A BACKGROUND DIGITAL SELF-CALIBRATION SCHEME FOR PIPELINED ADCS BASED ON TRANSFER CURVE ESTIMATION

Hanjun Jiang, Haibo Fei, Degang Chen and Randall Geiger

Department of Electrical and Computer Engineering
Iowa State University Ames, IA 50010 U.S.A

ABSTRACT

The pipelined architecture is one of the most popular ADC architecture. Various linear and nonlinear errors limit the pipelined ADC's performance. Many calibration algorithms to calibrate this architecture ADC have been reported in literature. In this paper, a new background self-calibration scheme for pipelined ADCs is presented and this calibration scheme can correct both linear and nonlinear errors in the pipelined data path. Simulation shows that with this calibration scheme, the ENOB of a 16-bit pipelined ADC can be improved from 10bits to about 15 bits.

1. INTRODUCTION

Analog-to-Digital Converters (ADCs) are identified as the world's largest volume mixed-signal circuits and are widely used in signal processing circuits and systems. While more and more stringent performance requirements such as high resolution and high speed are imposed on ADC design, the shrinking device feature size and decreasing supply voltage make ADC design more and more challenging.

Among the various ADC architectures, the pipelined architecture is one of the most popular architectures for achieving high resolution and modest high-speed conversion. In pipelined ADC design, there are many issues that limit the overall performance such as finite and nonlinear operational amplifier (op-amp) gain, capacitor mismatch and voltage coefficients, comparator offset error and so on. To overcome these limitations, in state-of-the-art pipelined ADC design, people usually consult to various calibration algorithms to improve the ADC performance [1]-[5].

The pipelined ADC overall performance is determined by the conversion and inter-stage transfer relationship of each stage. From the viewpoint of calibration, it is sufficient to study the pipeline inter-stage transfer curves. Among those errors, the gain error and offset error in the transfer curves are treated as linear errors and the nonlinearity in the transfer curves is treated as nonlinear error. Most reported background algorithms only handle the linear errors [1]-[3]. Nonlinear error with benign form is calibrated in [5]. Our goal is to correct both the linear error and the low frequency nonlinear error with all forms.

This paper presents a background digital self-calibration scheme for the pipelined ADCs based on the inter-stage transfer curve estimation. It is organized as the following. After this introduction, section 2 briefly reviews pipelined ADC modeling and inter-stage transfer errors. Section 3 presents the ADC architecture and operation for inter-stage transfer curve estimation. Then the linear and nonlinear calibration schemes will be discussed in section 4 followed by the simulation results given in section 5. Finally the work is summarized in section 6.

2. PIPELINED ADC MODELING

In this section, we will review the pipelined architecture briefly and analyze its possible errors. A pipelined ADC is composed of some pipeline stages. Each pipeline stage receives the analog signal from the previous stage (or the original input for the 1st stage), and outputs one or multiple digits and a residue voltage to the next stage. The most simple and widely used pipelined architecture is the one bit per stage architecture this architecture will be our example to propose this new calibration scheme due to its simplicity and popularity. A typical 1-bit pipeline structure is shown in figure 1.

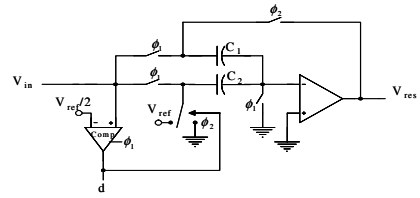


Figure 1: One bit/stage pipeline stage structure

For the structure shown in figure 1, the input range is [0, V_{ref}]. The comparator threshold voltage is $V_{th}=V_{ref}/2$. When the input voltage $V_{in}>V_{th}$, d=1; otherwise d=0. If the nonlinear errors such as op-amp nonlinearity and capacitor nonlinearity are not taken account in, the inter-stage transfer curve is linear and given by

