A BLIND IDENTIFICATION APPROACH TO DIGITAL CALIBRATION OF ANALOG-TO-DIGITAL CONVERTERS FOR BUILT-IN-SELF-TEST

Le Jin, Kumar L Parthasarathy, Degang Chen and Randall Geiger

Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011, U.S.A.
Email: djchen, rlgeiger@iastate.edu

ABSTRACT

A major bottleneck in Analog and Mixed-Signal Built-In-Self-Test (AMBIST) is the difficulty in generating high precision input stimuli that are required in most existing AMBIST schemes. This paper presents a new approach to AMBIST of ADCs using low precision input stimuli. With mild qualitative assumptions on input signals and the model of the ADC, the blind identification algorithms identify both the non-precision part in input signals as well as correction codes for the ADC from the ADC output codes. Initial simulation results show that a 12-bit ADC can be calibrated to achieve INL at the +0.5/-0.5 LSB level from an uncalibrated 25 LSB INL level with input stimuli have only 7-8 bit accuracy.

1. INTRODUCTION

Built-In-Self-Test (BIST) for analog and mixed-signal circuits has been an area of active research in the recent past. The widespread application of mixed-signal circuits in the communications and signal processing arena and the rapid shrinking of feature sizes have resulted in the emergence of large markets for low cost mixed-signal ICs with varied functionality. This has indirectly placed an upper bound on test solutions and the associated testing cost [1]. The cost effectiveness of existing methodologies of using commercial testers, which not only require an up front investment in high-cost testers, but also recurring costs associated with production time testing, has created an opportunity for a new approach for designing mixed-signal circuits with integrated Built-In Self-Test solutions.

One of the main obstructions to realizing a complete built-in solution is the unavailability of sufficient information about input test signals. This is similar to a typical problem of blind approximation often faced in many fields of engineering, in which an unknown system driven by an inaccessible signal must be identified solely based on the system's output. Even though the identification of a system with little or no information about the system seems to be an impossible task, elegant solutions could be achieved if some a priori knowledge is available about the input signal or the system. With the vast amount of information available about data converter operation and the various standard methods of on-chip signal generation, certain practical assumptions can be made about the system and the signal that can make it possible to identify the system without having direct access to the excitation source. In this work, one algorithm for testing and characterizing of data converters is introduced. The hardware implementation and complexity of the algorithm are not considered at this stage since the focus is to identify a means of system identification and provide a proof-of-concept.

2. ISSUES REGARDING PRESENT APPROACHES

Most existing Analog and Mixed-Signal BIST (AM-BIST) solutions, focuses on generating very precise input test signals on-chip that can be used to excite the device-under-test followed by a simple analysis of the resulting output signals to complete the device characterization [2]. In particular, efforts have been made towards generation of precise input signals like linear ramps, sinusoids etc., and towards suitable methods for identifying device properties from the resultant output. Invariably, the testing algorithms have been devised with an assumption that the on-chip generated input stimuli are significantly more accurate than the device under test. As we move towards high-resolution products, the challenges of generating these precise input signals is becoming more challenging than the design of the device under test itself. In the context of AMBIST, the problem of device characterization without complete knowledge of the input signal has received minimal attention by the research community and this has become a major drawback in the industrial adoption of many proposed AMBIST solutions.

A completely different strategy aimed at mathematically modeling the device and the associated error mechanisms due to the device and the input signal has been adopted in

This work was supported, in part, by the National Science Foundation, the Semiconductor Research Corporation, and the R. J. Carver trust.

this work. Such modeling helps us in identifying and exploiting certain characteristics of the system that could result in a simple and accurate test solution. The proposed method is based on the availability of a low accuracy input signal that can be practically and repeatedly generated on-chip, thereby eliminating the need for precision-stimulus generation.

