February 2008

Background Calibration With Piecewise Linearized Error Model for CMOS Pipeline A/D Converter

Jie Yuan
Hong Kong University of Science and Technology

Nabil H. Farhat
University of Pennsylvania, farhat@seas.upenn.edu

Jan Van der Spiegel
University of Pennsylvania, jan@seas.upenn.edu

Background Calibration With Piecewise Linearized Error Model for CMOS Pipeline A/D Converter

Abstract
A new all-digital background calibration method, using a piecewise linear model to estimate the stage error pattern, is presented. The method corrects both linear and nonlinear errors. The proposed procedure converges in a few milliseconds and requires low hardware overhead, without the need of a high-capacity ROM or RAM. The calibration procedure is tested on a 0.6- μm CMOS pipeline analog-to-digital converter (ADC), which suffers from a high degree of nonlinear errors. The calibration gives improvements of 17 and 26 dB for signal-noise-and-distortion ratio (SNDR) and spurious-free dynamic range (SFDR), respectively, for the Nyquist input signal at the sampling rate of 33 MSample/s. The calibrated ADC achieves SNDR of 70.3 dB and SFDR of 81.3 dB at 33 MSample/s, which results in a resolution of about 12 b.

Keywords
analog-to-digital converter (ADC), background calibration, CMOS ADC, nonlinear error calibration, pipeline ADC

Background Calibration With Piecewise Linearized Error Model for CMOS Pipeline A/D Converter

Jie Yuan, *Member, IEEE*, Nabil H. Farhat, *Life Fellow, IEEE*, and Jan Van der Spiegel, *Fellow, IEEE*

***Abstract*—A new all-digital background calibration method, using a piecewise linear model to estimate the stage error pattern, is presented. The method corrects both linear and nonlinear errors. The proposed procedure converges in a few milliseconds and requires low hardware overhead, without the need of a high-capacity ROM or RAM. The calibration procedure is tested on a 0.6-μm CMOS pipeline analog-to-digital converter (ADC), which suffers from a high degree of nonlinear errors. The calibration gives improvements of 17 and 26 dB for signal-noise-and-distortion ratio (SNDR) and spurious-free dynamic range (SFDR), respectively, for the Nyquist input signal at the sampling rate of 33 MSample/s. The calibrated ADC achieves SNDR of 70.3 dB and SFDR of 81.3 dB at 33 MSample/s, which results in a resolution of about 12 b.**

***Index Terms*—Analog-to-digital converter (ADC), background calibration, CMOS ADC, nonlinear error calibration, pipeline ADC.**

I. Introduction

CMOS Nyquist-rate pipeline analog-to-digital converters (ADCs) have been an active topic for analog/mixed-signal IC design. Early pipeline ADC designs [1]–[8] focused on using circuit techniques to reduce the overall error level. On the other hand, self-calibration techniques [9]–[12] are able to compensate for a part of the systematic error, so as to provide ADC designs with higher resolution at lower power consumption or at higher speed. In the early developed calibration techniques, such as that in [9], calibrations are carried out in the analog domain, which requires extra analog trimming circuits. The development of all-digital calibration techniques, such as those described in [10]–[12], leads to relaxed analog circuit design, reduction of the overall error level, and higher resolution, especially for low-power processes. In those foreground self-calibration techniques, the calibration procedure usually takes place at power-on. In order to track environmental changes, such as power supply migration, dedicated calibration cycles have to be periodically allocated, which interrupts normal operations. Hence, more desirable background calibration techniques [15]–[18] have been developed to place the calibration process outside the main signal path.

Manuscript received November 10, 2006; revised May 14, 2007. This work was supported in part by the Office of Naval Research under Grant N00014-94-1-0931 and by Army Research Office under Grant MURI DAAD-19-01-1-0603 via the Georgia Institute of Technology and Grant DURIP W911NF-04-1-0177. This paper was recommended by Associate Editor P. Carbone.

J. Yuan is with the Electronic and Computer Engineering Department, Hong Kong University of Science and Technology, Clearwater Bay, Hong Kong (e-mail: eeyuan@ust.hk).

N. H. Farhat and J. Van der Speigel are with the Electrical and System Engineering Department, University of Pennsylvania, Philadelphia, PA 19104 USA

Digital Object Identifier 10.1109/TCSI.2007.910645

All of the calibration techniques mentioned above aim to compensate for code gaps, or linear errors, which result from nonidealities, such as capacitance mismatch and finite opamp gain. Nonlinear errors can occur in many situations. In high-speed ADCs, large biasing currents are used to enable fast settling. However, the large biasing current reduces the signal swing range [22] of the opamps, especially for low-supply-voltage processes. Hence, in high-speed applications that require a wide input signal range, the output of the opamp can experience gain dispersion, which introduces nonlinear error. In other occasions, in order to save power, simple stages with small biasing current can be used [19], nonlinear error inherently exists in these stages. In addition, the dynamic settling process introduces nonlinear errors at high sampling rate. Therefore, calibrations for nonlinear errors can further reduce the error of the pipeline.

Few papers have been reported that deal with nonlinear error calibration [19]–[21]. In [19], the residual amplifier of the first stage in an ideal 14-bit pipeline is changed to an open-loop amplifier, which introduces a considerable amount of nonlinear error. Based on the open-loop model, a background calibration procedure was developed to compensate for this specific nonlinear error. Recently, another calibration scheme for nonlinear errors has been proposed [20]. The nonlinear stage gain is modeled by a second-order dependency on the output signal. A couple of major codes generated by a digital-to-analog converter (DAC) are calibrated at each stage. The method employs a recursive process to calibrate both the DAC and the ADC.

In this paper, a new nonlinear calibration procedure is introduced, which applies a piecewise linear model to estimate the stage error pattern. A brief version of this technique has been presented in [21]. The calibration procedure is general against nonlinear error patterns in the sense that it is not designed to compensate for a particular type. The all-digital calibration method can run in the background and has a fast convergence rate. The procedure requires about 20 k logic gates as calibration overhead, without the need for a high-capacity memory. To illustrate the technique, a prototype pipeline whose stages exhibited considerable nonlinear error was fabricated in a 0.6-μm CMOS process. Although the prototype design is not optimized for power, the proposed nonlinear calibration procedure can potentially lead to very simple stage designs to aggressively reduce the power consumption of the pipeline ADC.

The remainder of this paper is organized in the following five parts. In Section II, the error model of the conventional 1.5-bit pipeline multiplying digital-analog converter (MDAC) is formulated. In Section III, the new calibration method is introduced. In Section IV, the circuit design of a 14-stage pipeline ADC is described. The implementation details and experimental results are given in Section V. Section VI gives the final conclusions.

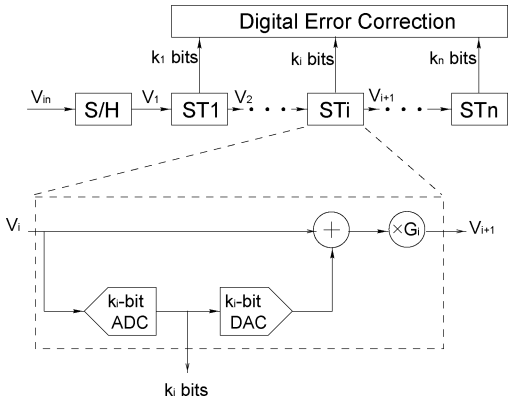


Fig. 1. Pipeline ADC with digital error correction.

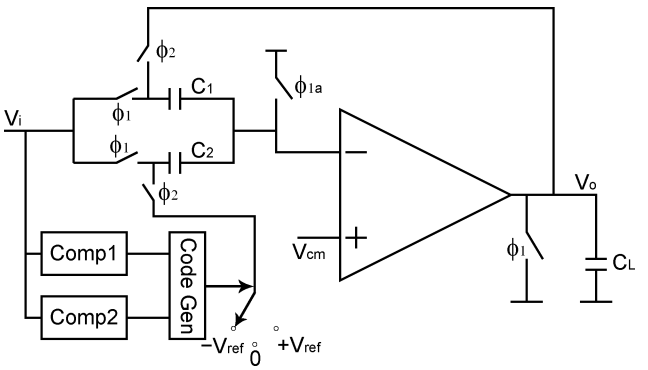


Fig. 2. Single-ended block diagram for a 1.5-bit stage.