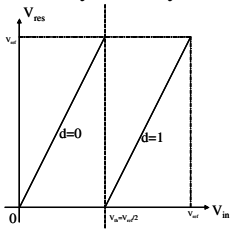
$$V_{res}=\frac{1}{1+\frac{C_2+C_{in}}{C_1}}\cdot\left[\left(1+\frac{C_2}{C_1}\right)\cdot V_{in}-\left(\frac{C_2}{C_1}\right)\cdot V_d(d)\right] \qquad (1)$$

where A_0 is the op-amp dc gain, C_{in} is the input capacitance at the op-amp input node, and $V_d(d)=d\cdot Vref$ is determined by the output code d. Ideally if A_0 is infinite and C_1 and C_2 are exactly matched, we have

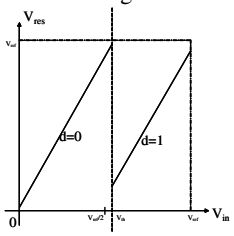
$$V_{res}=\left(1+\frac{C_2}{C_1}\right)\cdot V_{in}-\left(\frac{C_2}{C_1}\right)\cdot V_d(d)=2\cdot V_{in}-V_d(d) \qquad (2)$$

One ideal inter-stage transfer curve is depicted in figure 2(a). However, there are various errors that make the inter-stage transfer curve non-ideal. The comparator may have an offset error such that $V_{th}\neq V_{ref}/2$. In actual situation, A_0 is finite, and C_1 and C_2 may not be perfect matched. The op-amp may have input

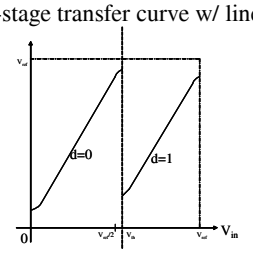
offset error. Besides, for over-range protection, the inter-stage gain is usually set to be less than 2 purposely. With above errors, the two pieces of inter-stage transfer curve are still straight but may deviate from the ideal position as depicted in figure 2 (b). Furthermore, there are also some nonlinear error sources such as the third order nonlinearity of the op-amp and the voltage coefficient of capacitance which cause the transfer curve bend a little bit as shown in figure 2(c). Both the linear and nonlinear errors should be removed by calibration. However, previous calibration algorithms usually can only handle the linear errors.



(a) Ideal inter-stage transfer curve



(b) Inter-stage transfer curve w/ linear errors



(c) Inter-stage transfer curve w/ linear and nonlinear errors

Figure 2: Pipeline stage inter-stage transfer curve

The pipelined ADC performance is totally determined by the inter-stage transfer curve of each stage. To calibrate the ADC, we can just inspect and correct the errors of the transfer curves without caring about the detail error sources. Therefore, we need to characterize the transfer curves for calibration. The calibration performance will depend on the inter-stage transfer curve estimation.

3. TRANSFER CURVE ESTIMATION

This section will discuss how to estimate the inter-stage transfer curve for pipeline stages. We will present a method to characterize transfer curves without a benign shape that is assumed in [5]. Generally speaking, to characterize the transfer curve we can send some stimulus to the pipeline stage and then measure the residue voltage. For a background calibration scheme, the signal path along the pipeline should not be interrupted during characterizing each stage. To capture the interaction of the stages in the signal path, the pipeline stages should not be taken out from the signal path for characterization. Our target is to develop a background calibration algorithm that calibrates the actual signal path. The block diagram of the pipelined architecture for a background calibration algorithm is shown in figure 3. Each stage is composed of a cross-point switch as depicted in figure 4 in addition to the standard pipeline. During normal operation, the cross-point switches direct the signal through successive stages. One CAL DAC and one cyclic ADC are used to accommodate the calibration. The CAL DAC provides the stimulus for the stage under characterization. Both the CAD DAC and cyclic ADC don't need to be accurate. Since the cyclic ADC will be used only once during hundreds of thousands conversion cycles for background calibration, the performance degradation of the whole ADC due to the cyclic ADC can be neglected. The hardware overhead for both the CAL DAC and the cyclic ADC can be minimal.

