied to M converters operated in parallel, as shown in Fig. 1(a) for $M = 2$, a 2 : 1-interleaved ADC. Circles indicate samples of the analog input signal spaced at intervals of $1/f_S$, where f_S is the sample rate of the entire converter. The advantage of interleaving is that each converter can be operated at a lower speed of f_S/M since the conversion operations for the ADC subchannels are spaced at intervals of $360^\circ/M$ in phase.

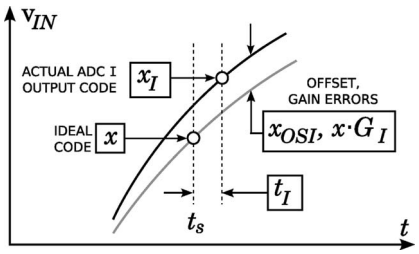

Fig. 2. Interleaved ADC error model.

The digital timing and control block plays a key role in the interleaving process, as can be seen by considering the timing diagram in Fig. 1(a). The usage of each ADC in time for the conversion process is indicated by the shaded rectangles in the figure. The rising arrow at the rectangle leading edge indicates the sampling of the input waveform and the start of the conversion process as controlled by timing signals *CNVSTI* and *CNVSTII*, which are developed by the digital block from the input *CNVST* signal at the full rate f_S. The falling arrow at the shaded rectangle trailing edge indicates the completion of the conversion process and the availability of ADC data x_I and x_{II}, which are multiplexed together, giving an output data flow x at the full rate f_S. The relationship between input samples $S1, S2, \ldots$ and the corresponding digital outputs is also indicated in the figure; note that the latency is determined by the conversion time of the individual subchannels.

B. Error Sources and Modeling

Mismatch among the interleaved ADC channels leads to errors in the digital output. Fig. 2 shows three major errors affecting an interleaved ADC operation: offset, gain, and aperture delay. Defining x as the correct ADC output code that would result from sampling the input v_{IN} at the correct time t_s, the actual output x_I will be affected by the following error contributions.

1) *Offset*: An offset error of x_{OSI} is added to each output.
2) *Gain*: Each output is multiplied by $(1 + G_I)$, reflecting a scale factor error of G_I.
3) *Aperture delay*: The actual sampling instant is displaced in time by the ADC aperture delay t_I, adding an error to the output $t_I(dx/dt)$ proportional to the input dv_{IN}/dt.

Using this model, we represent the output code x_I of each subchannel as the sum of an ideal code and the error terms

$$x_I = x + x_{OSI} + xG_I + \dot{x}t_I \tag{1}$$

where $\dot{x} = dx/dt$.

In the frequency domain, offset mismatch leads to periodic image spurs, while gain mismatch and aperture-delay mismatch lead to amplitude and phase modulation artifacts. A wide variety of techniques [4]–[8], [10], both analog and digital, have been proposed in the literature; a full review is beyond the scope of this brief. In general, analog techniques reduce the error term contributions in (1) to acceptable levels using additional circuitry, known input test signals, and/or offline calibration modes of operation. Examples of analog techniques that have been used to mitigate offset and gain include adjustable bias settings and comparator offsets [11] and input chopping [9].

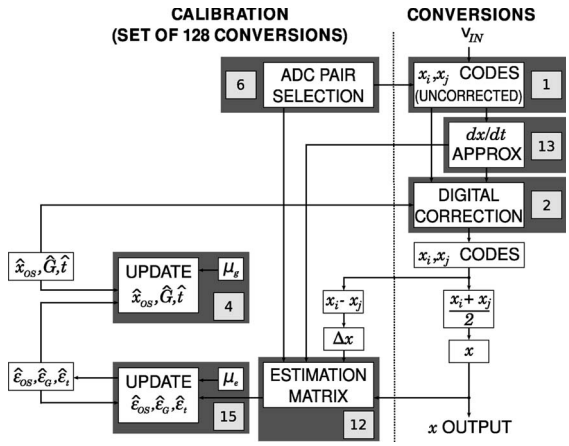

Fig. 3. Error estimation algorithm.