II. Pipeline Stage Error Model

Fig. 1 shows a general pipeline ADC. Following the sample/hold (S/H) front-end is a cascade of pipeline stages. Each stage (STi) takes the residual voltage (V_i) from the previous stage as the input signal. The flash ADC inside the stage quantize V_i into k_i bits. The recovered level is subtracted from the input signal V_i and amplified by G_i to generate the residual signal V_{i+1} for the next stage. In order to reduce the complexity of the circuit design, 1.5-bit MDACs are used. The single-ended version of a conventional 1.5-bit MDAC is given in Fig. 2. The MDAC operates under two nonoverlapping clocks ϕ_1 and ϕ_2. Clock ϕ_{1a} is an early version of ϕ_1. V_i is the input signal for the MDAC, while V_o is the output residual signal for the MDAC.

At the end of the ϕ_1 phase, the differential charge stored on the capacitors are Q_{p1} and

$$Q_{p1} = (C_1 + C_2)V_i \tag{1}$$

where V_i is the input voltage. At the end of the ϕ_2 phase, the differential charge stored on the capacitors are Q_{p2} and

$$Q_{p2} = (V_d - V_x)C_2 + (V_o - V_x)C_1 \tag{2}$$

where V_o is the output voltage. V_d is the recovered analog level from the sub-DAC with the following assignment:

$$V_d = \begin{cases} V_{\text{ref}}, & \text{if } V_i \in \left(\frac{1}{4}V_{\text{ref}},\ V_{\text{ref}}\right) \\ 0, & \text{if } V_i \in \left(-\frac{1}{4}V_{\text{ref}},\ \frac{1}{4}V_{\text{ref}}\right) \\ -V_{\text{ref}}, & \text{if } V_i \in \left(-V_{\text{ref}}, -\frac{1}{4}V_{\text{ref}}\right) \end{cases}. \tag{3}$$

V_x in (2) is the offset at the negative input of the amplifier

$$V_x = \delta - \frac{V_o}{A} \tag{4}$$

where δ is the opamp's static offset, and A is the gain of the amplifier.

If the charge injections by ϕ_{1a} switches are signal independent, the differential charge conserves

$$Q_{p1} = Q_{p2}. \tag{5}$$

Hence, the fully differential equivalent transfer function for the stage in Fig. 2, upon complete settling, can be derived to

$$V_o = \frac{C_1 + C_2}{C_1 + \frac{C_1 + C_2}{A}}V_i - \frac{C_2}{C_1 + \frac{C_1 + C_2}{A}}V_d + \frac{C_1 + C_2}{C_1 + \frac{C_1 + C_2}{A}}\delta. \tag{6}$$

If a mismatch of Δ exists between the capacitors as shown in

$$C_2 = C_1(1 + \Delta) \tag{7}$$

then the transfer function can be simplified into

$$V_o = \frac{2 + \Delta}{1 + \epsilon}\left(V_i - \frac{1 + \Delta}{2 + \Delta}V_d + \delta\right) \tag{8}$$

where

$$\epsilon = \frac{2 + \Delta}{A}. \tag{9}$$

With Taylor expansion, (8) can be rewritten as

$$V_o = (2 + \Delta + g(\epsilon))\left(V_i - \frac{1 + \Delta}{2 + \Delta}V_d + \delta\right) \tag{10}$$

where

$$g(\epsilon) = -\epsilon(2 + \Delta)(1 - \epsilon + \epsilon^2 - \epsilon^3 \ldots). \tag{11}$$

In the case that the opamp gain is large, ϵ would be a small number and the effect of $g(\epsilon)$ can be neglected. Then, the dominant error source would result from the capacitance mismatch Δ, which causes linear error. In low-voltage processes, however, the opamp's gain is limited. The effect of ϵ needs to be considered. If the opamp gain A is flat within the signal range, ϵ in (9) is signal-independent. Hence, $g(\epsilon)$ in (11) can be annexed into Δ to produce an overall linear error.

However, if the opamp has gain dispersion over the signal range, ϵ would be a function of the input signal, i.e., $\epsilon(V_i)$. Then, the ϵ term will introduce nonlinear error into the stage. In the design of a high-speed opamp, the opamp specifications, such as power, bandwidth, and output signal swing, are intertwined [22]. As a result, large-signal swing cannot always be accommodated. The reduction of the allowable signal swing and reduced power supplies in the opamp will push the output transistors close to the linear regime, near the edge of the signal range, which causes a considerable amount of gain reduction of the opamp.

On the other hand, in high-speed pipelines, stages usually cannot reach complete settling, especially for large input signals, which results in longer slewing durations. As a result, the dynamic performance of the pipeline suffers worse degradation from nonlinear error. The 1.5-bit MDAC in Fig. 2 operates under two phases, sampling phase ϕ_1 and holding phase ϕ_2.

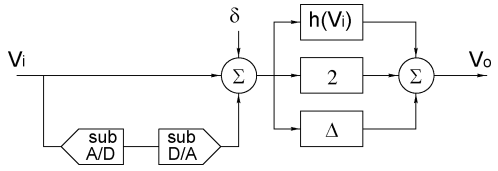


Fig. 3. Signal flow of the stage.

For slewing-limited MDAC, a considerable amount of nonlinear error arises at switches for incomplete settling during ϕ_2. If the sampling rate is too high for the common-mode of MDAC to settle during ϕ_1, the resulting nonlinear error not only depends on the current code, but also on the previous codes. Nonlinear errors of this type cannot be compensated effectively by any calibration scheme.

For an MDAC with a nonlinear error that is dependent only on the current code, a general transfer function is given in

$$V_o = (2 + \Delta + h(V_i)) \left(V_i - \frac{1+\Delta}{2+\Delta} V_d + \delta \right). \quad (12)$$

The corresponding signal flow is shown in Fig. 3.

III. Calibration Method

A. Calibration Model

Although the stage error model derived above is helpful in determining error sources, the derivation of an analytical recursive backward transfer function for a general nonlinear error $h()$ is difficult. Instead, we use the error pattern diagram for the development of our nonlinear calibration method. If we define $\mathrm{ITF}()$ as the ideal transfer function, which is given in

$$\begin{aligned} \mathrm{ITF}(V_i) &= 2V_i - V_d \\ &= \begin{cases} 2V_i - V_{\mathrm{ref}}, & \text{if } V_i \in \left(\frac{1}{4}V_{\mathrm{ref}}, V_{\mathrm{ref}}\right) \\ 2V_i, & \text{if } V_i \in \left(-\frac{1}{4}V_{\mathrm{ref}}, \frac{1}{4}V_{\mathrm{ref}}\right) \\ 2V_i + V_{\mathrm{ref}}, & \text{if } V_i \in \left(-V_{\mathrm{ref}}, -\frac{1}{4}V_{\mathrm{ref}}\right) \end{cases} \end{aligned} \quad (13)$$

and $\mathrm{STF}()$ as the real stage transfer function (12), the error pattern (EP) can be defined by subtracting the **ideal** inverse transfer function from the **real** inverse transfer function, shown in

$$\mathrm{EP}(V_i) = V_{\mathrm{real}} - V_{\mathrm{ideal}} = V_i - \mathrm{ITF}^{-1}(\mathrm{STF}(V_i)). \quad (14)$$

which is actually an input-referred error pattern.

The transfer function of an ideal 1.5-bit MDAC is shown in Fig. 4. A typical nonlinear error pattern generated by stages with the error model of (12) is shown in Fig. 5. In effect, offset might exist in the error pattern. However, as long as over-ranging does not happen in the pipeline, it does not affect the overall linearity. Therefore, offset is excluded from our analysis. The two jumps β at the stage thresholds are one-stage code gaps (CGs). The branches of the error pattern in different decision regions can then be shifted to form a continuous nonlinear function $f(v)$, as shown in Fig. 6. $f(v)$ includes the effect of both linear errors and nonlinear errors. In the case that only linear error exists, $f(v)$ will be a straight line.

In order to compensate for the nonlinearity effectively, the error pattern need to be modeled. An easy way to model the nonlinear $f(v)$ is to formulate a look-up table (LUT). It gives superior accuracy; however, the expensive high-speed RAM for the large LUT renders this method impractical, although memory compression techniques can be used to reduce the memory usage.

Instead of using an LUT, we use linear segments to model the nonlinear function. Because the transfer function is usually symmetric about the origin, we will focus our discussion to one side. In Fig. 6, one internal breakpoint is used, which we define as a first-order modeling. With increased nonlinearity, more internal breakpoints can be used. Therefore, the task of the nonlinear calibration is to estimate the location of the breakpoints from the sampled data points during ADC operation. With positions of the internal breakpoints optimized, the raw code out of the pipeline can go through a compensation routine that generates codes in real time with good linearity.