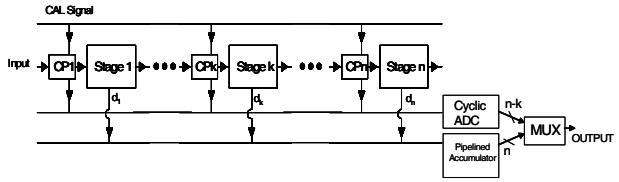


Figure 3 Background calibration pipelined architecture

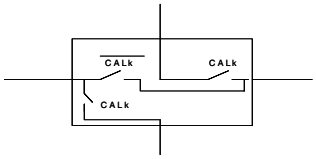


Figure 4 Cross-point switch of stage k

To characterize stage k, we may assume the tail stages following stage k are already calibrated. During characterization of stage k, the CAL signal generated by the CAL DAC is sent to the input of stage k with the cross-point switch and the output of stage k-1 is directed to the cyclic ADC. The cyclic ADC will determine the remaining n-k+1 bits of the signal in the next n-k+1 clock cycles. The same clock that is used to clock the pipelined architecture will be applied to the cyclic ADC. What interests us is the residue of stage k quantized by stage k+1~n with the CAL input. The same CAL signal is then presented to the input of stage k+1 and the output of stage k is directed to the cyclic ADC so that the CAL signal is also quantized by stage k+1~n. Using the input and residue values of stage k quantized by stage k+1~n, the inter-stage transfer curve of stage k can be estimated.

The CAL DAC outputs multiple voltage levels (for example, P=129 levels, P is always odd for convenience) to the ADC stages that cover the whole ADC input range. When the first (P+1)/2 input voltages are presented to stage k, the output code d of this stage is forced to be 0 and hence (P+1)/2 points ($D_{in\_l}(i)$, $D_{res\_l}(i)$) (i=1,…,(P+1)/2) on the left piece of transfer curve are collected. Similarly when last (P+1)/2 input voltages are presented to stage k, d is forced to be 1 and (P+1)/2 points ($D_{in\_r}(i)$, $D_{res\_r}(i)$) (i=1,..,(P+1)/2) on the right piece of transfer curve are collected. Note that the ($D_{in\_l}(P+1)/2$), $D_{res\_l}(P+1)/2$)) and ($D_{in\_r}(1)$, $D_{res\_r}(1)$) both correspond to the median of CAL DAC outputs which is around $V_{ref}/2$ for the single ended case.
We use these quantized values ($D_{in\_l}(i)$, $D_{res\_l}(i)$)'s and($D_{in\_r}(i)$, $D_{res\_r}(i)$)'s to estimate the inter-stage transfer curve by curve fitting. Various basis functions can be used to represent the curve. In this work, we adopt high ordered polynomial fitting. Since the tail stages after stage k are already calibrated, the quantized values by these stages contain only quantization errors. To estimate the inter-stage transfer curve precisely, the

least mean square method (LMS) is used to accomplish the curve fitting. Assume the fitted two pieces of transfer curves are represented by two t^{th} ordered polynomials $f_l(x)$ and $f_r(x)$.

$$f_l(x)=\sum_{i=0}^{t}a_{il}x^i \quad f_r(x)=\sum_{i=0}^{t}a_{ir}x^i \tag{3}$$

The coefficients a_{il} and a_{ir} (i=0,1,…,t) are found by minimizing the mean square error MSE

$$MSE=\sqrt{\sum_{j=1}^{(P+1)/2}\left(D_{res\_l}(j)-\sum_{i=0}^{t}a_{il}D_{in\_l}^{i}(j)\right)^2+\sum_{j=1}^{(P+1)/2}\left(D_{res\_r}(j)-\sum_{i=0}^{t}a_{ir}D_{in\_r}^{i}(j)\right)^2} \tag{4}$$

To cancel the effect of quantization errors during curve fitting, the number of points collected and thereby the number of CAL DAC output levels P should be adequately large. Besides, the order of polynomials should also be chosen properly. In this work, P is chosen to be 129, and the polynomial order is chosen to be 6. Since averaging technique [1] is not used, the number of measurements is reasonable. To do 6^{th} ordered polynomial fitting using on-chip DSP in large system, a memory space to store a 6 by 6 matrix is required. The computation cost is acceptable.