3. ALGORITHM DESCRIPTION

If $D_{out}\,(D_0, D_1, \ldots D_N)$ is the digital output word of an A/D converter and X and V_{Ref} are the analog input and reference signals respectively, then the signals are ideally related by the following equation:

$$Y_{ideal} = V_{Ref}(D_1 2^{-1} + \ldots + D_N 2^{-N}) = X + Q_n \quad (1)$$

where Y_{ideal} is the analog value corresponding to D_{out} and Q_n represents the quantization noise and is limited to ± 0.5 LSB [3]. The output of a non-ideal ADC however differs from Y_{ideal} by an amount $\epsilon(Y)$ such that

$$Y = Y_{ideal} + \epsilon(Y) = X + \epsilon(Y) + Q_n \quad (2)$$

where $\epsilon(Y)$ is the error code of Y. It is reasonable to assume that $\epsilon(Y)$ is a function of Y and every digital output has one ideal digital correction code associated with it. All error codes can be precisely identified and stored for calibration if the characteristics of the input signal are known exactly. Unfortunately in most applications, little or no knowledge is available about the non-idealities of the input. This complicates the task of system identification. For instance, if $X(t)$ refers to the input signal at any instant of time, then it can be represented as sum of two components,

$$X(t) = s(t) + f(t) \quad (3)$$

where $s(t)$ is the intended input and $f(t)$ is some unknown function that results in a non-ideal input. In the following two algorithms are proposed that can be used to estimate the error correction code in the presence of one or more input signals as described by (3).

3.1. Algorithm 1

The following algorithm describes identification of the system with just one input signal. The performance improvement obtained by this method along with the limitations and alternate solutions are also described.

In what follows it will be assumed that the full-scale input is 1 (if a voltage ADC, input range is 1V). Consider the desired input stimuli to be a ramp signal. The actual (non-ideal) input can be defined as

$$X_{actual}(t) = t + \sum_{i=0}^{\infty} a_i f_i(t) \quad \text{for } 0 \le t \le 1 \quad (4)$$

where the first term on the right-hand side is the ideal ramp and the second term models the non-linearity of the input in terms of a set of known basis functions with coefficients that are not known precisely. In the following analysis, only the first two terms of the infinite series are included to simplify the analysis. The implications of this assumption will be described later. The input can then be written as

$$X(t) = t + a_1 f_1(t) + a_2 f_2(t) \quad \text{for } 0 \le t \le 1 \quad (5)$$

If the ADC is sampled at n time points t_i that are uniformly distributed between 0 and 1, then from (2) and (5) we get,

$$\begin{aligned} Y_i &= X_i + \epsilon(Y_i) + Q_{n,i} \\ &= t_i + a_1 f_{1,i} + a_2 f_{2,i} + \epsilon(Y_i) + Q_{n,i} \\ &\quad \text{for } i = 1, 2 \ldots n \end{aligned} \quad (6)$$

The subscript i denotes that the input is sampled at t_i. It will be assumed that the sampling is sufficiently fast so that a large number of sample of the input correspond to each digital output code. Because of the over sampling, several different sets of $\{t_i, f_{1,i}, f_{2,i}\}$ correspond to a same output code Y_j and hence an average value of the set is used as follows:

$$\begin{aligned} Y_j = &\operatorname*{avg}_{Y_i = Y_j} \{t_i\} + a_1 \operatorname*{avg}_{Y_i = Y_j} \{f_{1,i}\} + a_2 \operatorname*{avg}_{Y_i = Y_j} \{f_{2,i}\} \\ &+ \epsilon(Y_j) + \operatorname*{avg}_{Y_i = Y_j} \{Q_{n,i}\} \quad \text{for } j = 0, 1, 2 \ldots 2^N \end{aligned} (7)$$

Since we assume that a unique $\epsilon(Y)$ exists for each Y, the average value of $\epsilon(Y)$ remains the same. In addition, if n is large, the average value of the quantization noise approaches zero and can be neglected. (7) can be written in the matrix form as :

$$\hat{Y} = \hat{t} + \hat{A}\hat{a} + \epsilon(\hat{Y}) \quad (8)$$

where,

$$\hat{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{2^N} \end{pmatrix}, \hat{t} = \begin{pmatrix} \overline{t_1} \\ \overline{t_2} \\ \vdots \\ \overline{t_{2^N}} \end{pmatrix}, \hat{A} = \begin{pmatrix} \overline{f_{1,1}} & \overline{f_{2,1}} \\ \overline{f_{1,2}} & \overline{f_{2,2}} \\ \vdots & \vdots \\ \overline{f_{1,2^N}} & \overline{f_{2,2^N}} \end{pmatrix},$$

$$\hat{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \epsilon(\hat{Y}) = \begin{pmatrix} \epsilon(Y_1) \\ \epsilon(Y_2) \\ \vdots \\ \epsilon(Y_{2^N}) \end{pmatrix}$$ wherein $\overline{t}$ and $\overline{f}$ are average values. Moving $\hat{t}$ to the left hand side and multiplying by $\hat{A}^T$we get

$$\hat{A}^T \left(\hat{Y} - \hat{t}\right) = \hat{A}^T \hat{A}\hat{a} + \hat{A}^T \epsilon(\hat{Y}) \quad (9)$$