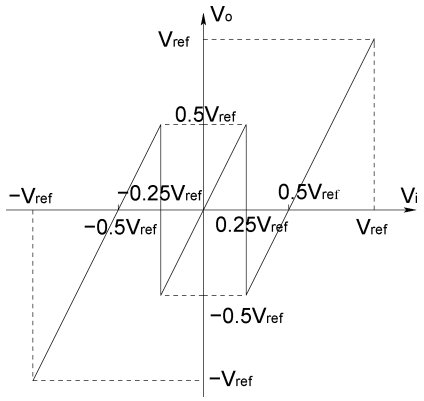


Fig. 4. Ideal transfer function of 1.5-bit stage.

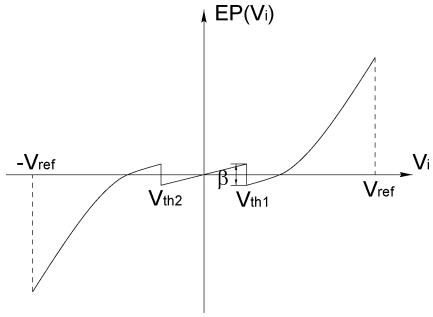


Fig. 5. Nonlinear error pattern.

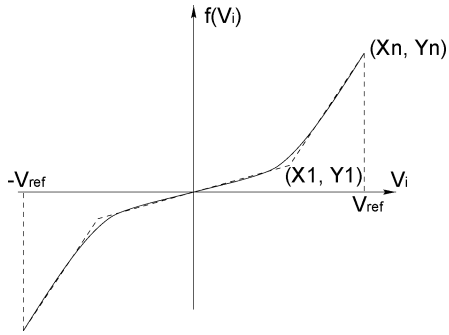


Fig. 6. Nonlinear error pattern with CGs removed.

B. Optimization Process Formulation

In the case of first-order modeling as shown in Fig. 6, there are two breakpoints and four parameters (β, x_1, y_1, y_n) that need to be determined. It is difficult, if not impossible, to determine all four parameters by direct and analytical methods. The most practical way is to use optimization methods.

The optimization process starts with an estimated parameter vector, $\hat{\mathrm{PV}} = (\hat{\beta}, \hat{x}_1, \hat{y}_1, \hat{y}_n)$. With the estimated stage error pattern $\hat{f}()$ available, we can estimate the input $\mathrm{voltage}(V_i)$ of stage i from its residual output $\mathrm{signal}(V_{i+1})$ and the generated digital $\mathrm{code}(c_i)$. Using the definitions in (13) and (14), we can derive the following equations:

$$V_{i+1} = \mathrm{STF}(V_i) = \mathrm{ITF}(V_i - \mathrm{EP}(V_i)) \tag{15}$$

$$V_i = \frac{1}{2}V_{i+1} + \frac{1}{2}V_{di} + \mathrm{EP}(V_i). \tag{16}$$

Therefore

$$\begin{aligned}\hat{V}_i &= \frac{1}{2}\hat{V}_{i+1} + \frac{1}{2}V_{di} + (\hat{f}(\hat{\hat{V}}_i) - \hat{\beta}(c_i - 1)) \\ &= \frac{1}{2}\hat{V}_{i+1} + \left(\frac{1}{2}V_{\mathrm{ref}} - \hat{\beta}\right)(c_i - 1) + \hat{f}(\hat{\hat{V}}_i)\end{aligned} \tag{17}$$

where

$$\hat{\hat{V}}_i = \frac{1}{2}\hat{V}_{i+1} + \frac{1}{2}V_{di} = \frac{1}{2}\hat{V}_{i+1} + \frac{1}{2}V_{\mathrm{ref}}(c_i - 1) \tag{18}$$

and c_i takes the value in {0,1,2}. Each stage in the pipeline can have its own error pattern, which is described by $(\beta_i, f_i())$. Using this set of recursive equations, the overall input voltage V_1 to the N-stage pipeline can be estimated as

$$\begin{aligned}\hat{V}_1 =& \frac{1}{2^N}\hat{V}_{N+1} \\ &+ \sum_{i=1}^{N}\frac{1}{2^{i-1}}\left(\frac{1}{2}V_{\mathrm{ref}} - \hat{\beta}_i\right)(c_i - 1) + \sum_{i=1}^{N}\frac{1}{2^{i-1}}\hat{f}_i(\hat{\hat{V}}_i).\end{aligned} \tag{19}$$

With several extra stages in the pipeline, the first term in (19) can be neglected. Also, (19) shows the well-known cascading effect that code gaps and nonlinearity in the front stages have a more significant effect on the performance of the ADC than the ones in the rear. Therefore, only the front N_c stages need to be compensated for these nonidealities. As a result, the final estimation of the input voltage is given as

$$\hat{V}_1 = \sum_{i=1}^{N}\frac{1}{2^i}V_{\mathrm{ref}}(c_i - 1) + \sum_{i=1}^{N_c}\frac{1}{2^{i-1}}(\hat{f}_i(\hat{\hat{V}}_i) - \hat{\beta}_i(c_i - 1)) \tag{20}$$

where

$$\hat{\hat{V}}_i = \frac{1}{2}\hat{V}_{i+1} + \frac{1}{2}V_{\mathrm{ref}}(c_i - 1). \tag{21}$$

In order to perform the calibration in the digital domain, we need to derive a digital version of the estimation process. Similar to (17) and (18), the digital recursive stage functions are given as

$$\hat{D}_i = \hat{D}_{i+1} + 2^{N-i}c_i + (\hat{f}_d(\hat{\hat{D}}_i) - \hat{\beta}_d(c_i - 1)) \tag{22}$$

$$\hat{\hat{D}}_i = \hat{D}_{i+1} + 2^{N-i}c_i. \tag{23}$$

The estimation of the final digital input code is shown as

$$\hat{D}_1 = \sum_{i=1}^{N}2^{N-i}c_i + \sum_{i=1}^{N_c}\frac{\hat{f}_{di}(\hat{\hat{D}}_i) - \hat{\beta}_{di}(c_i - 1)}{2^{i-1}} \tag{24}$$

where

$$\hat{\hat{D}}_i = 2^{i-1}(\hat{D}_{i+1} + 2^{N-i}c_i). \tag{25}$$

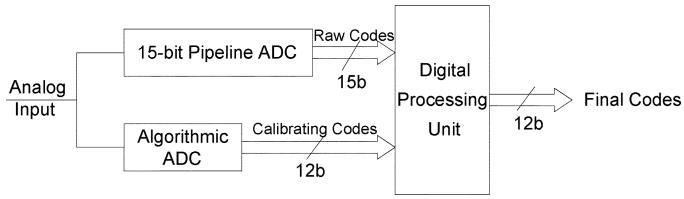


Fig. 7. Calibration architecture of the ADC.

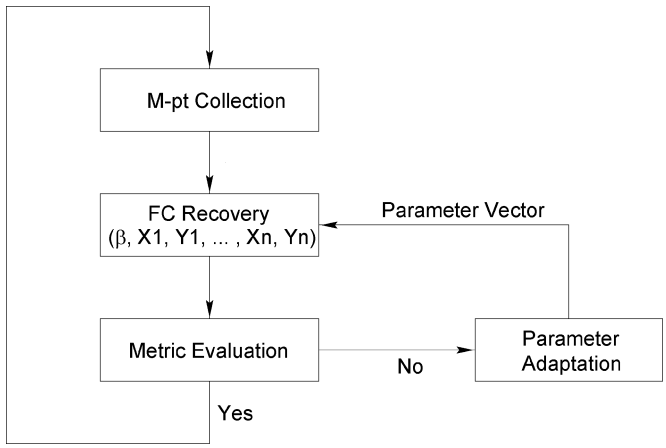


Fig. 8. State machine of the calibration process.

C. Calibration Process

We use a similar calibration architecture as the one used in [18], and the architecture is given in Fig. 7. The main pipeline ADC includes 14 1.5-bit stages, which generates 15-bit raw codes (RCs) that need to be calibrated. The algorithmic ADC generates 12-bit accurate calibration codes (CCs). The algorithmic ADC runs in the background, at a much slower rate than the main pipeline (1/32 of the main sampling rate in [18]). Hence, it is able to provide 12-bit resolution with low power consumption and small die area [18]. The digital processing unit (DPU) recovers the final codes (FCs) and estimates the transfer function.

