A novel implementation of the histogram-based technique for measurement of INL of LUT-based correction of ADC

Vincent Kerzérho, Serge Bernard, Florence Azaïs, Mariane Comte, Olivier Potin, Chuan Shan, Guilherme Bontorin, Michel Renovell

A novel implementation of the histogram-based technique for measurement of INL of LUT-based correction of ADC

V. Kerzérho, S. Bernard, F. Azaïs, M. Comte, O. Potin, C. Shan, G. Bontorin, and M. Renovell

LIRMM, CNRS-University of Montpellier 2, 161, rue Ada, Montpellier 34095 France

Abstract

The histogram-based technique is commonly used for testing of Analog-to-Digital Converters (ADC). One of the parameters measured thanks to this technique is the Integral Non Linearity (INL). INL is also used as an initial data related to the ADC performances for the computation of a correction table in case of a LUT-based correction technique. In this context of embedded INL measurement and embedded computation of the table for LUT-based correction of ADC, we propose a new implementation establishing what we consider the best trade-off between silicon area overhead and computing time. We compare our solution with the state of the art: (a) with VHDL-level simulation we compare time performance, and (b) with FPGA placer we estimate the final surface head-out.

Keywords: Analog-to-Digital Converter (ADC); Look-Up Table (LUT); Integral Non-linearity (INL); Histogram-based ADC correction

1. Introduction

The use of Lookup-Table (LUT) for improving Analog-to-Digital Converter (ADC) performances is common and a 20-years old research topic [1-3]. The fundamental principle consists in integrating a table that contains correction codes pre-computed during production phase, based on non-linearity measurements. In the application, the output code of the ADC is used to address the LUT that substitutes the output code with the correction code stored in the table. This method has demonstrated a great interest to avoid performance loss due to manufacturing process variability. A recent trend [3] tends to integrate resources dedicated to signal generation and computation in order to embed the correction table generation.

The advantage of this approach is that it can be used not only during the production phase to correct errors due to manufacturing process variability, but also in-situ in the application in order to correct for aging effects or specific use conditions.

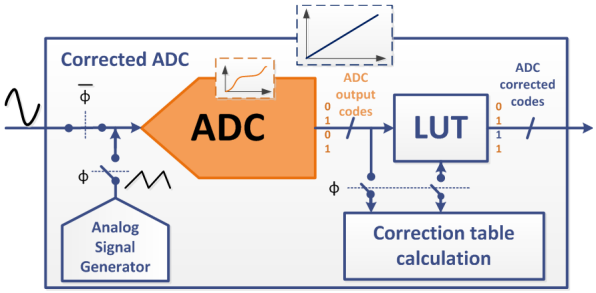

Figure 1 : in-situ LUT-based ADC correction.

In this publication we assume that the case study is in the context of LUT-based corrected ADC. According to the significant literature [4-7] on embedded analog signal generators, we assume that the generator will be accurate enough to implement the LUT-based self-correction technique. The study strictly focuses on the digital signal processing required to operate an embedded measurement of INL and to compute the correction table.

Our innovative computation technique reuses the embedded resources for measuring INL and computing the correction table. The main advantages

are: (a) the reduced silicon area overhead and (b) the fast computation time. The reduced silicon area overhead impact directly the economical viability of our solution but allows us to maintain the reliability of the circuit despite of its operation computation is not related to the function. The fast computation time is also mandatory to reduce the time for which the circuit is not operative because of the procedure for correction table computation.

In section 2 we revise the state of the art in LUT-based correction and embedded INL computation techniques. The Section 3 states our three-step method that measures the INL and computes the correction table. In section 4 we present our simulation results and compare them to the state of the art, both for silicon overhead and computing load. At last, section 5 concludes our method for INL computation for LUT-based correction of ADC, with what we consider the best trade-off between silicon overhead and computing timing.

