Pipelined ADC Digital Calibration by Using Polynomial Inverse Function

HamidReza Mafi*, Hossein Shamsi**, Ehsan Shami*** and Reza Mohammadi****
*Qazvin University, Qazvin, Iran, hamid.mafi@gamil.com
** Electrical & Computer Faculty of K. N. Toosi University of Technology, Tehran, Iran, shamsi@eetd.kntu.ac.ir
***Electrical & Computer Faculty of K. N. Toosi University of Technology, Tehran, Iran, ehsan.shami@ee.kntu.ac.ir
****Electrical & Computer Faculty of K. N. Toosi University of Technology, Tehran, Iran, R.Mohammadi@ee.kntu.ac.ir

Abstract: *In this paper, a novel and simple digital technique to correct pipelined ADC errors is presented. Linearity of the ADC is improved by simultaneously adjusting the weights of the stages and the coefficients of the polynomials inverse function, using presented scheme. Output codes of the stages are applied to digital unit for calibration. Errors include capacitors mismatch, finite and nonlinear gain of Operational Amplifier (OPAMP), offset of OPAMP, and offset of sampling switches. The used algorithm is digital, completely. Trade-off between the speed and the accuracy of the pipelined ADC is improved by this technique, extensively. In the proposed technique, design problems of analogue circuits are moved inside digital circuit. The technique is simulated on a prototype 10bit 1.5bit/stage pipelined ADC with nonlinear OPAMP by using MATLAB/ Simulink. After the proposed calibration, the ADC achieves 64.5 dB SNDR and 73dB SFDR which are restricted to 51.2 dB and 57dB respectively before calibration.*

Keywords: pipelined ADC, digital calibration, polynomials inverse function.

1. Introduction

Demand for high speed data conversion in multimedia applications is increased for mobile and fixed communications [1]. Standard IEEE 802.16 which is used in cell phones and mobile networks requires ADCs with the sampling rate of 15 MHz to 250 MHz and the resolution of 10 to 16 bits [2, 3]. Pipelined ADCs are popular in high resolution and medium speed applications. Therefore, research span of pipelined ADCs are concentrated in the techniques which increase the speed and reduce the power consumption [4].

Unfortunately by enhancement of resolution, power dissipation of pipelined ADCs increases due to the high gain requirements of OPAMPs [2]. Also in sub_100nm technologies, power dissipation increases due to power supply reduction. This causes OPAMP gain decreases remarkably which influence design of high resolution and low power ADCs [5]. New researches on enhancement of battery life in mobile communications have led to the design of OPAMP with low gain and low accuracy [4] or replacement of OPAMPs with low power circuits [6]. Moreover, there are two concerns about OPAMP design in sub_100nm technology: gain and slew rate. This leads to use digital calibration technique which relaxes design complexity of analog circuits [3].

In the proposed technique to calibrate the output codes, errors of all stages are eliminated simultaneously by adaptive algorithm. In this technique, analog circuits which exist in the path of the signal are not manipulated. Therefore, the speed of the ADC which depends on the process is not reduced. By using this technique, memory-less errors such as capacitors mismatch, charge injection, finite gain, nonlinearity of the gain, and finite bandwidth of OPAMPs can be eliminated. Therefore, by the use of this technique we can reduce the power dissipation and increase the accuracy.

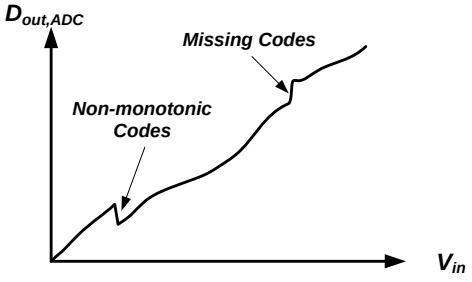

Fig. 1: Transfer function of a pipelined ADC

2. The Proposed Digital Calibration

In this section, calibration of output codes for pipelined ADC by using polynomial inverse function is described. The used technique is based on [3] and [6], but an improved version of them. A pipelined ADC samples the input voltage and converts it to N-bit code. The result code from an ideal ADC (D_{ADC}), which is the desired output code of the ADC, is the quantized version of the input voltage ($V_{in,ADC}$).