If $\hat{A}^T$ is orthogonal to $\epsilon(\hat{Y})$, the second term on right side reduces to 0 and the value of vector a (referred as $a_{\hat{get}}$) can be obtained as

$$\hat{a_{get}} = \left(\hat{A}^T\hat{A}\right)^{-1}\hat{A}^T\left(\hat{Y} - \hat{t}\right) \qquad (10)$$

This value of $\hat{a_{get}}$ can then be substituted in (8) to obtain the error codes, $\hat{\epsilon_{get}(Y)}$ as follows

$$\hat{\epsilon_{get}(Y)} = \hat{Y} - \hat{t} - \hat{A}\hat{a_{get}} \qquad (11)$$

However, if $\hat{A}^T$ is not orthogonal to $\epsilon(Y)$, then the value of $\hat{a_{get}}$ differs from actual a by the value given below

$$\hat{a_{get}} = \hat{a} + \left(\hat{A}^T\hat{A}\right)^{-1}\hat{A}^T\epsilon(\hat{Y}) \qquad (12)$$

and $\hat{\epsilon_{get}(Y)}$is related to actual $\epsilon(\hat{Y})$ by equation (13).

$$\epsilon(\hat{Y}) - \hat{\epsilon_{get}(Y)} = \hat{A}\left(\hat{A}^T\hat{A}\right)^{-1}\hat{A}^T\epsilon(\hat{Y}) \qquad (13)$$

It can be seen that the difference depends on the real error codes and the choice of basis functions. It is also evident that the larger the number of terms in $\hat{A}$, the worse the result becomes, because more components in $\epsilon(\hat{Y})$ that are correlated to the basis functions are extracted from $\hat{\epsilon_{get}(Y)}$ into $\hat{a_{get}}$. On the other hand, a choice of small set of basis functions also leads to an erroneous result. This is because if not all the a terms are included in $\hat{a_{get}}$, they remain in $\hat{\epsilon_{get}(Y)}$ and corrupt the signal. Thus an optimum set of basis functions exists that gives the best result. But this optimum cannot be reached easily since $\epsilon(\hat{Y})$ is not known beforehand. Simulation results in Section 4 indicate the improvement that can be obtained using just one input and also shows the limitations of assuming $\hat{A}^T$ to be orthogonal to $\epsilon(\hat{Y})$. This problem can be eliminated by using the alternate algorithm described below.

3.2. Algorithm 2

In this approach, we use two independent signal sources as input to calculate the error codes accurately. The two input signals are defined as:

$$X_1(t) = s_1(t) + a_1f_1(t) + a_2f_2(t) \quad \text{for } 0 \leq t \leq 1 \quad (14)$$

$$X_2(t) = s_2(t) + a_3f_3(t) + a_4f_4(t) \quad \text{for } 0 \leq t \leq 1 \quad (15)$$

where $s_i(t), i = 1, 2$ are ideal expected signals, and $f_i(t), i = 1, 2, 3, 4$ are four independent basis functions used to depict the deviations of the real input signals. The task is to identify the coefficients of the basis functions, $a_i, i = 1, 2, 3, 4$ based on the output results obtained.