2. State of the art

2.1. LUT computation

The intrinsic purpose of the LUT-based correction of ADC is to compensate non-linearity (NL) errors of the component. The LUT computation has been widely addressed as a research topic. Static [1] and dynamic [2, 3] characteristics of NL have been considered among different solutions. A widely used solution (e.g. [8]) consists in computing the correction table embedded in the LUT, using the measure of the Integrated Non Linearity (INL). The most common technique for INL measurement is called the histogram one [1]. The characteristics of this technique are its high accuracy and the large number of samples required for computation. This parameter is historically measured using a DSP-based measurement setting and an external Automated Test Equipment (ATE) [9, 10].

INL measurement using histogram-based technique suffers from limitations [11]. Indeed defective behaviors such as non-monotonicity or hysteresis are not detected. As a consequence the LUT-based correction technique cannot address such problems. If these non-detected defects are critical for the targeted product, a specific strategy should be defined.

2.2. Techniques for embedded INL measurement

The direct implementation of the histogram-based method for embedded INL measurement is not a viable option as it requires a huge amount of on-chip resources. Alternative implementations of the histogram-based method have been proposed in the literature for BIST purpose, based either on time decomposition [12] or space decomposition [13]. The main interested of the solution developed in [12] is that it offers a very low silicon area overhead but at the price at relatively long test time. In contrast, the solution developed in [13] presents interesting test time features but necessitates significant silicon area overhead.

In order to avoid constraints due to histogram technique, some publications presents non-histogram techniques embedded for BIST purpose [14, 15]. Authors of [14] describe a test method using a noise as stimulus. This method is very interesting as it reduces constraints on stimulus generation. However the integration of this method seems difficult mainly because it requires computing a Fast Fourier Transform (FFT), moreover the integration has not been demonstrated. [15] presents a very interesting test method allowing a concurrent procedure to the normal operation. This method apparently gives accurate measurements of INL, the main drawback is the potential significant silicon area overhead, notably because of the need of an embedded DAC or ADC dedicated to test.

For a BIST purpose, the integration of INL measurement suffers from particular constraints. Indeed dedicated silicon resources should be minimized but the INL measurement should be accurate in order to ensure the test accuracy.

A way to address the issues related to the Built-In Self-test of ADC is to relax constraints on stimulus generation and to rely the test accuracy on digital signal processing such as double-histogram methods [16 -18]. In our case study, the problem is addressed differently. Indeed the integration of efficient signal generators is research topic by itself, providing already significant results [19]. As a consequence, our interest is in the minimization of digital resources and computing time, in order to limit the impact of

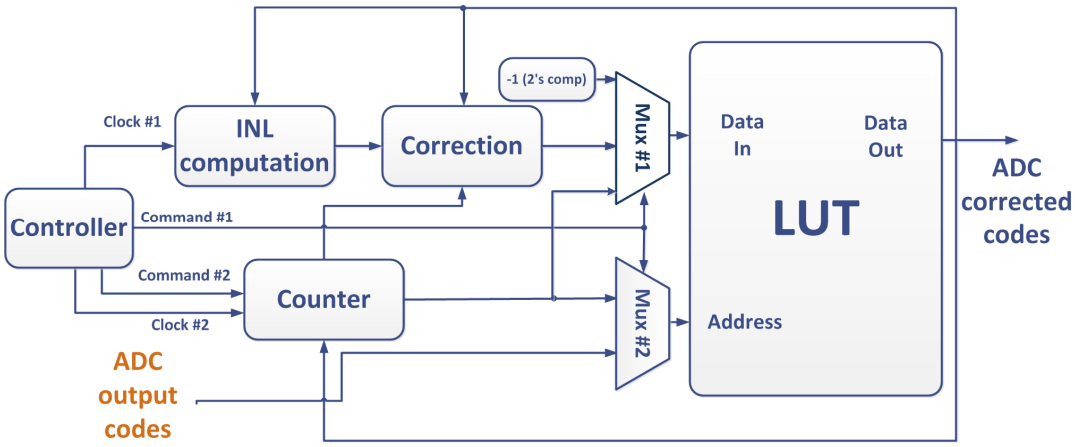

Figure 2 : hardware architecture for LUT computation

the measurement process as it will be done on-chip notably for the computation of correction table.