Once the fitted curves $f_l(x)$ and $f_r(x)$ are obtained for each stage with the aid of on-chip DSP, the calibration codes can then be generated based on these estimated inter-stage transfer curves.

4. LINEAR & NONLINEAR ERROR CALIBRATION

In this section, we will talk about how to generate the calibration codes based on the estimated inter-stage transfer curves.

For most applications, the ADC linearity performance is the most significant specification and the overall transfer gain is not critical, which means that is if the ADC decision intervals are mapped into digital numbers linearly, then it is a "good" ADC. To achieve the overall linearity, the basic idea is to make the transfer curve "straight" for each stage. The linear and nonlinear calibration codes will be generated respectively corresponding to the linear and nonlinear errors. The calibration is carried out starting from the tail stages. The linear and nonlinear errors are calibrated simultaneous for stage k. After stage k is calibrated, we can then calibrate the previous stage k-1 and so on.

a) Improved linear error calibration

For linear error correction, the two pieces of transfer curve of each stage are "connected" by interpreting output code "1" correctly for certain stage. The basic idea from [1]&[4] is as following. As shown in figure 5 (a), input V_x (equal or close to V_{th}) to stage k, force the output digit to be 0 and the residue quantized by following stages is D_1, and then force the digital output of stage k to be 1 and the quantized residue is D_2. Then for stage k, the interpretation of the output digit 1 is D_2-D_1. It must be pointed out that the comparator offset does not hurt since this offset error only makes one piece of transfer curve longer and the other piece shorter but the overall transfer curve of that stage after interpretation is still straight as long as the digit 1 is interpreted correctly.

However, such interpretation may have error up to 1 LSB. As shown in figure 5, due to the random offsets, the code width W_1 and W_2 are not controllable, and they may vary from 0 to 1 LSB. After the "connection", the code width around the major transition point is W_1+W_2, which may vary from 0 to 2 LSB. Such kind of error can be accumulated along the whole pipelined data path, and then the total error after linear error correction may be not satisfying.

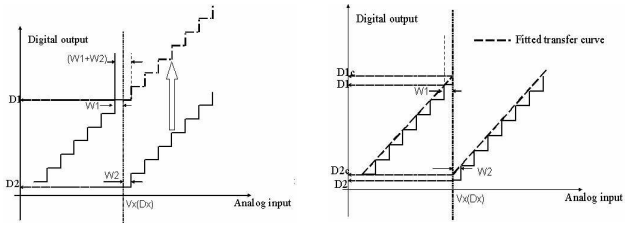
To solve this problem, we make use of the estimated transfer curve. D_1 and D_2 are obtained similarly as the previous method. Here V_x is the median of the P output samples from the CAL DAC. Quantize V_x to a digital number D_x using the tail stages. As shown in figure 5(b), calculate D_{1c} and D_{2c}

$$D_{1c}=f_l(D_x)\ \&\ D_{2c}=f_r(D_x) \tag{5}$$

Then W_1 and W_2 in LSB can be calculated as

$$W_1=D_{1c}-D_1\ \&\ W_2=D_2+1-D_{2c} \tag{6}$$

Now interpret digital 1 as D_1-D_2 if W_1+W_2<1.5LSB and D_1-D_2+1 if W_1+W_2≥1.5LSB. With this method, there will not be any code width larger than 1.5LSB after linear correction. Compared to the technique presented in [1], this proposed method can achieve satisfying linear error correction without truncation while somehow truncation means loss of number of bits.