The calibration process is shown in Fig. 8. The DPU first formulates an M point testing set out of the two ADCs, $\mathrm{TS} =$

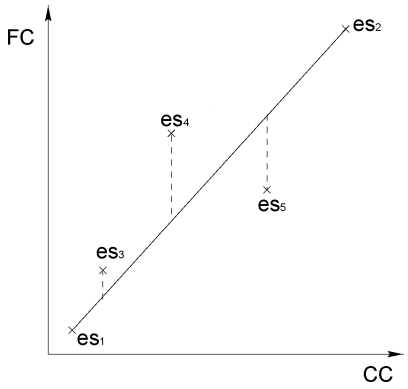


Fig. 9. Linearity evaluation method showing the FCs versus CCs.

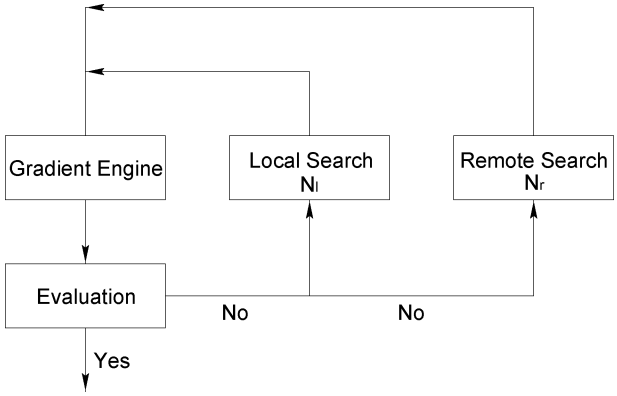


Fig. 10. Flow diagram illustrating the parameter adaptation procedure.

$\{\mathrm{ts}_i = (\mathrm{CC}_i, \mathrm{RC}_i) | i \in [1, \mathrm{M}]\}$. Then, the RCs go through a recovery operation, employing the estimation formula of (24) and (25), to generate the FCs, and formulate an evaluation set, $\mathrm{ES} = \{\mathrm{es}_i = (\mathrm{CC}_i, \mathrm{FC}_i) | i \in [1, \mathrm{M}]\}$. ES is evaluated by a linearity metric. If the evaluation result lies outside the allowed boundary, the parameters in the recovery unit adapts according to a certain adaptation rule. The FCs are recovered again and reevaluated until the evaluation result lies within the boundary. At this point, the calibration process starts another TS collection.

1) Linearity Evaluation: In order to simplify the hardware complexity of the linearity evaluation module, we use the method illustrated in Fig. 9 for linearity check. The first two points in ES, not necessarily the end points as shown in Fig. 9, are used to draw a line between the two points. Then, the positive FC distances of the other points in ES to the line are accumulated as

$$\mathrm{Metric} = \sum_{i=1}^{M} \left| \mathrm{FC}_i - \frac{\mathrm{FC}_{\max} - \mathrm{FC}_{\min}}{\mathrm{CC}_{\max} - \mathrm{CC}_{\min}} (\mathrm{CC}_i - \mathrm{CC}_{\min}) \right|. \tag{26}$$

The metric is similar to the mean-square metric, rather than arithmetic averaging. The total metric is compared with boundaries to determine the linearity.

The method in Fig. 9 is actually an M-pt INL accumulation. The recovered FC has 15 b. If the FCs have a strict linearity of 12 b, and the dynamic range of the FC is not much smaller than that of the CC, each node in an ES should result in a distance less than 4 LSB. Therefore, the total metric would be less than the boundary

$$\mathrm{BD}_1 = 4\mathrm{M}. \tag{27}$$

On the other hand, even if an ES is found to be within BD_1, it is not certain that the overall linearity of the recovered FCs reaches 12 b. However, there are two ways to reduce this positive false probability. An obvious method is to increase M. However, the hardware cost and the length of calibration cycle increases at least linearly with M. Another method is to continuously evaluate the linearity of recovered codes. If the evaluations pass linearity check constantly, we will call the calibration process converging. Although the recovered FCs may have codes out of the 0.5 LSB DNL and INL rules boundary, our simulation and measurement show that the frequency spectrum of a converged pipeline provides good performance in terms of overall SNR, SFDR, and THD.

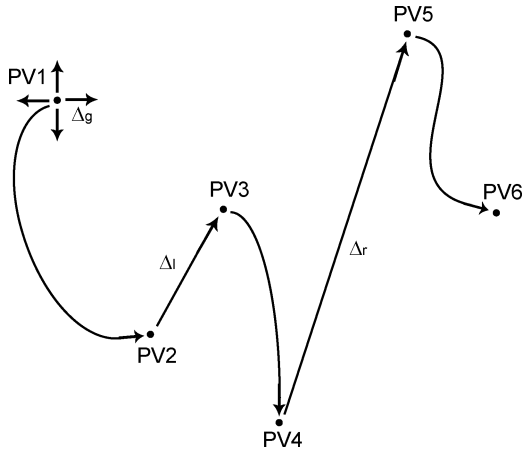


Fig. 11. PV status transfer: an example for parameter adaptation.

2) Parameter Adaptation: The logical procedure of parameter adaptation is shown in Fig. 10. An example of the status transfer for the parameter vector (PV) is shown in Fig. 11. The core of the module is a gradient engine, which adapts PV in the gradient direction. Within the gradient engine, elements of PV increase or decrease by a small fixed amount (Δ_g) each time. The adaptation is only validated when the evaluated metric improves. This process continues until a local minimum is reached, which is PV2 in Fig. 11. If the gradient adaptation stops with the metric still outside the boundary, the module searches the neighboring parameter space for a better PV. During a local search, PV changes by a random amount (Δ_l) to PV3. Then, PV is adapted by the gradient engine to another local minimum PV4. The random number can be generated by a pseudo random number generator. Although it requires a multiplier, which can be shared with other operation in the DPU, the random number generator does not increase the hardware cost. If PV4 is less than PV2, the local jump is beneficial, so PV4 is preserved. Otherwise, PV retracts to PV2 for another local jump. After N_l runs of failed local searches, the module jumps further (Δ_r) searching for a better PV solution. The previous procedure repeats until a local minimum PV6 is resulted. Again, the minimum is preserved. In the example of Fig. 11, PV4 is the final

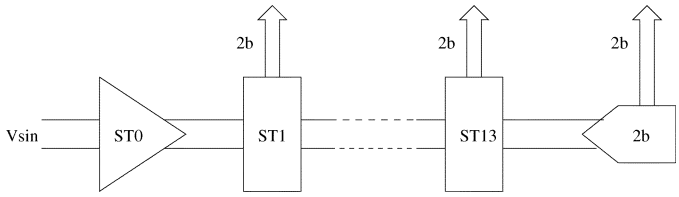


Fig. 12. Structure of the designed pipeline ADC.

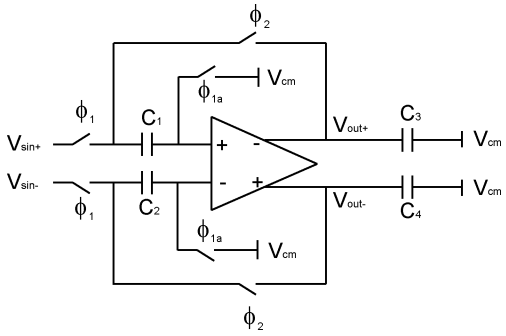


Fig. 13. S/H front-end.

adaptation solution. Practically, the choice of Δ_l and Δ_r will affect the convergence speed and the possibility of trapping into local minimum. In our experiment, the parameters are empirically chosen. Δ_l is limited within $\pm 2\Delta_g$, and Δ_r is limited within $\pm 20\Delta_g$. The convergence results are shown in the testing section.

The proposed adaptation procedure is a combination of the LMS-typed gradient method and simulated annealing [23]. Although, theoretically, it is difficult to reach the global minimum, the procedure is able to settle to a quasi-optimum in a short time, with a constrained parameter space and appropriate N_l and N_r. However, the efficiency of the adaptation greatly depends on the size of the PV, because the size and complexity of the parameter space increase exponentially with the PV size. The increased complexity prevents us from pursuing higher order modeling for practical reasons.

IV. Circuit Description

The designed fully differential ADC includes an S/H front-end, 13 1.5-bit stages, and a back-end to resolve the final two bits, as shown in Fig. 12. In order to save power, pipeline ADCs usually employ stage scaling [12]–[14]. This may cause different error patterns for the different types of stages. As the complexity of the adaptation problem of Section III-C2 grows exponentially with the size of the PV, the effectiveness of the calibration procedure has to be tested against the weighted pipeline. Therefore, we design three types of stages for the pipeline.