In an in-situ LUT-based correction context, the constraints are different. Indeed important resources are available for the INL computation (the LUT), on contrary as the LUT computation is done in-situ, the INL measurement should be done as fast as possible. Like for BIST purpose, in order to ensure the accuracy of the correction the INL measurement should be accurate. The most accurate INL measurement technique is the histogram one. As a consequence, we have developed an embedded histogram computation algorithm to optimize the study case constraints: surface, time and accuracy.

3. New technique for fastening the embedded histogram-based INL measurement

In the LUT-based context, we propose an optimized technique for embedded INL measurement. This technique aims to adapt the greedy linear histogram technique according to the available resources; in this case it means the memory used to store the correction table. The purpose is to reduce the computation time. Indeed in the case of in-situ LUT computation, the processing should be done when the ADC is not functioning.

The hardware resources required to implement the LUT-based INL measurement are presented in Fig. 2.

The controller aims to manage the clock domains synchronizing the different blocks. It also commands the multiplexers (MUX #1 and #2) and counter block. The counter block has two functions: counting from 0 to 2^n (n the resolution of the ADC) and incrementing its input value.

The INL computation block computes the INL values from the DNL values temporarily stored for each code in the LUT. The correction block determines the correction table based on computed INL values. The correction codes are then loaded in the LUT and used in the application.

The procedure for computing correction codes and loading the LUT is made of three steps.

3.1. First step: initialization of the LUT

The first step of the method consists in initializing all the words from the LUT to -1 in two's complement. The words are initialized to -1 considering the case of an ideal converter, for which each code should appear ones during the procedure. One increment of a word corresponding to a code during the procedure will set the final value of the word to zero. It means no non-linearity for this code. For a 10-bit ADC, the words are 10-bit long, 5 bits for the integer part and 5 bits for the decimal part. Then each word is initialized at "1111100000". The size of the word is related to the ADC resolution because this memory will be used later to store the corrected codes of the ADC. The distribution of bits between integer and decimal parts is related to the targeted accuracy of DNL/INL measurements. The precise explanation is given in the following section.

In order to generate the whole addresses of the LUT, the counter block is set by the controller in its first mode, consisting in counting from 0 to 2^n-1. In order to provide the -1 value to each word of the LUT, the controller commands the multiplexer #1.

3.2. Second step: Differential Non-Linearity (DNL) computation

The second step consists in computing the DNL using the initialized LUT.

According to fundamentals of histogram-based measurement of DNL [9], the DNL measurement using a ramp stimulus is given by equation 1.

$$DNL(i) = \frac{H_{exp}}{H_{ideal}} - 1 \qquad \text{equation 1}$$

Where H_{exp} is the real and H_{ideal} the ideal number of occurrences of code i. When the expected and the real number of occurrences of code i are the same, there is obviously no error and the DNL(i) is equal to zero. The accuracy of the measurement is given by equation 2.

$$\delta_{DNL} = \frac{1}{H_{ideal}} \qquad \text{equation 2}$$

The relation between the total number of samples and H_{ideal} is given by equation 3.

$$H_{ideal} = N_T \frac{FS}{2^n A_{in}} \qquad \text{equation 3}$$

Where N_T is the total number of samples, n the resolution of the ADC, FS the full scale of the ADC and A_{in} the amplitude of the ramp. In the case of FS and A_{in} are equal the total number of samples is equal to H_{ideal} multiplied by 2^n.