(a) Commonly used method (b) Code width adjustment method

Figure 5: Linear error correction methods

b) Nonlinear error calibration

With nonlinear error present, the transfer curve is continuous after linear error correction but may be still not straight. Nonlinear error calibration for further performance improvement is required. The target of nonlinear error calibration is to calibrate the bowed transfer curve to a straight line. For the two pieces of transfer curve of stage k, the targeted straight curves can be the end point fit-lines. We already have the estimated curves $f_l(x)$ and $f_r(x)$. For the left piece transfer curve the targeted straight curve $g_l(x)$ is the connection of $(D_L,f_l(D_L))$ and $(D_x,f_l(D_x))$, and $g_r(x)$ by connecting $(D_x, f_r(D_x))$ and $(D_H,f_r(D_H))$ for the right piece. Here D_L and D_H are the quantization of the lowest input and highest input voltages.

Suppose that for one input to stage k, that stage outputs digit d and the following stages output raw code D. Then the calibration code for the combination of d and D is given by

$$C(d,D)=\begin{cases} g_l\left(f_l^{-1}(D)\right)-D, & \text{when } d=0 \\ g_r\left(f_r^{-1}(D)\right)-D, & \text{when } d=1 \end{cases} \tag{7}$$

It might be too complicated to calculate the reverse polynomial function $f_l^{-1}(D)$ & $f_r^{-1}(D)$. As an approximation, we can use the following formula instead of (7):

$$C(d,D)\approx\begin{cases} D-f_l\left(g_l^{-1}(D)\right), & \text{when } d=0 \\ D-f_r\left(g_r^{-1}(D)\right), & \text{when } d=1 \end{cases} \tag{8}$$

We can calculate C(d,D) for each raw output code and round it to the integer number $\hat{C}(d,D)$. Figure 7 shows an example of calibration codes for different D when d=0.

It is impractical to store calibration codes for every raw code. The raw codes need to be grouped and calibration codes is generated and stored for each group. We can group the raw codes into a bunch of intervals bounded by D_{s0}, D_{s1},D_{s2},…as

shown in figure 6. The interval starts from D_{si} share the same calibration code $\hat{C}_i$ (i=0,1,2,…). We need to store D_{si} and $\hat{C}_i$ (i=0,1,2,…) for d=0 and 1 respectively. Then when a raw code (d,D) is received, we compare it to D_{si} (i=0,1,2,…) to determine the calibration code. The nonlinear error amount is limited, so the calibration scheme requires only a small lookup table for each stage. If we assume the nonlinear errors don't exceed 7 LSB (3 bits), with 6th ordered polynomial fitting, in worst case 96 $\hat{C}$'s and D's need to be stored for each stage.

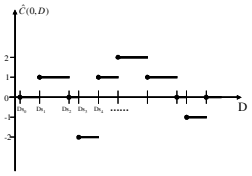
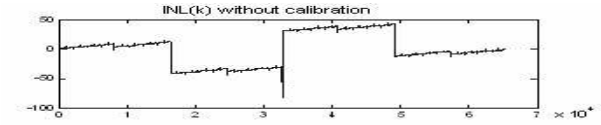


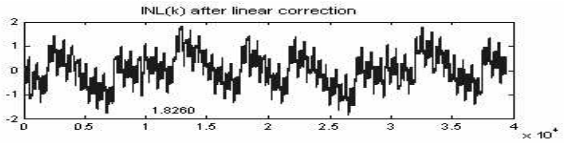
Figure 6: Possible calibration code distribution

5. SIMULATION RESULTS

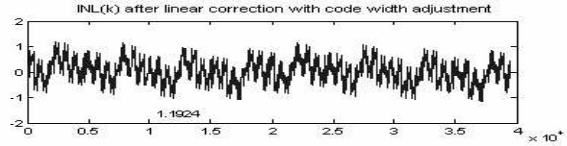
This section will present the simulation results for a 16 bit pipelined ADC. Modeling with MATLAB First we will show the performance improvement with the improved linear error calibration scheme. The simulated ADC has only linear errors for this simulation. Figure 7(a) shows that before linear error calibration, the INL can be up to 81.5 LSB. Figure 7(b) shows that when there is conventional linear error calibration, the INL is reduced to 1.83 LSB. Figure 7(c) shows after the improved linear error calibration, the INL can be further reduced to 1.19LSB. Note that all the INL's are calculated at 16-bit level without truncation. It is clearly shown that the improved linear error calibration scheme is effective.