Though in the ideal ADC, the output binary codes depend on the input linearly. Sampling capacitors mismatches,

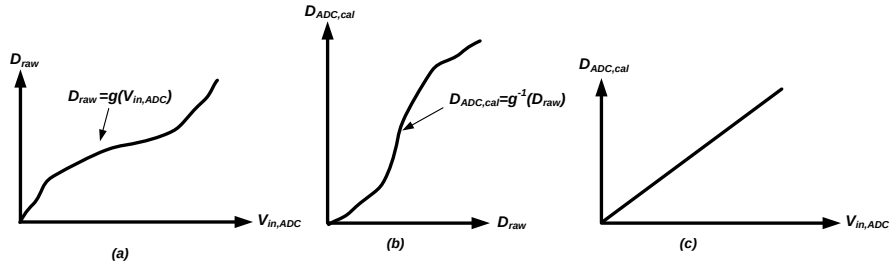

Fig. 2: (a) Transfer function of the calibrated ADC by using linear calibration technique, (b) Inverse function of (a), (c) the corrected output codes using inverse function technique.

finite and nonlinear gain cause distortion in ADC. Therefore, generated codes without calibration ($D_{out,ADC}$) in the ADC have sectional curves which cause nonmonotonicity, missing codes and hysteresis effect [6]. Fig. 1 shows the transfer function of a real pipelined ADC before error calibration.

Pipelined ADC which suffers from different errors can be modelled as $D_{out}=f(V_{in,ADC})$ which is shown in Fig. 1. The inverse function ($f(x)^{-1}$) in digital domain can eliminate all of errors which exist in the output of the ADC. But as shown if Fig. 1, the output codes are sectional which makes difficult implementation of such digital circuits [3].

When linear calibration algorithms (for example LMS algorithm) are used, linear calibrated output code (D_{raw}) is mainly one-to-one [6]. Therefore the output code of an uncalibrated ADC ($D_{out,ADC}$) which is shown in Fig. 1 is changed to the linear calibrated output code (D_{raw}) which is shown in Fig. 2(a). In the presence of nonlinear errors, the calibrated code by linear calibration techniques will have distortions. Fig. 2(a) shows the linear calibrated code (D_{raw}). D_{raw} is the nonlinear function of $D_{raw}=g(V_{in,ADC})$. The inverse function ($g^{-1}(D_{raw})$) can be used to achieve the ideal output codes (D_{ADC}) (see Fig. 2(b) ,(c)).

The proposed technique in this paper is approximating the inverse function by using polynomials.

$$D_{\mathrm{ADC,cal}} = g^{-1}(D_{\mathrm{raw}}) \approx \sum_{k=1}^{M} a_{\mathrm{k}} D^{k}{}_{\mathrm{raw}} \qquad (1)$$

Equation (1) shows Taylor expansion of the inverse function $g^{-1}(D_{raw})$ where $D_{ADC,cal}$ is the calibrated code by using the proposed technique. To implement the inverse function of polynomials, the factors of the polynomials inverse function are defined by the adaptive calibration techniques such as LMS technique.

To generate the linear calibrated code (D_{raw}), the decision code vector $\underline{D} = [D_1\ D_2\ \ldots\ D_N]^T$ is defined first. D_k is the digital output of the stage k which is shown in Fig. 3. In the pipelined ADC with the structure of 1.5bit/stage, the linear calibrated code (D_{raw}) is defined by the inner product of $\underline{D}$ and the weighing vector $\underline{W}=[w_1\ w_2\ \ldots\ w_N]^T$.

$$D_{\mathrm{raw}} = \underline{W^T D} = \sum_{k=1}^{N} w_{\mathrm{k}} D_{\mathrm{N}} \qquad (2)$$

D_{raw} is defined as linear function of the decision code vector $\underline{D}$. For an ideal ADC, the weight of the vector $\underline{W}$ will be $\underline{W}= [2^{N-1}\ 2^{N-2}\ \ldots 2^0]^T$. In this case the ADC doesn't have any distortion.

3. The Proposed Magnification Factor (S)

As it is given in equation (4), in the approximated polynomials inverse function, the calibrated code by the proposed technique ($D_{ADC,cal}$) is summations of the powered inputs. Every term in equation (4) lowers by the enhancement of the power in the polynomials because of $D_{raw}<1$. To use efficiently the polynomials inverse function and to have less bits in practical implementations, magnification factor (S) is used and $D_{ADC,cal}$ is defined as:

$$D_{\mathrm{ADC,cal}} = g^{-1}(D_{\mathrm{raw}}) \approx \sum_{k=1}^{M} a_{\mathrm{k}} S^{k} D^{k}{}_{\mathrm{raw}} \qquad (3)$$

If S is not used, the terms with the greater power than 3 will become so small. In the foreground digital calibration to relax the circuit implementations, the magnification factor (S) is approximated.