The aperture-delay mismatch error can be eliminated using a front-end master track and hold operating at the system's full sampling rate [9], [10]; however, this may not be feasible, depending on the power, speed, and noise penalty of the additional analog circuitry [12]. Another approach to the timing mismatch problem is presented in [1], which uses a known input test signal to correlate timing mismatch information in the background.

III. Digital Error Correction and Calibration

A. Digital Error Correction

An alternative to analog techniques is the correction of errors in the digital domain, which takes advantage of the strengths of aggressively scaled CMOS technologies. Fig. 3 shows a block diagram of the calibration and correction algorithm in this work. The shaded boxes indicate the calculations implemented and relevant equations from the following analysis. The figure is partitioned into calculations that occur in every conversion for digital correction (right side) and those that occur in the background updating calibration parameters (left side). Digital correction is implemented using estimated digital values $\hat{x}_{OSI}$, $\hat{G}_I$, and $\hat{t}_I$ representing the analog error sources; the corrected code $\hat{x}_I$ is determined by simply subtracting the error terms from the uncorrected output code x_I

$$\hat{x} = x_I - \hat{x}_{OSI} - x_I\hat{G}_I - \dot{x}_I\hat{t}_I. \tag{2}$$

Substituting (2) into (1) gives

$$\hat{x} = x + \underbrace{x_{OSI} - \hat{x}_{OSI}}_{\varepsilon_{OSI}} + x_I \underbrace{(G_I - \hat{G}_I)}_{\varepsilon_{GI}} + \dot{x}_I \underbrace{(t_I - \hat{t}_I)}_{\varepsilon_{tI}} \tag{3}$$

showing that the error in the corrected output code x is the sum of errors in the digital estimates of offset, gain, and aperture-delay errors represented by ε_{OSI}, ε_{GI}, and ε_{tI}. From (3), we see that correction can be achieved if the estimate errors are small enough. Thus, the task of the digital calibration algorithm is to determine $\hat{x}_{OSI}$, $\hat{G}_I$, and $\hat{t}_I$ to sufficient accuracy.

B. LMS Calibration Loop

The estimated digital values $\hat{x}_{OSI}$, $\hat{G}_I$, and $\hat{t}_I$ used for correction are continuously updated in the background by an

LMS feedback loop. For example, the estimated gain error for the ith ADC is updated by

$$G_{(i)}^{(\text{new})} = G_{(i)}^{(\text{old})} - \mu_g \varepsilon_{G(i)}. \tag{4}$$

The LMS loop uses $\hat{\varepsilon}_{OS(i)}$, $\hat{\varepsilon}_{G(i)}$, and $\hat{\varepsilon}_{t(i)}$, which are estimated errors in the $\hat{x}_{OS(i)}$, $\hat{G}_{(i)}$, and $\hat{t}_{(i)}$ values. The LMS multiplier μ_g controls the dynamics of the loop convergence. A key advantage of the LMS approach is that the $\hat{\varepsilon}_{OS(i)}$, $\hat{\varepsilon}_{G(i)}$, and $\hat{\varepsilon}_{t(i)}$ need not be accurate; all that is required is that they are of zero bias and (on the average) steer the convergence of (4) in the correct direction. The general LMS technique is well established [12], [15]; the novel contributions of this work are in the use of the "split ADC" approach to extract calibration information in the background, as will now be described.

C. Split ADC Applied to Time-Interleaved Converters

To apply the split ADC approach to an interleaved ADC, it is necessary to split each ADC into half-sized sub-ADCs, as shown in Fig. 1(b). The generalization of the split ADC concept is to use all possible pairs of sub-ADCs when processing the input. As shown in the example in the figure, sample S1 is processed by a split ADC composed of ADCs 1 and 2, sample S2 uses ADCs 3 and 4, and so on. In each case, the average of whichever two ADCs are used is reported as the output code, and the difference is used for the calibration signal. From the resulting differences, it will be possible to estimate the gain, offset, and aperture-delay mismatch errors of each ADC.