A. S/H Front-End

The fully differential S/H front-end is designed to achieve low random noise level and high linearity. It uses bottom-plate sampling, as shown in Fig. 13. A high linearity of the S/H front-end is critical for the overall performance of the ADC, because the calibration procedure is only useful against errors occurring in the pipeline.

TABLE I
Stages in the Designed Pipeline

	ST0	ST1-2	ST3-4	ST5-13
Sampling Cap. (pF)	4	0.4	0.2	0.1
Steering Current (mA)	4	4	2	1
kT/C Noise (μV)	32	102	144	204

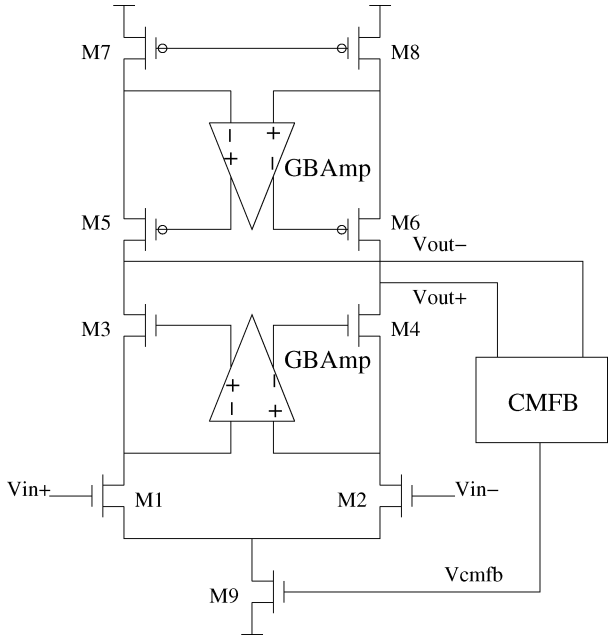


Fig. 14. Telescopic gain-boosted opamp structure.

In order to suppress the third-order harmonic, bootstrapping is used for the switches in the S/H front-end. To reduce the kT/C noise [25], the sampling capacitors in the S/H front-end is set to be 4 pF. Employing GBOPCAD as detailed in [22], we are able to achieve SNDR and SFDR over 80 dB for the S/H front-end. Although the performance of the S/H front-end cannot be measured directly, measurement of the overall performance proves that the S/H front-end is able to provide over 12-bit linearity.

B. 1.5-bit MDAC

The random noise control of the 1.5-bit stages in Fig. 2 is critical for the pipeline, especially for the front stages. Therefore, the sampling capacitors should not be too small. On the other hand, smaller sampling capacitors allow faster settling time for stages, which consume less power. After optimization and simulation, the sampling capacitors of the first stage is set to be 400 fF. At this capacitance, the kT/C noise level is still sufficiently low. The scalings of the stages are listed in Table I. NMOS switches are used for the MDAC, with differential comparators [5].

Telescopic gain-boosted opamps, as shown in Fig. 14, are used in our stages for less power and better design control by the computer-aided design (CAD) tool GBOPCAD. In order to introduce nonlinear error into the pipeline, the opamps are designed at the signal swing specification of 1 V in GBOPCAD, while the input signal swing is 2 V. Simulation shows that the dc gain of the amplifier is reduced to 18 dB at both ends of the

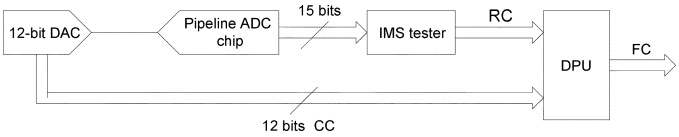


Fig. 15. Calibration setup.

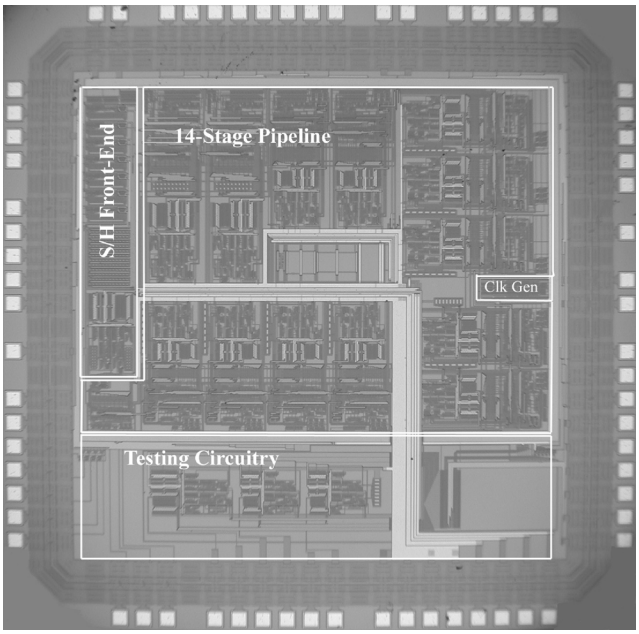


Fig. 16. Chip die photograph.

input range, while the normal dc gain of the amplifier is 86 dB. Hence, a considerable amount of gain dispersion exists within the signal range.

V. Implementation and Experimental Results

A. Experimental Setup

An algorithmic ADC could have been used as described in [18] but was omitted for simplicity. To prove the concepts behind this work, instead, a 12-bit DAC from Tektronics AWG420 is used to substitute the effect of a calibrated 12-bit algorithmic ADC. The actual prototype calibration setup for this work is shown in Fig. 15.

A 15-bit pipeline ADC is designed as described in Section IV and fabricated in a 0.6-μm CMOS process. The die photograph of the fabricated chip is shown in Fig. 16. The active die area is 3 mm^2. The supply voltage for the chip is 5 V. The voltage range of the analog input differential signal is within ± 2 V. The power consumption of analog circuits is measured to be 350 mW. The on-chip clock buffers are designed oversized to reduce clock skews at high clock rates, which consume a large part of the power. At the sampling rate of 33 MSample/s, the total power consumption of the chip is 650 mW.

In order to reduce the design complexity, the DPU is not included on the chip. Instead, the raw codes out of the pipeline are captured by an MSTS test station from integrated measurement systems (IMS). A software DPU is designed to calibrate the captured raw codes for the prototype. A device-under-test (DUT) board is designed to level-shift the common mode of the DAC differential output signal and to interface with the MSTS station.

Although the algorithmic ADC and DPU are not included on the prototype chip, our calibration method is expected to give considerable savings in die area for the complete system because large-capacity high-speed memories are not required. In [19], a 64-kb on-chip ROM is used, while a 256 k $\times$ 18-bit off-chip SRAM is used in [20].

B. Calibration Processor Implementation

The actual structure of the calibration DPU is shown in Fig. 17. The major calibration system includes the blocks at the downstream of the M-point TS buffer. It runs in the background. Any abnormality in the operation of the pipeline ADC can be picked up by the system, as the tested metric deviates beyond the boundaries. The adapter will be activated to adapt the parameter vector of the recovery module until the metric settles into the boundaries again. A pipelined recovery module is needed in the path of the FC to enable true background calibration in real time. This recovery module only takes settled PVs from the PV adapter. Besides the fundamentals introduced in Section III, several implementation issues, which affect the performance of the calibration system, are worth mentioning.

1) Dynamic Boundary Setting: The convergence rate of the calibration process inversely depends on the metric boundary. A larger boundary setting leads to faster convergence, but results in worse linearity. Although BD1 in (27) is an ideal boundary in terms of resulting linearity, the convergence can be rather slow for pipelines with complex nonlinear error patterns, which cannot suitably be modeled at the current order. Besides, pipeline with excessive noise is not able to converge within the small boundary as well. Theoretically, it takes long enough iterations to tell whether the boundary is too small for the current pipeline and the estimation model, for which case we call it singular. In order to run the calibration process within reasonable time, the boundary is dynamically adjusted in our procedure.

The setting starts with BD1. After the expiration of Counter2 in Fig. 17, which records unsuccessful iterations, the boundary increases. The iteration process continues with the larger boundary. As the result of the dynamic boundary setting, the calibration procedure always converges. If the final boundary is BD1, the calibrated ADC can achieve the specified linearity. If the final boundary is larger than BD1, the linearity of the calibrated ADC is expected to be worse than the specified linearity. To determine whether the current calibration setup is singular, the process can be repeated with a larger Counter2.