Then the operations done in the second step are the following. The controller commands the multiplexers in order to provide to the data input, the counter output and to address the LUT by the ADC output code. The generated stimulus is supposed to be a ramp with a high linearity. Every time the code *i* gets out of the ADC, the word at the address *i* is incremented by 1. The 1 is added to the decimal part. The increment is done by the counter which is set in his second function by the controller. Now let's consider the case study given in the previous section. The 10-bit word has been distributed in 5 bits of integer and 5 bits of decimal, which leads to the -1 value coded by the word 111110000 in 2's complement. The choice of 5 bits for decimal part of word is related to the definition of H_{ideal}, which is set to 32 (2^5). According to equation 2, this value is high enough to provide an accurate measurement but can be adapted to the given test conditions in order to reduce the test time. By the end of the ramp, each word contains the DNL of the code corresponding to its address. Indeed, for instance, if code *i*, is fault-free, then H_{ideal} will be equal to H_{exp} and the corresponding word will be incremented by $32=2^5$. By the end the word initially equal to 1111100000 will be 0000000000. As illustration, let us consider the case for which DNL(i) is equal to 1.5LSB. Then H_{exp}(i) is equal to 80 and the word corresponding to code *i* is incremented 80 times providing the value 0000110000. As a consequence, by defining the right initial value (-1), the right size of the decimal part according to H_{ideal} and by incrementing the word each time the code occurs, the DNL value can be directly read from the final value of the corresponding word without any expensive division as requested by equation 1. In case of a missing code, the corresponding word has not been incremented and the DNL value is -1. Despite missing codes can be identified, the considered correction technique cannot correct this kind of error. To give an explanation, let us consider the worst-case scenario. The output code is a constant value whatever the input voltage is. There are approximately 2^n-1 missing codes. As there is no information at the output, rather than a code, it is impossible to know what the input voltage is to correct the corresponding code.

It is important to notice that the controller generates different clock frequencies for the counter in the first and second steps. Indeed for the first step the initialization is done as fast as possible. For the second step the counter is synchronized with the ADC, i.e., operations of addressing a LUT word and incrementing its value are done during an ADC clock cycle

3.3. Third step: INL and correction computation

The final step consists in computing INL values from DNL values stored in the LUT in the previous step and generating the correction table. In this step, the data output of the LUT is connected to the input of the INL computation block, the data input of the LUT is connected to the output of the correction block and the address input of the LUT is connected to the counter output.

The counter is set by the controller in the function counting from 0 to 2^n-1. The output value of the counter is used to address the LUT and to take out the DNL value corresponding to the addressing code. The INL is then computed by the INL block using the following equation

$$INL(i) = \sum_{k=0}^{i} DNL(k) - 0.5 * DNL(i) \quad for\ i = 0..2^n - 1 \qquad \text{equation 4}$$

The corrected codes provided to the LUT are computed using equation 5.

$$corr_{code(i)} = i - [INL(i)] \qquad \text{equation 5}$$

where [ ] corresponds to the round function.

Once the correction table is loaded in the LUT, only the LUT is used during the normal operation mode: the ADC output code is provided as an address to the LUT, which substitutes this code with the corrected one.

4. Simulation results

The objective of the presented work is not to discuss the effectiveness of the considered correction method. This subject has been addressed in [20]. In order to evaluate the performance of the proposed implementation for INL histogram-based measurements, we have focused on two criteria: computation time and silicon area overhead.

References [12] and [13] provide equations to compute the theoretical computation time. Equation 6 is the computation time using the time-decomposition algorithm [12].

$$T_1 = 2.25 * \frac{H_{ideal}}{F_s} * 2^{2n} \qquad \text{equation 6}$$

Where n is the converter resolution, F_s the sampling frequency.

Equation 7 is the computation time using the space decomposition algorithm [12].

$$T_2 = 3 * \frac{H_{ideal}}{F_s} * 2^n \qquad \text{equation 7}$$

Table 1:Comparison of computing time for different embedded histogram-based methods.

Implementation	Fs=1MHz	Fs=20MHz	Fs=50MHz
This	2.1 ms	102 µs	41 µs
Time-decomposition-based method [6]	294 ms	14.8 ms	5.9 ms
Space-decomposition-based method [7]	6.2 ms	307 µs	123 µs

The test time using our new implementation is given by equation 8.

$$T_3 = \frac{H_{ideal}}{F_s} * 2^n + \frac{4}{F_s} \qquad \text{equation 8}$$

The first part of the right side of equation 8 is the time for counting the occurrences of codes resulting in the calculation of the parameters. The second part of the right side of equation 8 is the time for the initialization. According to equation 7 and 8, the computation time of our new approach is almost three times faster than the space decomposition approach [13]. Table 1 presents the comparison between the three implementations for a 6-bit ADC and different sampling frequencies. The proposed implementation permits to drastically reduce the computation time compared to the time decomposition method [12], and is almost three times faster than the space decomposition approach [13]. Because [13] references its results in comparison with time-decomposition technique: “The hardware overhead is about four times of the hardware overhead of the time-decomposition technique", later consideration about area overhead due to our new implementation will be only compared to the time-decomposition method.