Note that the choice of ADC pairs is constrained by the time overlap of interleaving. For example, in the timing diagram in Fig. 1(b), when ADCs 1 and 2 complete the conversion of sample S1, converters 3 and 4 are still busy converting sample S2. To provide timing flexibility so that all possible pair permutations can be used, one additional split channel is necessary. The need for the extra half-ADC does impose a small die-area penalty (a fractional increase of $1/2M$ for an interleaving factor of M ADCs) but imposes no power penalty since there is always one of the sub-ADCs that is not used and need not be powered.

To begin the development of the calibration algorithm, consider the difference Δx of the outputs of ADCs 1 and 2. For simplicity, we assume for the moment that offset is the only error source. Using (3), we have for the conversion of sample S1

$$\Delta x_{1;2} = \hat{x}_1 - \hat{x}_2 = \varepsilon_{OS1} - \varepsilon_{OS2}. \tag{5}$$

After a set of ten conversions covering all possible pairings, we can express the results in matrix form as

$$\overbrace{\begin{bmatrix} \Delta x_{1;2} \\ \Delta x_{3;4} \\ \Delta x_{5;1} \\ \Delta x_{2;3} \\ \Delta x_{4;5} \\ \Delta x_{1;3} \\ \Delta x_{2;4} \\ \Delta x_{3;5} \\ \Delta x_{1;4} \\ \Delta x_{2;5} \end{bmatrix}}^{\mathbf{\Delta}} = \overbrace{\begin{bmatrix} +1 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 \\ -1 & 0 & 0 & 0 & +1 \\ 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & +1 & -1 \\ +1 & 0 & -1 & 0 & 0 \\ 0 & +1 & 0 & -1 & 0 \\ 0 & 0 & +1 & 0 & -1 \\ +1 & 0 & 0 & -1 & 0 \\ 0 & +1 & 0 & 0 & -1 \end{bmatrix}}^{\mathbf{S}} \overbrace{\begin{bmatrix} \varepsilon_{OS1} \\ \varepsilon_{OS2} \\ \varepsilon_{OS3} \\ \varepsilon_{OS4} \\ \varepsilon_{OS5} \end{bmatrix}}^{\mathbf{e}}. \tag{6}$$

The "selection matrix" $\mathbf{S}$ defines which sub-ADC channels were selected for use in each conversion, as well as the sign of that channel's contribution to the Δx term. The task of the calibration algorithm is to determine the values of the estimate errors in $\mathbf{e}$ given the observed Δx values in $\mathbf{\Delta}$.

Consider the example of finding the $\varepsilon_{OS(1)}$ value. Intuitively, the Δx values from conversions involving ADC 1 have the most information about $\varepsilon_{OS(1)}$; from column 1 of the $\mathbf{S}$ matrix, we see that the conversions of rows 1, 3, 6, and 9 used ADC 1. We collect this information by adding the corresponding rows, inverting where indicated by the sign of the term in column 1 of $\mathbf{S}$, to give

$$\Delta x_{1;2} + \Delta x_{1;3} + \Delta x_{1;4} + \Delta x_{1;5} = 4\varepsilon_{OS1} - \varepsilon_{OS2} - \varepsilon_{OS3} - \varepsilon_{OS4} - \varepsilon_{OS5}. \tag{7}$$

Repeating for each column, we see that this is equivalent to matrix multiplication by the transpose of $\mathbf{S}$, which gives

$$\mathbf{S^T\Delta} = \overbrace{\begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix}}^{\mathbf{S^TS}} \overbrace{\begin{bmatrix} \vdots \\ \varepsilon_{OS(i)} \\ \vdots \end{bmatrix}}^{\mathbf{e}}. \tag{8}$$