(a) INL(k) with linear error and no calibration



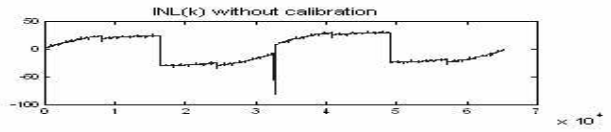
(b) INL(k) after conventional linear error calibration



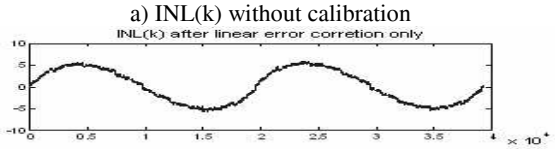
(c) INL(k) after improved linear error calibration

Figure 7: INL(k) before and after linear error correction

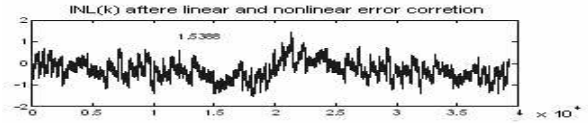
For actual ADCs, the linear error calibration is not sufficient. The effect of combining both linear and nonlinear error calibration is also simulated and the results are given in figure 8. In simulation, the 16bit pipeline ADC has both linear and nonlinear errors (the op-amp is not linear). The nonlinear error is set at about 3 bits level. Figure 8(a) is the INL(k) plot for the pipelined ADC without any calibration. The INL is about 82.5LSB. Figure 8(b) shows the INL(k) plot after linear error calibration only. The INL is reduced to 5.9 LSB with an effective number of bits (ENOB) of about 13 bits. Figure 8(c) shows that without truncation the INL of the pipelined ADC is reduced to 1.5LSB with the combination of linear error and nonlinear error calibration. The nonlinear error calibration can achieve an extra ENOB improvement of 2 bits comparing with the ADC with linear error calibration only. Since both the linear and nonlinear calibrations are done in digital domain, the final INL is larger than 1 LSB due to quantization effect.



a) INL(k) without calibration



b) INL(k) after linear error calibration



c) INL(k) after linear and nonlinear error calibration

Figure 8: INL(k) before &after linear+nonlinear error calibration

6. SUMMARY

We have proposed and simulated a background digital self-calibration scheme for pipelined ADCs in this paper. The algorithm calibrates the linear errors with an improved approach compared to previous schemes. The algorithm also compensates for the nonlinear errors in the data path. Simulation results show that with this calibration algorithm, a pipelined ADC with an original ENOB of 10bits can be improved to have an ENOB of about 15 bits. The calibration scheme is reliable and practical for hardware implementation. Further work will integrate this scheme on chip in a pipelined ADC design.

7. REFERENCES

[1] A. N. Karanicolas, H.-S. Lee, and K. L. Bacrania, "A 15-b 1-Msample/s digitally self-calibrated pipeline ADC," IEEE J. Solid-State Circuits, vol. 28, pp. 1207–1215, December 1993.

[2] J. M. Ingino, B. A. Wooley, "A Continuously Calibrated 12-b 10MS/s 3.3-V A/D Converter," IEEE J. Solid-State Circuits, vol. 33, pp. 1920–1931, December 1998.

[3] D. Fu, K. C. Dyer, S. H. Lewis, and P. J. Hurst, "A Digital Background Calibration Technique for Time-Interleaved Analog-to-Digital Converters," IEEE J. Solid-State Circuits, vol. 33, pp. 1904–1911, June 1998.

[4] S.-Y. Chuang, and T. L. Sculley, "A Digitally Self-calibrating 14-bit 10-MHz CMOS Pipelined A/D Converter," IEEE J. Solid-State Circuits, vol. 37, pp. 674–683, June 2002

[5] B. Murmann and B.E. Boser, "A 12b 75MS/s pipelined ADC using open-loop residue amplification", Dig. of Technical Papers, ISSCC 2003, vol. 1, pp. 328-497, February 2003