4. The ADC Structure

The output voltage of any stage is ideally equal to $V_{out}=2V_{in}+DV_{ref}$ where D is the output of the stage. The transfer function of the OPAMP is defined as $V_{out} = f(V_N)$. V_N can be defined as a function of the output voltage of the OPAMP ($V_N = f^{-1}(V_{out})$). In this case, the transfer function of the stage can be defined as:

$$\begin{aligned} C_1 V_{\mathrm{out}} &= V_{\mathrm{in}}(C_1 + C_2) \\ &\quad - (C_1 + C_2 + C_{\mathrm{p}}) f^{-1}(V_{\mathrm{out}}) - DV_{\mathrm{ref}} C_2 \end{aligned} \qquad (4)$$

Where V_{ref} is reference voltage, C_p is parasitic capacitance of the OPAMP input node, and D is the output of the stage which can be -1,0, or 1. Equation (4) can be written as:

$$\begin{aligned} V_{\mathrm{in}} &= V_{\mathrm{out}}\left(\frac{C_1}{C_1 + C_2}\right) \\ &\quad - \left(\frac{C_1 + C_2 + C_{\mathrm{p}}}{C_1 + C_2}\right) f^{-1}(V_{\mathrm{out}}) - DV_{\mathrm{ref}}\left(\frac{C_2}{C_1 + C_2}\right) \end{aligned} \qquad (5)$$

Equation (5) is defined as a function of the output. Dividing equation (5) by the voltage reference, the input code of the stage is defined as $D_{in}=V_{in}/V_{ref}$ and the output of the stage is defined as $D_{out}=V_{out}/V_{ref}$. Then equation (5) is changed to:

$$\begin{aligned} D_{\mathrm{in}} &= D_{\mathrm{out}}\left(\frac{C_1}{C_1 + C_2}\right) + \\ &\quad \left(\frac{C_1 + C_2 + C_{\mathrm{p}}}{C_1 + C_2}\right) f^{-1}(D_{\mathrm{out}}) + D\left(\frac{C_2}{C_1 + C_2}\right) \end{aligned} \qquad (6)$$

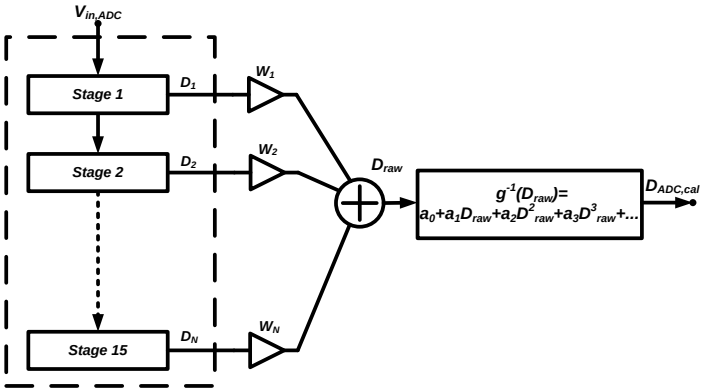

Fig. 3: proposed calibration technique.

Equation (6) is rewritten as:

$$\begin{aligned} D_{\text{in}} &= D_{\text{out}}\alpha + D\beta + \gamma . f^{-1}(D_{\text{out}}) \\ \alpha &= (\frac{C_1}{C_1 + C_2}) \\ \beta &= (\frac{C_2}{C_1 + C_2}) \\ \gamma &= (\frac{C_1 + C_2 + C_p}{C_1 + C_2}) \end{aligned} \tag{7}$$

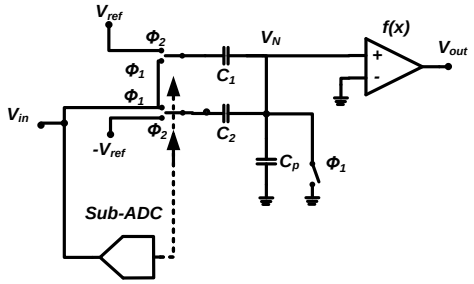

Fig. 4: scheme of 1.5 bit gain stage.

Equation (7) depends on the inverse function of D_{out}. Since the input of a stage is the output of the previous stage, we have $D_{in,k}=D_{out,k-1}$. Therefore, a recursive equation is achieved. By the expansion of equation (7), D_{raw} for the total ADC is defined as:

$$\begin{aligned} D_{raw} &= A + B + \Gamma \\ A &= \sum_{k=1}^{N} D_{\text{k}} u_{\text{k}} \\ B &= D_{\text{out,N}} \prod_{k=1}^{N} \alpha_{\text{k}} \\ \Gamma &= -\sum_{k=1}^{N} f^{-1}(D_{\text{out,k}}) p_{\text{k}} \end{aligned} \tag{8}$$

Equation (8) is formed of three parts (term): the first part (A) is the desired part, the second part (B) is the quantization noise, and the third part (Γ) is the nonlinearity of the ADC. The desired values for α_k and β_k are 1/2.