Matrix $\mathbf{S^TS}$ in (6) is singular in this case. This intuitively makes sense: we cannot calibrate offset errors in an absolute sense, only offset differences. To resolve this problem, we allow the overall ADC offset to equal the average offset of the subchannel ADCs; we only correct the offset mismatch. This gives an additional constraint that the sum of the offset errors for all ADCs must be zero, adding a line to the $\mathbf{S^TS}$ matrix

$$\begin{bmatrix} \vdots \\ \mathbf{S^T\Delta} \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \overbrace{\begin{bmatrix} \vdots \\ \varepsilon_{OS(i)} \\ \vdots \end{bmatrix}}^{\mathbf{e}}. \tag{9}$$

Adding the new row in (9) to each of the others gives

$$\mathbf{S^T\Delta} = \overbrace{\begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}}^{\mathbf{T}} \overbrace{\begin{bmatrix} \vdots \\ \varepsilon_{OS(i)} \\ \vdots \end{bmatrix}}^{\mathbf{e}} \tag{10}$$

which is easily solved for $\varepsilon_{OS(i)}$. Note that the digital hardware requirements of this approach are modest: each row of $\mathbf{S^T\Delta}$ is an accumulation of Δx values in a similar fashion to the left side of (4). Row i is just the sum of the Δx values resulting each time the ith ADC is selected. Since $\mathbf{T}$ is diagonal, it need not be stored at all; the LMS update in (4) need not be precise, and the division by the diagonal element in $\mathbf{T}$ can be incorporated into the μ_g factor.

The key insight from (10) is that by adding the relevant Δx values for each ADC as determined by the $\mathbf{S}$ matrix, the resulting sum is dominated by that ADC's error term.

Extending to the results including gain and aperture-delay errors, we generalize (6) to

$$\underset{\substack{\uparrow\\ 128\\ \text{CONVS}\\ \downarrow}}{} \overbrace{\begin{bmatrix}\vdots\\ \Delta x_{i;j}\\ \vdots\end{bmatrix}}^{\mathbf{\Delta}} = \overbrace{\begin{bmatrix}\vdots & \vdots & \vdots\\ \mathbf{S} & \mathbf{x} & \hat{\mathbf{x}}\\ \vdots & \vdots & \vdots\end{bmatrix}}^{\mathbf{B}} \overbrace{\begin{bmatrix}\varepsilon_{OS(i)}\\ \varepsilon_{G(i)}\\ \varepsilon_{t(i)}\end{bmatrix}}^{\mathbf{e}} \tag{11}$$

where $\mathbf{x}$ and $\dot{\mathbf{x}}$ in matrix $\mathbf{B}$ represent the output code and estimated derivative values. In a similar fashion to the process that led to (10), we collect information corresponding to each ADC's error terms. The background calibration process operates over a subset of 128 conversions; the value of 128 was chosen as a compromise between the update rate and the averaging out of noise in calibration parameters.

Unlike the entries in $\mathbf{S}$, which are restricted to $\{-1, 0, +1\}$, the x and $\dot{x}$ values can take on any value between -1 and 1; therefore, we use the sgn function when accumulating the Δx values

$$[\mathrm{sgn}(\mathbf{B})]^{\mathbf{T}} \begin{bmatrix}\vdots\\ \Delta x_{i;j}\\ \vdots\end{bmatrix} = \overbrace{[\mathrm{sgn}(\mathbf{B})]^{\mathbf{T}}\mathbf{B}}^{\mathbf{C}} \begin{bmatrix}\varepsilon_{OS(i)}\\ \varepsilon_{G(i)}\\ \varepsilon_{t(i)}\end{bmatrix}. \tag{12}$$

The $\mathrm{sgn}()$ function reduces digital complexity: multiplications involving the $[\mathrm{sgn}(\mathbf{B})]^{\mathbf{T}}$ matrix are simple since its values are restricted to $\{-1, 0, +1\}$, and in a similar fashion to (10), the coefficient matrix $\mathbf{C}$ is (for most signals) dominated by its diagonal elements. Sections III-D and III-E describe issues involved with the derivative estimation and operation of the LMS loop.