Similar to the method adopted in Algorithm 1, the output of the ADC for two different inputs can be written in matrix form as shown in (16) & (17).

$$\hat{Y_1} = \hat{s_1} + \hat{A_1}\hat{a_1} + \epsilon(\hat{Y_1}) \qquad (16)$$

$$\hat{Y_2} = \hat{s_2} + \hat{A_2}\hat{a_2} + \epsilon(\hat{Y_2}) \qquad (17)$$

where $\hat{s_i} = \begin{pmatrix} \overline{s_{i,1}} \\ \overline{s_{i,2}} \\ \vdots \\ \overline{s_{i,2^N}} \end{pmatrix}$, $\hat{A_1} = \begin{pmatrix} \overline{f_{1,1}} & \overline{f_{2,1}} \\ \overline{f_{1,2}} & \overline{f_{2,2}} \\ \vdots & \vdots \\ \overline{f_{1,2^N}} & \overline{f_{2,2^N}} \end{pmatrix}$,

$\hat{A_2} = \begin{pmatrix} \overline{f_{3,1}} & \overline{f_{4,1}} \\ \overline{f_{3,2}} & \overline{f_{4,2}} \\ \vdots & \vdots \\ \overline{f_{3,2^N}} & \overline{f_{4,2^N}} \end{pmatrix}$, $\hat{a_1} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ and $\hat{a_2} = \begin{pmatrix} a_3 \\ a_4 \end{pmatrix}$

The first subscript i of the elements of $\hat{s}$ indicates the input, the second subscript j indicates the output code and $'-'$ refers to the average value . The elements of $\hat{Y_1}$ and $\hat{Y_2}$ are arranged in the same order, such that $\hat{Y_1} = \hat{Y_2}$ and $\epsilon(\hat{Y_1}) = \epsilon(\hat{Y_2})$. Subtracting (17) from (16), we get

$$0 = (\hat{s_1} - \hat{s_2}) + [\hat{A_1} \quad -\hat{A_2}]\begin{pmatrix} \hat{a_1} \\ \hat{a_2} \end{pmatrix} \qquad (18)$$

The coefficients values are then given by:

$$\begin{pmatrix} a_{1,get} \\ a_{2,get} \\ a_{3,get} \\ a_{4,get} \end{pmatrix} = \left(\hat{A'}^T\hat{A'}\right)^{-1}\hat{A'}^T(\hat{s_2} - \hat{s_1}) \qquad (19)$$

where $\hat{A'} = [\hat{A_1} \quad -\hat{A_2}]$.

If the effect of quantization noise and incomplete set of basis functions are neglected, then the above obtained values of $\hat{a_{get}}$ are the exact values. Replacing these coefficients values in (16) & (17), we get two values for each error code which are ideally the same and are given by:

$$\begin{aligned} \hat{\epsilon_{get}(Y)} &= \left(\hat{Y} - \hat{s_1}\right) - \hat{A_1}\hat{a_{1,get}} \\ &= \left(\hat{Y} - \hat{s_2}\right) - \hat{A_2}\hat{a_{2,get}} \qquad (20) \end{aligned}$$

4. MODELING AND SIMULATION RESULTS

A Flash ADC was modeled in MATLAB to validate the performance of the algorithm. All the resistors in the model of an ideal Flash converter were chosen to be Ro. However, in reality, the resistance of each resistor will be different from the ideal value due to various systematic and random effects. To simulate the non-ideal converter a random value was added to each resistance based on a normal distribution. Each of the resistance can then be modeled as:

$$R_i = R_o(1 + \delta_i) \quad \text{for } i = 1, 2, \ldots 2^N \qquad (21)$$

where δ_i refers to the deviation of the real resistance from the ideal value and was randomly chosen between +/-0.5. Such a deviation of resistance results in INL values of several LSBs due to the random-walk pattern observed in flash architectures.

Algorithm 1 was first used to characterize a 12-bit flash ADC and the effect of the number of samples in each bin and the coefficients of the basis functions on the output were observed. $s(t)$ was chosen as a ramp signal and the two basis functions were:

$$f_1(t) = t^2 - t \text{ and } f_2(t) = t^3 - 1.5t^2 + 0.5t \qquad (22)$$

The values of a_1 and a_2 were selected as 0.08 and 0.1 respectively, corresponding to an input signal of 6 bit accuracy. The number of samples in each bin was approximately 10. Figure 1 gives a plot of the maximum and minimum INL before and after correction for 50 runs. A different set of random resistance was used in each run to model different INL patterns. As seen from the figure, a improvement of 1-2 bit in INL is obtained by using just one input. Simulations with different number of samples per bin and with different choice of $a1$ and $a2$ indicate that the results are not dependent on them. To see the effect of multiple inputs,