For an ideal ADC, p_k and u_k will be 2^{-k}. The inverse function in equation (8) can be defined by Taylor polynomials. It can be shown that by first four sentences of Taylor series, the desired accuracy is achieved:

$$\begin{aligned} f^{-1}{}_{\text{approx}}(D_{\text{out}}) = b_0 &+ b_1 D_{\text{out}} \\ &+ b_2 D^2{}_{\text{out}} + b_3 D^3{}_{\text{out}} \end{aligned} \tag{9}$$

In equation (9), by the power enhancement of polynomials, the b_i factors are decreased by the factor of ten [3]. In equation (8) by the enhancement of k, p_k is decreased by the order of two. It can be deduced that the nonlinearity effects of the first stages influence the nonlinearity of the total code more than last stages. To reduce the nonlinearity term (Γ) in the output code, the Taylor polynomials (Taylor polynomials factors can be achieved by the adaptive algorithms) can be used. To calculate the factors of W in equation (2) and a_i in equation (3), digital adaptive algorithm such as LMS algorithm can be used. These factors are adjusted by a recursive technique.

5. Adjusting the weights of stages and the polynomial coefficients

A Least-Mean-Square (LMS) algorithm is used for adjusting the weight of the stages, the polynomial coefficients and the proposed magnification factor (S). LMS Algorithm adjusts them by using an specified input values ($V_{in,ADC}$) and desired output codes (D_{ADC}). LMS is an iterative algorithm which needs to be repeated until the weights become adjusted [7]. The weights and coefficients are extracted as follows:

$$\begin{aligned} a_k[n+1] &= a_k[n] + \eta e S^k D^k{}_{raw} \\ w_k[n+1] &= w_k[n] + \eta e D_k \sum_{i=1}^{N} i a_i S^i D^{i-1}{}_{raw} \\ S[n+1] &= S[n] + \eta e D_k \sum_{i=1}^{N} i a_i S^{i-1} D^i{}_{raw} \\ e &= D_{ADC} - D_{ADC,cal} \end{aligned} \tag{10}$$

In equations (10), e is the difference between the desired output codes and the calibrated output codes by polynomial inverse function.

η is the update step-size in *LMS* which have to be between $0< \eta <1$.

5. Simulation Results

The proposed technique was simulated using MATLAB/Simulink. In these simulations, to achieve the 10-bit accuracy, a 14-stage inaccurate pipelined ADC is

used. A 14-bit ideal pipelined ADC is necessary to calibrate the inaccurate ADC. The gain of the OPAMPs in the stages is ten (A=10). The nonlinearity factor is defined as follows:

$$V_{out} = f(V_{in}) = \left(\frac{A(V_{in})}{1 + A(V_{in})/2}\right)V_{in} \qquad (10)$$

$$A(V_{in}) = A(1 - 0.25V_{in}^{\ 2})$$

The offset for the stages OPAMP is assumed 25mV. For the inverse function of polynomials, the order is chosen 3. The uncalibrated ADC suffers from a large number of missing codes and an INL of 50 LSB.

By using the proposed technique, the ADC achieves 0.5 LSB INL, 64.5 dB SNDR and 73dB SFDR which are restricted to 2 LSB INL, 51.2 dB SNDR and SFDR 57dB respectively by the technique explained in [6]. Thanks to the proposed technique, the total accuracy of the ADC is improved about 2 bits in comparison with the technique explained in [6] and the inaccurate ADC is corrected up to 10-bit.

Table I shows simulation results by using both the proposed technique and the technique explained in [6]. Fig. 5 shows the simulated integral nonlinearity of the ADC before calibration. Fig. 6 shows the simulated integral nonlinearity of the ADC after calibration by the proposed technique and the technique used in [6]. Fig. 7 shows the output spectrum of the ADC after calibration by the used technique in [6]. Fig. 8 shows the output spectrum of the ADC after calibration using the proposed technique.

TABLE I
Comparison of the proposed technique and the technique explained in [6].

	SFDR	SNDR	INL
The Proposed technique	73 dB	64.5 dB	0.5 LSB
The technique explained in [6]	57 dB	51.2 dB	2 LSB

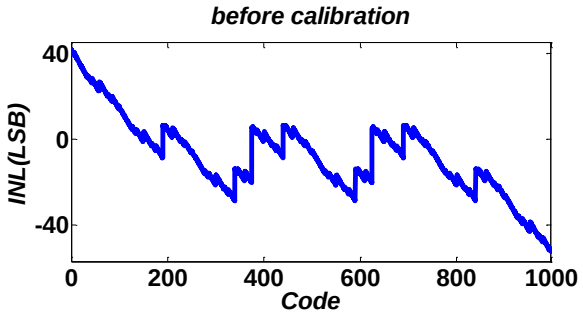

Fig. 5: Simulated INL before calibration.