D. Derivative Estimate

As shown in (11) in Section III-C, an estimated value of the derivative must be used since the true value of dv_{IN}/dt is unknown. The simplest estimate of the derivative is from adjacent samples of v_{IN}

$$\hat{\dot{x}}[n] = \frac{\hat{x}[n+1] - \hat{x}[n-1]}{2} \tag{13}$$

which adds a latency of one additional conversion time. The magnitude of the error in the corrected output code x from using the approximate derivative depends on the frequency content of v_{IN}. The first-order derivative estimate may prove impractical with input frequencies close to the Nyquist rate. A higher order approximation may be required at the cost of additional latency. As an alternative, an analog approach is to use a single high-speed front-end track and hold stage to eliminate aperture-delay mismatch; then, only offset and gain errors need to be corrected. For dc signals, the derivative estimation coefficients in $\mathbf{C}$ approach zero, effectively disabling the aperture-delay calibration, which is not needed for dc signals.

E. LMS Loop Operation

An exact solution of (12) would require matrix inversion, which is both computationally intensive and prone to numerical problems. An exact solution is also unnecessary, since the LMS method does not need an accurate error estimate. Simulation shows that the matrix $\mathbf{C}$ in (12) is diagonally dominant for "active" signals that span most of the input range. In a similar fashion to (10), we take advantage of the relaxed accuracy requirements of the LMS algorithm to avoid any calculations involving $\mathbf{C}$ and approximate $\mathbf{e}$ with

$$\begin{bmatrix}\varepsilon_{OS(i)}\\ \varepsilon_{G(i)}\\ \varepsilon_{t(i)}\end{bmatrix} = \mu_e [\mathrm{sgn}(\mathbf{B})]^{\mathbf{T}} \begin{bmatrix}\vdots\\ \Delta x_{i;j}\\ \vdots\end{bmatrix}. \tag{14}$$

The result of (14) is recalculated for every set of 128 conversions. To average out the errors over a larger range of conversion information, we adopt the approximate matrix solution in [3], which applies an LMS iteration to the error estimation of $\mathbf{e}$. Applying the method in [3] and expanding the matrix multiplication of (14) for each of the components of $\mathbf{e}$ gives

$$\begin{aligned}
\varepsilon_{OS(i)}^{(\mathrm{new})} &= (1-\mu_e)\varepsilon_{OS(i)}^{(\mathrm{old})} + \mu_e \left([\mathrm{sgn}(\mathbf{S})]^{\mathbf{T}} \mathbf{\Delta}\right)\\
\varepsilon_{G(i)}^{(\mathrm{new})} &= (1-\mu_e)\varepsilon_{G(i)}^{(\mathrm{old})} + \mu_e \left([\mathrm{sgn}(\mathbf{x})]^{\mathbf{T}} \mathbf{\Delta}\right)\\
\varepsilon_{t(i)}^{(\mathrm{new})} &= (1-\mu_e)\varepsilon_{t(i)}^{(\mathrm{old})} + \mu_e \left(\left[\mathrm{sgn}(\hat{\dot{\mathbf{x}}})\right]^{\mathbf{T}} \mathbf{\Delta}\right).
\end{aligned} \tag{15}$$

Parameter μ_e affects the dynamics of the LMS loop convergence; as in [3], we choose $\mu_e > \mu_g$ for the stability of the LMS loop.

For some input signals (for example, dc), the off-diagonal elements of $\mathbf{C}$ in (12) may be significant, which slows convergence, as will be seen in Section IV.