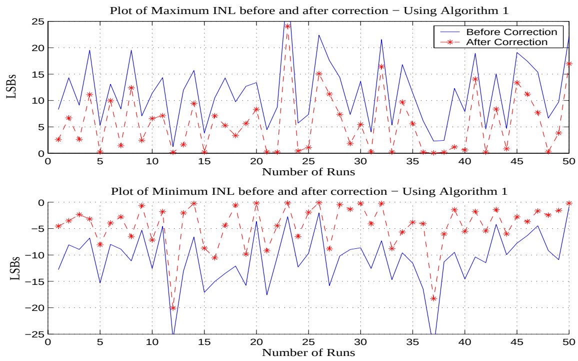

Fig. 1. Results of Algorithm 1

Algorithm 2 was then used to characterize the same ADC. $X_1(t)$ was chosen to be the same signal as that used in Algorithm 1, while $s_2(t)$,$f_3(t)$ and $f_4(t)$ were chosen as:

$$s_2(t) = 0.5(\cos(\pi t) + 1)$$
$$f_3(t) = \sin \pi t \text{ and } f_4(t) = \sin 2\pi t \qquad (23)$$

a_3 and a_4 were selected to be -0.05 and -0.02 respectively, limiting the accuracy of input signal to be less than 6 bits. The resistances were again varied by +/-0.5R_o and the number of samples was restricted to 4. Figure 2 gives a plot of the INL introduced and the INL after correction using the estimated value. As can be seen, the INL after correction reduced to within $\pm 0.5LSB$ from an uncorrected value of $\pm 10LSB$, indicating that the information from multiple inputs can lead to much better characterization. Similar to algorithm 1, the simulation was repeated 50 times and the results of maximum and minimum INL before and after correction were recorded and are shown in Figure 3. It can be seen that after correction, the accuracy of the ADC is improved from 7-8 bits to 11 bits.

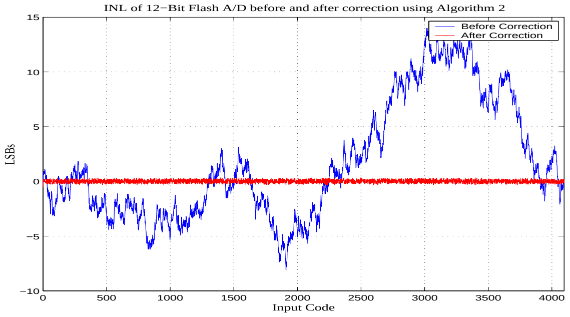

Fig. 2. INL before and after correction using Algorithm 2

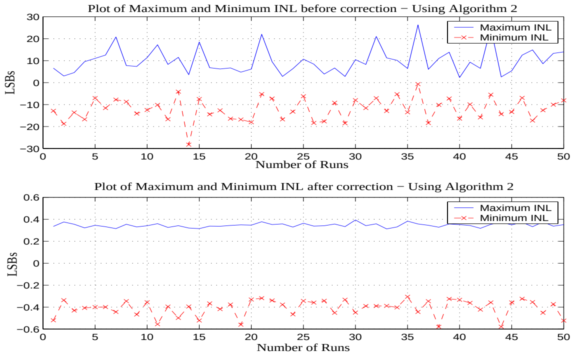

Fig. 3. Results of Algorithm 2

5. CONCLUSION

Two different algorithms based on the blind approximation approach have been proposed to characterize an A/D converter that is driven by inputs that are not precisely known. One algorithm is based on a single input signal and works well if the actual errors in the converter are not very large. The second algorithm uses outputs obtained from multiple input signals and identifies the non-linearities of the converter to a higher accuracy. Simulation results on a 12-bit flash A/D converter show that the second algorithm can correct INL to within +/-0.5LSBs if the uncalibrated A/D has an INL due to random Gaussian variations of the trip point of around 32LSB.

6. REFERENCES

[1] M. Burns and G.W. Roberts, *An Introduction to Mixed-Signal IC Test and Measurement*, Oxford University Press, New York, USA, 2000.

[2] Jiun-Lang Huang and et.al., "A BIST Scheme for On-chip ADC and DAC Testing," in *Design, Automation and Test in Europe Conference & Exhibition*, 2000, pp. 216–220.

[3] D.A. Johns and K. Martin, *Analog Integrated Circuit Design*, John Wiley & Sons, Inc., 1997.