2) Modeling Order: Theoretically, high-order models can provide better modeling accuracy, which leads to better calibration and higher linearity. However, the multiple-stage modeling, due to the existence of multiple stage types in the pipeline, already expands the optimization space drastically. For first-order modeling, twelve parameters [($\hat{\beta}$, $\hat{x}_1$, $\hat{y}_1$, and $\hat{y}_n$) for each type of stage] need to be optimized in our optimization problem. Further raising the modeling order will exponentially expand this already large optimization space. Hence, the convergence

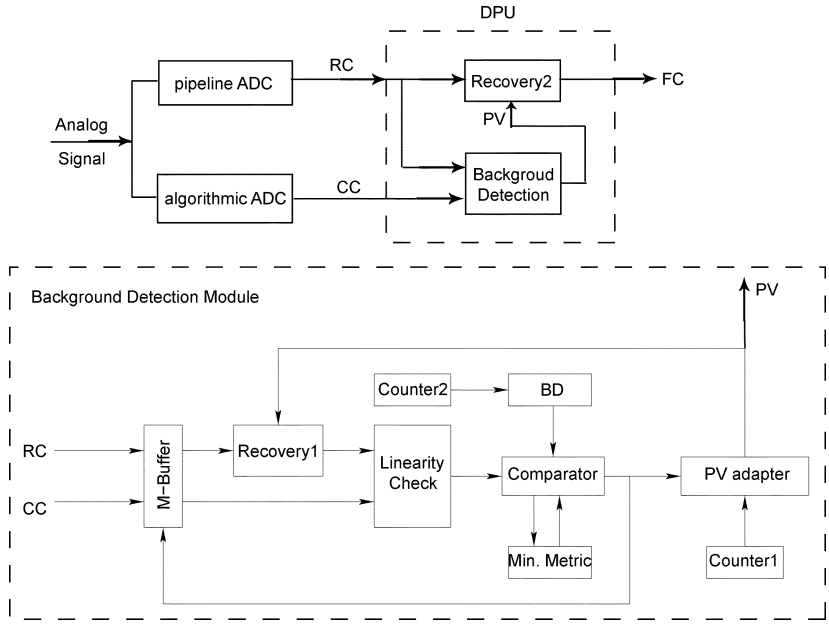


Fig. 17. Block diagram of the calibration DPU.

speed of high-order modeling will be significantly slower than that of the first-order modeling. On the other hand, in practical implementations, the large optimization space is much more sparsely sampled in high-order modeling than in the first-order modeling situation, which makes it more difficult for the optimization process to settle to the global optimum. As a result, the final linearity obtained with higher order models is not significantly improved. Accordingly, only first-order modeling is used here. Nonetheless, our experimental measurements show that first-order modeling converges in 9 ms for 12-bit linearity on our designed pipeline.

3) Hardware Cost: In the hardware implementation of DPU, 13-bit signed integers are used for the y-direction of the error pattern in Fig. 6, and 15-bit unsigned integers are used for the x-direction. A floating point representation is used for the slope, with 12-bit mantissa and 5-bit exponent.

Through simulations, modeling of the first six stages is found to be able to provide 12-bit calibration resolution. A longer modeled pipeline is not able to dramatically improve the resolution, while it linearly increases the DPU size. Therefore, in our implementation, only the first six stages are modeled. The rest of the stages are treated as ideal. Hence, inside the recovery module, there is a seven-stage pipeline as can be seen from (24) when $N_c = 6$. Each pipeline stage implements (22), which requires a 15×12 unsigned multiplier to calculate $\hat{f}_{di}()$. Therefore, a total of six 15×12 unsigned multipliers are required for this recovery module. This recovery module consumes most of the calibration hardware.

The operations in the detection module are carried out in series. Multipliers are shared among serial modules. A maximum of two 15×12 unsigned multipliers is required by the detection module, which is the width of the linearity check module.

Therefore, a total of eight 15×12 unsigned multipliers are required for the DPU for background calibration. In the case of foreground calibration, only six 15×12 multipliers are required. The 15×12 multiplier has been synthesized by Synopsis tools. It requires about 1400 gates. For a 0.25-μm CMOS process, the multiplier consumes about 4.6 mW power at the speed of 33 MHz. Hence, a DPU would require around 20 k gates, and the power consumption is less than 66 mW in a 0.25-μm CMOS process at 33 MHz.

TABLE II
CONVERGENCE TEST OF THE CALIBRATION PROCEDURE

	M = 20	M = 40	M = 80
Convergence on first ES	539	1050	1000
Final convergence	6135	1590	2820

Compared with the previous background calibration schemes, the DPU does not require ROM. The main usage of RAM is the M-Buffer in Fig. 17, which uses around 1-kb RAM for our prototype ADC. For the remainder of the DPU, only a small amount of registers are required to store intermediate computation results.

C. Experimental Results

The convergence property of the calibration procedure is tested at a sampling rate of 50 MSample/s. The boundary of BD1 is set for the calibration. Different block sizes(M) are used. The numbers of linearity check performed before the procedure reaches convergence are tallied and listed in Table II. The first row records the iteration counts for the procedure upon convergence on a single evaluation set. The second row lists the iteration counts before the procedure reaches final convergence, which is achieved when it passes linearity check for any evaluation set during operation. One observation is that the first convergence takes up a large amount of the total iteration process, which shows a high correlation among the convergence of different evaluation sets. This indicates the possibility of a common structure for raw codes recovery. Also, as shown in

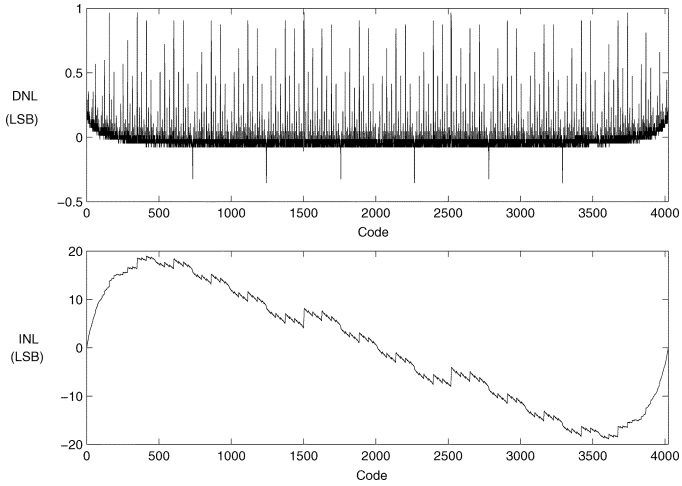


Fig. 18. Static performance without calibration at a sampling rate of 33 MSamples/s.

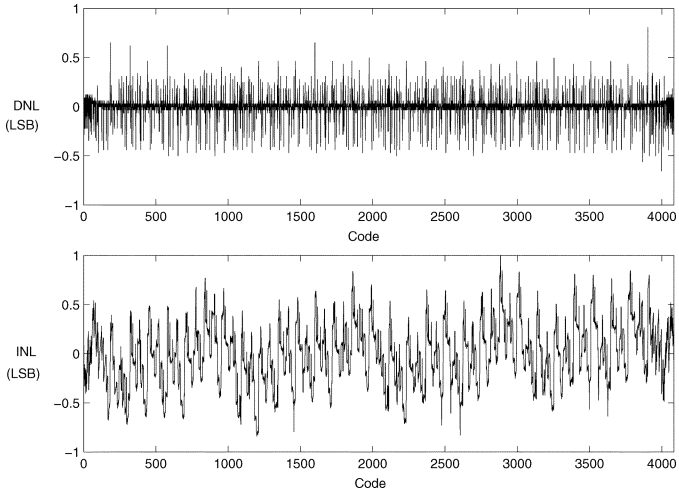


Fig. 19. Static performance with calibration at a sampling rate of 33 MSamples/s.

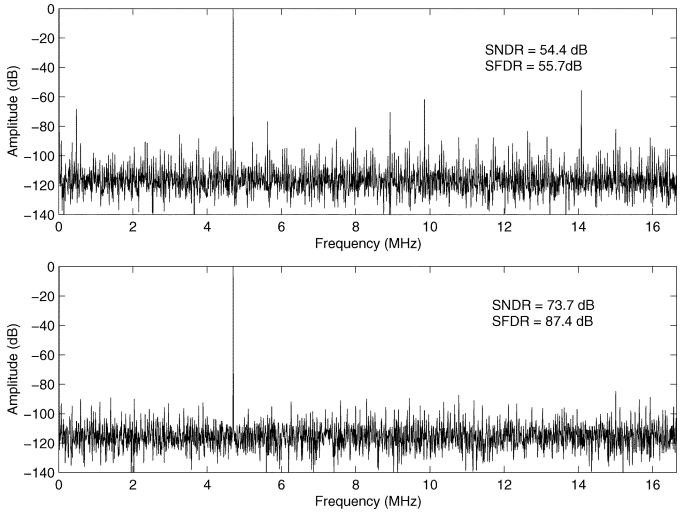


Fig. 20. 4 k-point FFT test results for ADC without and with calibration, $f_{\rm sig} = 4.7$ MHz.