TABLE I
BEHAVIORAL SIMULATION PARAMETERS

PARAMETER		VALUE
Subchannel ADC	Resolution	16b
	Sample Rate	3 MSps
	SNR	90.2 dB
	INL	±1LSB
	DNL	±0.5LSB
	Bandwidth	50 MHz ±1%
Error Range	Offset	±0.1% FSR
	Gain	±0.2% FSR
	Aperture Delay	±50 ps
LMS Parameter	μ_g	1/1024
	μ_e	1/32
Internal Digital Precision		24b
Interleaved ADC	Interleaving	4:1
	Sample Rate f_S	12 MSps
	SNDR Uncorrected	34.2 dB
	SNDR Corrected	92.9 dB

IV. RESULTS

A 4 : 1-time-interleaved system based on nine AD7621 16-bit SAR converters was designed in a 0.25-μm process with added digital logic for timing and input–output [14]. This layout was extracted to a model combining SPICE and HDL and tested with a MATLAB simulation of the calibration algorithm. For comparison purposes, the HDL block of the algorithm was realized in a very large scale integration layout with an area of 0.5 mm^2, a 1% increase in die area. The simulation parameters and results are shown in Table I. Fig. 4(a) shows the fast Fourier transform of a 0.9-FSpk-amplitude two-tone input, for the uncorrected interleaved ADC, with sub-ADC pairs chosen in a repeating pattern. There are numerous spurious tones due to the

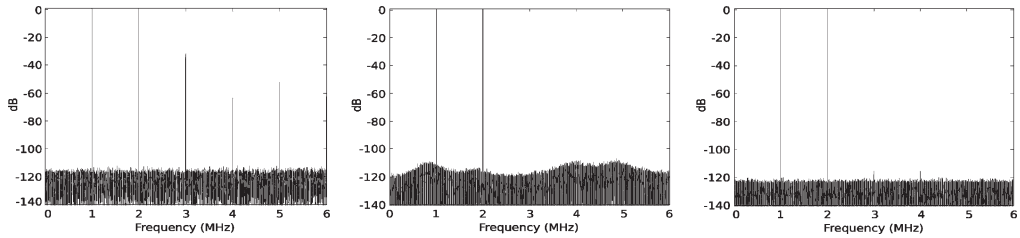

Fig. 4. (a) Uncorrected ADC output. (b) Shuffled and uncorrected output. (c) Corrected and shuffled output.

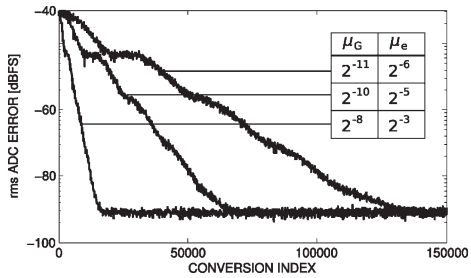

Fig. 5. ADC error convergence for different μ values.

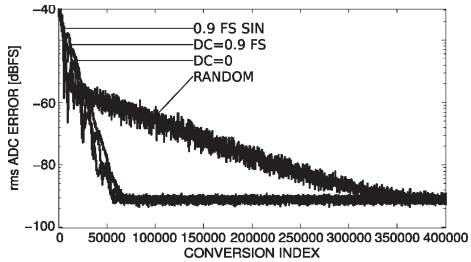

Fig. 6. ADC error convergence for sine, random, and dc inputs.

uncorrected error sources. In Fig. 4(b), errors are uncorrected, but the sub-ADC pair choices are shuffled in a pseudorandom fashion. Note that the spurious-free dynamic range (SFDR) improves since the repeating ADC error pattern is broken; however, the total error power is essentially unaffected. In Fig. 4(c), shuffling and error correction are enabled, showing a performance improvement due to the "split ADC" error calibration and correction: SFDR improves to 104.8 dB, and the signal-to-noise-and-distortion ratio (SNDR) improves to 92.9 dB. According to [2] and [3], averaging the ADC outputs should yield a 3-dB improvement over the 90.2-dB SNR of the subchannel ADC. The simulated performance is within 0.3 dB of this limit, indicating that the correction provided by the split ADC approach is functioning as expected. The harmonic spurs shown in Fig. 4(c) at 3 and 4 MHz are due to the integral-nonlinearity (INL), differential-nonlinearity (DNL), and bandwidth mismatch of the sub-ADCs. Fig. 5 shows how the time to convergence depends on the μ_g and μ_e of the LMS algorithm. The estimation parameter convergence for the different input signals is shown in Fig. 6. Here, the error settles within 1 LSB of the ideal signal within 400 000 conversions. With $f_S = 12$ MS/s, the convergence time of 33 ms is fast enough to track out parameter variations due to environmental changes.