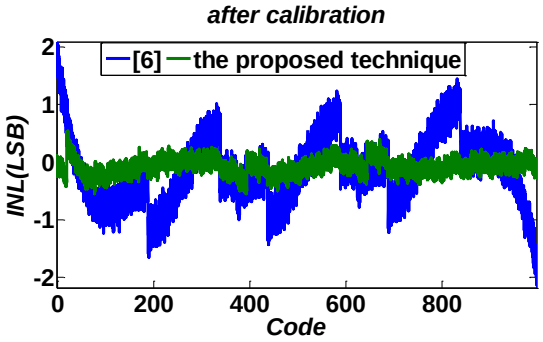

Fig. 6: Simulated differential and integral nonlinearity after calibration by the proposed technique and the technique used in [6].

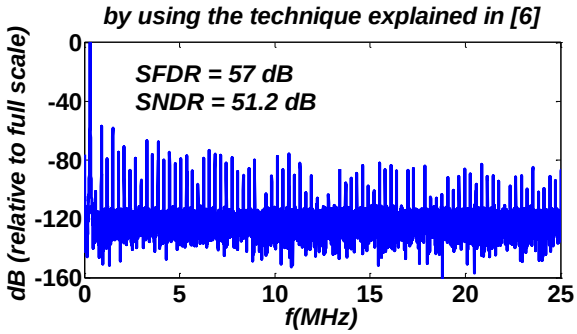

Fig. 7: Output spectrum after calibration by [6] for 485 KHz input frequency and 50-MHz sampling rate.

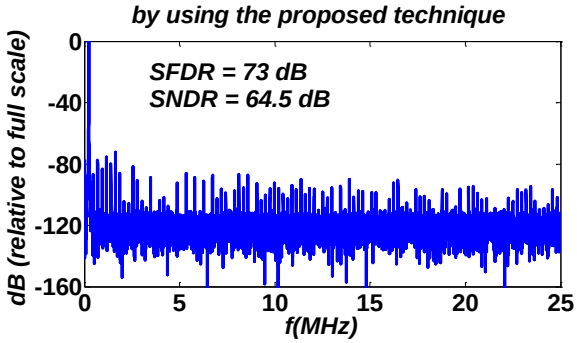

Fig. 8: Output spectrum after calibration by the proposed technique for 485 KHz input frequency and 50-MHz sampling rate.

6. Conclusions

In this paper a proposed digital calibration technique is presented to eliminate the linear errors of 1.5bit/stage pipelined ADCs produced by the finite and nonlinear gain in stages OPAMPs. The effectiveness of the proposed technique is demonstrated. The proposed technique improves nonlinearity errors of the stages, better than other techniques. In this technique, analog circuits which exist in the path of the signal are not manipulated.

References

[1] S. Jiang, M.A. Do , K. S. Yeo and W. M. Lim, "An 8-bit 200-MSample/s Pipelined ADC With Mixed-Mode Front-End S/H Circuit ," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 55, NO. 6, pp. 1430-1440, JULY 2008.

[2] A. Panigada, and I. Galton, "Digital Background Correction of Harmonic Distortion in Pipelined ADCs," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 53, NO. 9, pp. 1885-1895, SEPTEMBER 2006.

[3] A. Verma and B. Razavi, “A 10-Bit 500-MS/s 55-mW CMOS ADC,” 3048 IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 44, NO. 11, pp. 3039-3050, NOVEMBER 2009.Jk

[4] I, Ahmed, J. Mulder, and D. A. Johns, “A Low-Power Capacitive Charge Pump Based Pipelined ADC,” IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 45, NO. 5, pp. 1016-1027, MAY 2010.

[5] A. Annema, B. Nauta, R. Langevelde and H. Tuinhout, “Analog Circuits in Ultra-Deep-Submicron CMOS”, IEEE Journal of Solid-State Circuits, VOL. 40, NO. 1, JANUARY 2005.

[6] Y, Chiu, C. W. Tsang, B. Nikolic, and P. R. Gray, “ Least Mean Square Adaptive Digital Background Calibration of Pipelined Analog-to-Digital Converters,” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 51, NO. 1,pp.38-46 JANUARY 2004.

[7] S. S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ:Prentice-Hall, 1996.