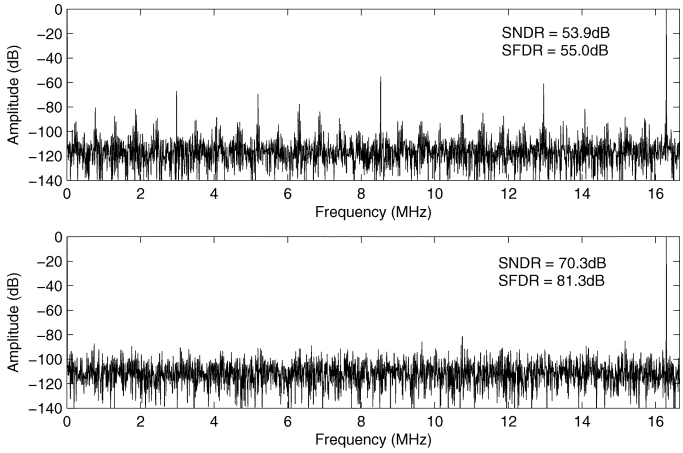


Fig. 21. 4 k-point FFT test results for ADC without and with calibration, $f_{\rm sig} = 16.3$ MHz.

Table II, the procedure converges faster on a single evaluation set at smaller block size. However, when the block size is too small, the convergence on one evaluation set correlates less with the other evaluation sets, which leads to a much higher overall iteration number. On the other hand, the convergence is slower when the block size is too large. Therefore, we use a block size of 40 for the other testings. Because each linearity check takes up about 7M cycles to complete, the convergence process lasts $1590 \times 7 \times 40 \times 20$ ns $= 8.9$ ms.

The ADC chip is tested for its static and dynamic performances. At the sampling rate of 33 MSample/s, the DNL and INL of the ADC chip is measured by the histogram method and plotted in Fig. 18. The INL indicates plenty of nonlinearity at both ends, with code gaps at major code transitions. After the background calibration procedure converges, the calibrated ADC is tested for the static performance again and is plotted in Fig. 19. Although the DNL and INL of some codes can still go beyond the 0.5-LSB boundary, the compensation over nonlinearity and code gaps is apparent. In order to test the dynamic performance of the ADC, the chip is tested for the 4 k-point fast Fourier transform (FFT) routine. At the sampling rate of 33 MSample/s, the chip is applied with a $4V_{pp}$ 4.7-MHz sinusoidal signal. The frequency spectra, with and without calibration, are measured and plotted in Fig. 20. The raw ADC codes are found to have high harmonic levels, especially a high third-order harmonic. The overall SNDR is 54.4 dB and SFDR is 55.7 dB. After calibration, the harmonics are significantly suppressed. The achieved SFDR is 87.4 dB, while the overall SNDR is 73.7 dB. The SNDR improvement is 19 dB, while the SFDR improves by 31 dB. The spectra for the 16.3-MHz input signal is shown in Fig. 21. The high-frequency input signal enhances the noise of the pipeline, which undermines the effect of the tcscalibration.

The calibration procedure is also tested at different sampling frequencies. The measured SNDR and SFDR under each sampling frequency are plotted in Fig. 22. At the sampling rate of 50 MSample/s, with the input signal of 7 MHz, the calibrated ADC achieves an SNDR of 70.2 dB and an SFDR of 82 dB. For sampling rates under 50 MSample/s, all calibrated ADCs achieve SNDRs beyond 71 dB and SFDRs beyond 82 dB. For

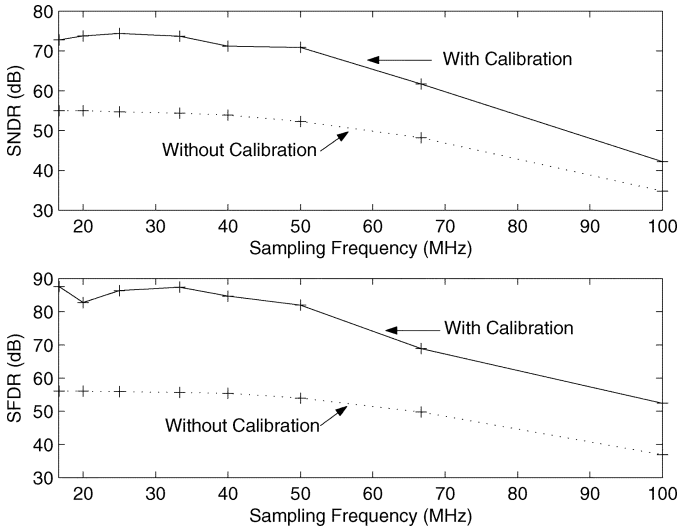


Fig. 22. SNDR and SFDR over different sampling frequencies.

sampling rates above 50 MSample/s, the frequency spectrum deteriorates quickly. At 66.7 MSample/s, the calibration procedure converges at a boundary of 168, which equals 4.2M. The resulting SNDR and SFDR are 61.7 and 68.9 dB, respectively.

Testing results show that the proposed technique can effectively compensate for the distortion in the pipeline. However, the noise in the fabricated pipeline limits the final resolution of the calibrated ADC. To run the proposed calibration technique effectively for 12-bit resolution, the noise level of the prototype ADC was designed to be less than 1 LSB of the 15-bit pipeline. In order to calibrate for 14-bit resolution, the noise level should be lower than 1 LSB of a 17-bit pipeline, which is challenging to achieve, particularly at high sampling frequency. Besides, a 14-bit algorithmic ADC is required.

VI. Conclusion

A nonlinear error model is studied for a pipeline ADC. Based on the analysis, a new piecewise linear calibration procedure is proposed. The procedure corrects not only the linear errors, but also the nonlinear errors. The digital processing unit, which performs the calibration procedure, requires about 20 k logic gates, without the need for a large quantity of ROM or RAM. The process runs in the background, without interrupting the normal operation cycles. The proposed calibration architecture potentially allows for designs of high-speed high-resolution pipeline ADCs at low power consumption.

A 0.6-μm CMOS prototype pipeline ADC chip which exhibited a considerable amount of nonlinear errors was used to evaluate the calibration procedure. Although the power consumption in this prototype was not optimized, it can be improved considerably by using a more advanced process and a lower supply voltage and further optimizing the design of clock buffers and more power-aware amplifiers. After calibration, the ADC achieves 12-bit resolution at 33 MSample/s. Another advantage is the fast convergence of the calibration procedure, of the order of milliseconds. The characteristics of the prototype chip and the calibration method are listed in Table III.

TABLE III
Performance Summary(Room Temperature)

Process	$0.6\mu m$ CMOS	
Supply Voltage (V)	5	
Sampling Rate (MS/s)	33	
Target Resolution (bit)	12	
Full Scale Range (V)	$4V_{pp}$	
Power (mW)	350(analog), 650(total)	
	Without Cal.	With Cal.
DNL (LSB)	± 1	± 0.8
INL (LSB)	± 20	± 1
SNDR (dB)	53.9	70.3 (f_{sig} = 16.3MHz)
SFDR (dB)	55.0	81.3 (f_{sig} = 16.3MHz)
ENOB (bit)	8.7	11.4

Acknowledgment

The authors would like to thank K. Nagaraj at Texas Instruments for valuable technical advice and review comments, as well as Q. Li at Texas Instruments for technical discussions. The authors would also like to thank W. Sansen at Katholieke Universiteit Leuven for his helpful comments. The authors are also grateful to R. Van Berg, M. Newcomer, G. Mayers, and M. Reilly of the High Energy Physics Group at the University of Pennsylvania for valuable discussions and advice during testing.

References

[1] S. Lewis and P. Gray, "A pipelined 5-Msample/s 9-bit analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. SC-22, no. 6, pp. 954–961, Dec. 1987.

[2] B. Song, M. Tompsett, and K. Lakshmikumar, "A 12-bit 1-Msample/s capacitor error-averaging pipelined A/D converter," *IEEE J. Solid-State Circuits*, vol. 23, no. 12, pp. 1324–1333, Dec. 1988.