V. Conclusion

A time-interleaved ADC concept using the split ADC architecture has been demonstrated. The offset, gain, and aperture-delay errors are digitally corrected, and calibration is continuously performed in the background with a fully digital implementation. A behavioral simulation of a 4 : 1-interleaved system with nine subchannels shows calibration convergence within 400 000 samples and corrected performance within 0.3 dB of the theoretical limit imposed by the sub-ADCs.

Acknowledgment

The authors would like to thank the members of the Precision Nyquist Converter and High-Speed Converter Groups at Analog Devices, Inc.

References

[1] E. Iroaga, B. Murmann, and L. Nathawad, "A background correction technique for timing errors in time-interleaved analog-to-digital converters," in *ISCAS*, May 2005, vol. 6, pp. 5557–5560.

[2] J. McNeill, M. Coln, and B. Larivee, "A 'split-ADC' architecture for a deterministic digital background calibration of a 16 b 1MS/s ADC," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2437–2445, Dec. 2005.

[3] J. McNeill, M. Coln, D. R. Brown, and B. Larivee, "Digital background calibration algorithm for 'Split ADC' architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 2, pp. 294–306, Feb. 2009.

[4] K. C. Dyer, D. Fu, P. J. Hurst, and S. H. Lewis, "An analog background calibration technique for time-interleaved analog-to-digital converters," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1912–1919, Dec. 1998.

[5] S. M. Jamal, D. Fu, M. P. Singh, P. J. Hurst, and S. H. Lewis, "Calibration of sample-time error in two-channel time-interleaved analog-to-digital converter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 130–139, Jan. 2004.

[6] J. Elbornsson, F. Gustafsson, and J.-E. Eklund, "Blind adaptive equalization of mismatch errors in a time-interleaved A/D converter system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 151–158, Jan. 2004.

[7] C. Vogel, "The impact of combined channel mismatch effects in time-interleaved ADCs," *IEEE Trans. Intrum. Meas.*, vol. 54, no. 1, pp. 415–427, Feb. 2005.

[8] B. Brannon and A. Barlow, "Aperture uncertainty and ADC system performance," Analog Devices, Norwood, MA, Analog Devices Application Note AN-501, 2006.

[9] Z.-M. Lee, C.-Y. Wang, and J.-T. Wu, "A CMOS 15-bit 125-MS/s time interleaved ADC with digital background calibration," *IEEE J. Solid-State Circuits*, vol. 42, no. 10, pp. 2149–2160, Oct. 2007.

[10] J. Eklund and F. Gustafsson, "Digital offset compensation of time-interleaved ADC using random chopper sampling," in *IEEE Symp. Circuits Syst.*, May 2000, pp. 447–450.

[11] S. Louwsma, A. J. M. van Tuijl, M. Vertregt, and B. Nauta, "A 1.35 GS/s, 10b, 175 mW time-interleaved AD converter in 0.13 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 778–786, Apr. 2008.

[12] Y. Chiu, C. W. Tsang, B. Nikoliæ, and P. R. Gray, "Least mean square adaptive digital background calibration of pipelined analog-to-digital converters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 38–40, Jan. 2004.

[13] J. Elbornson, F. Gustafasson, and J. E. Elkund, "Analysis of mismatch effects in a randomly interleaved A/D converter system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 465–476, Mar. 2005.

[14] R. Croughwell, "A 16-b 10 Msample/s split-interleaved analog to digital converter," M.S. thesis, Worcester Poly. Inst., Worcester, MA, 2007.

[15] B. Murmann and B. E. Boser, "A 12 b 75 MS/s pipelined ADC using open-loop residue amplification," in *ISSCC Dig. Tech. Papers*, Feb. 2003, pp. 328–329.