[3] S. Sutarja and P. Gray, "A pipelined 13-bit, 250-ks/s, 5-V analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 23, no. 12, pp. 1316–1323, Dec. 1988.

[4] S. Lewis, S. Fetterman, G. Gross, R. Ramachandran, and T. R. Viswanathan, "A 10-bit 20-Msample/s analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 27, no. 3, pp. 351–358, Mar. 1992.

[5] A. Abo and P. R. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 599–606, May 1999.

[6] Z. Tao and M. Keramat, "A 10-bit 100 MS/s 50 mW CMOS A/D converter," in *Proc. IEEE Midw. Symp. Circuits Syst.*, Aug. 2000, vol. 1, pp. 48–51.

[7] L. Singer, S. Ho, M. Timko, and D. Kelly, "A 12-bit 65 MSample/s CMOS ADC with 82 dB SFDR at 120 MHz," in *ISSCC Dig. Tech. Papers*, Feb. 2000, pp. 38–39.

[8] H. Pan, M. Segami, M. Choi, J. Cao, F. Hatori, and A. Abidi, "A 3.3 V, 12 b, 50 MSample/s A/D converter in 0.6 μm CMOS with over 80 dB SFDR," in *ISSCC Dig. Tech. Papers*, Feb. 2000, pp. 40–41.

[9] Y. Lin, B. Kim, and P. Gray, "A 13-bit 2.5-MHz self-calibrated pipelined A/D converter in 3-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 26, no. 4, pp. 628–636, Apr. 1991.

[10] S. Lee and B. Song, "Digital-domain calibration of multistep analog-to-digital converters," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1679–1688, Dec. 1992.

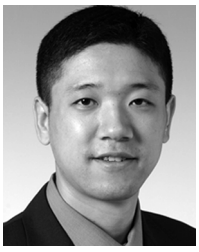
[11] A. Karanicolas, H. Lee, and K. Bacrania, "A 15-bit 1-Msample/s digitally self-calibrated pipeline ADC," *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1207–1215, Dec. 1993.

[12] S. Chuang and T. Sculley, "A digitally self-calibrating 14-bit 10-MHz CMOS pipelined A/D converter," *IEEE J. Solid-State Circuits*, vol. 37, no. 6, pp. 674–683, Jun. 2002.

[13] T. Cho and P. Gray, "A 10-bit 20 Msample/s, 35 mW pipeline A/D converter," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 166–172, Mar. 1995.

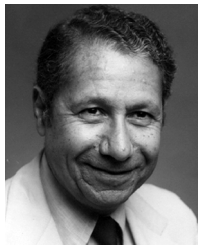
[14] D. Cline and P. Gray, "A power optimized 13-bit 5 Msample/s pipelined analog-to-digital converter in 1.2 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 31, no. 3, pp. 294–303, Mar. 1996.

[15] U. Moon and B. Song, "Background digital calibration techniques for pipelined ADC's," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 2, pp. 102–109, Feb. 1997.
[16] M. Taherzadeh-Sani and A. A. Hamoui, "A digital background calibration technique for capacitor mismatch errors in pipelined ADCs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 9, pp. 966–970, Sep. 2006.
[17] O. Erdogan, P. Hurst, and S. Lewis, "A 12-bit digital-background-calibrated algorithmic ADC with -90-dB THD," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1812–1820, Dec. 1999.
[18] X. Wang, P. Hurst, and S. Lewis, "A 12-bit 20-Msample/s pipelined analog-to-digital converter with nested digital background calibration," *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 1799–1808, Nov. 2004.
[19] B. Murmann and B. Boser, "A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2040–2050, Dec. 2003.
[20] C. Grace, P. Hurst, and S. Lewis, "A 12-bit 80 MSample/s pipelined ADC with bootstrapped digital calibration," *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1038–1046, May 2005.
[21] J. Yuan, N. Farhat, and J. Van der Spiegel, "A 50 MS/s 12-bit CMOS pipeline A/D converter with nonlinear background calibration," in *Proc. CICC*, Sep. 2005, pp. 399–402.
[22] J. Yuan, N. Farhat, and J. Van der Spiegel, "GBOPCAD: A synthesis tool for high-performance gain-boosted opamp design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1535–1544, Aug. 2005.
[23] R. Rutenbar, "Simulated annealing algorithms: An overview," *IEEE Circuits Devices Mag.*, vol. 5, no. 1, pp. 19–26, Jan. 1989.
[24] H. Ohara, H. Ngo, M. Armstrong, C. Rahim, and P. Gray, "A CMOS programmable self-calibrating 13-bit eight-channel data acquisition peripheral," *IEEE J. Solid-State Circuits*, vol. SC-22, no. 12, pp. 930–938, Dec. 1987.
[25] D. Johns and K. Martin, *Analog Integrated Circuit Design*. New York: Wiley, 1997.



Jie Yuan (S'03–M'06) received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 2001 and 2006, respectively.

In August 2006, he joined the Electronic and Computer Engineering Department, Hong Kong University of Science and Technology (HKUST), Kowloon, where he is currently an Assistant Professor. His research interests include high-speed analog and mixed-signal IC design, analog synthesis CAD tools, and biomedical related information processing systems.



Nabil H. Farhat (S'58–M'63–SM'72–F'81–LF'99) received the B.Sc. degree from the Technion Haifa, Haifa, Israel, in 1957, the M.Sc. degree from the University of Tennessee, Knoxville, in 1959, and the Ph.D. degree from the University of Pennsylvania , Philadelphia, in 1963, all in electrical engineering.

In 1964, he joined the Faculty of the Moore School of Electrical Engineering, University of Pennsylvania, where he is now a Professor of Electrical and Systems Engineering and heads the Electro-Optics and Photonic Neuroengineering Laboratory. His current research interests are in collective nonlinear dynamical information processing, neural networks, photonic realization of neurocomputers, and Corticonics, where he is applying concepts and tools from nonlinear dynamics, bifurcation theory, information-driven self-organization, and chaos to the modeling and study of cortical information processing. His teaching includes courses in neurodynamics and neural networks, EM theory, electrooptics, electron and light optics, and holography on both graduate and undergraduate levels. His past research included microwave diversity imaging, holography, automated target recognition, optical information processing, and the study of the interaction of EM radiation with plasmas and solids in the context of millimeter-wave and laser output energy measurement with glow discharge plasmas. He has held the Ennis Chair in Electrical Engineering and served as a Distinguished Visiting Scientist at the Jet Propulsion Laboratory, Pasadena, CA.

Dr. Farhat is a Fellow of the Optical Society of America and a member of the Electromagnetics Academy, the American Institute of Physics, Sigma Xi, and Eta Kappa Nu. He was a recipient of the University of Pennsylvania Christian R. and Mary F. Lindback Foundation Award for Distinguished Teaching. He has served on the National Board of Directors of Eta Kappa Nu, as an RCA consultant, and as Editor of *Advances in Holography,* Associate Editor of *Acoustical Imaging and Holography*, action editor for both *Neural Networks* and *Neural Computation*, and advisory editor for *Optics Letters.*

Jan Van der Spiegel (S'73–M'79–SM'90–F'02) received the M.S. degree in electromechanical engineering and the Ph.D. degree in electrical engineering from the University of Leuven, Leuven, Belgium, in 1974 and 1979, respectively.

He is a Professor with the Electrical and Systems Engineering Department and the Director of the Center for Sensor Technologies at the University of Pennsylvania, Philadelphia. His primary research interests are in high-speed, low-power analog and mixed-mode VLSI design, biologically based sensors and sensory information processing systems, micro-sensor technology, and analog-to-digital converters. He is the author of over 160 journal and conference papers and holds four patents.

Dr. Van der Spiegel is a member of Phi Beta Delta and Tau Beta Pi. He was the recipient of the IEEE Third Millennium Medal, the UPS Foundation Distinguished Education Chair, the Bicentennial Class of 1940 Term Chair, the Christian and Mary Lindback Foundation, the S. Reid Warren Award for Distinguished Teaching, and the Presidential Young Investigator Award. He has served on several IEEE program committees (IEDM, ICCD, ISCAS and ISSCC) and is currently the technical program Vice-Chair of the International Solid-State Circuit Conference (ISSCC2006). He is an elected member of the IEEE Solid-State Circuits Society (SSCS) and is also the SSCS chapters Chairs coordinator and former Editor of *Sensors and Actuators A* for North and South America.