Table 2:

Test time vs. ADC resolution and sampling frequency

ADC resolution	Fs (MHz)	test time (µs)
8 bits	1	9728
	20	486
	50	195
10 bits	1	38912
	20	1946
	50	778
12 bits	1	155648
	20	7782
	50	3113
14 bits	1	622592
	20	31130
	50	12452

The new approach has been implemented on a FPGA for different ADC resolutions. Table 2 provides the computation time for different ADC resolutions and sampling frequencies. Even for a high-resolution converter of 14 bits and a relatively low sampling frequency of 1MHz, the computation time remains below 1 second while the time decomposition approach necessitates more than 19,000 seconds.

Table 3 presents the silicon area overhead resulting from the FPGA implementation of our new method and from the integration of the method introduced in [12]. Both methods have been implemented for a 6-bit ADC. The LUT area is not included in the following figures as this study has been done in a context of LUT-based correction of ADC, and this area is the same for both methods.

The silicon area overhead is slightly the same for both methods. The new implementation requires 15% more cell area, but this logic area remain small compared to the LUT area. Indeed the LUT is made of 2^n word of n bits, where n is the ADC resolution.

Table 3:

Silicon area overhead for two methods

Implementation	Cells	Cell area (mm²)	Net area (mm²)
This	257	0.028701	0.004455
Time-decomposition-based method [7]	223	0.025917	0.004464

5. Conclusion

LUT-based correction of ADC is effective. The current trend tends to integrate the procedure to compute the correction table. Our innovative computation technique has demonstrated the interest in re-using the embedded resources for measuring INL and computing the correction table. The silicon area overhead required for correction-table calculation is very small compared to the LUT area. The main advantage is the fast computation time. Indeed, as we are in the case of self-adaptive correction, the time for which the circuit is not operative because of the correction procedure should be as short as possible.

References

[1] F.H. Irons, D.M. Hummels, S.P. Kennedy, “Improved Compensation for Analog-to-Digital Converters”, IEEE Transactions on Circuits and Systems, Vol. 38(8), pp. 958 – 961, 1991.

[2] S. Medawar, P. Händel, N. Björsell, M. Jansson, “ADC Characterization By Dynamic Integral Nonlinearity”, IMEKO 13th Workshop on ADC Modelling and Testing, pp 1037-1042, 2008

[3] H. Fraz, N. Bjorsell, J.S. Kenney, R. Sperlich, "Prediction of Harmonic Distortion in ADCs using dynamic Integral Non-Linearity model," IEEE Behavioral Modeling and Simulation Workshop,.pp.102-107, 17-18 Sept. 2009.

[4] B.Provost, E. Sanchez-Sinencio "On-chip ramp generators for mixed-signal BIST and ADC self-test," IEEE Journal of Solid-State Circuits, vol.38, no.2, pp. 263- 273, Feb 2003

[5] Jing Wang, E. Sanchez-Sinencio, F. Maloberti, "Very linear ramp-generators for high resolution ADC BIST and calibration,", Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, vol.2, pp.908-911 2000

[6] B.K. Vasan, S.K. Sudani, D.J. Chen, R.L. Geiger "Sinusoidal signal generation for production testing and BIST applications," IEEE International Symposium on Circuits and Systems, pp.2601-2604, 20-23 May 2012

[7] F. Azaïs, S. Bernard, Y. Bertrand, X. Michel and M. Renovell “A low-cost adaptive ramp generator for analog BIST applications”, Proc. VLSI Test Symposium, pp. 266-271, 2001

[8] S. Bernard, F. Azaïs, M. Comte, O. Potin, V. Kerzérho and M. Renovell, “Adaptive LUT-based System for In Situ ADC Auto-correction”, IEEE 16th International Mixed-Signals, Sensors and Systems Test Workshop (IMS3TW), pp. 1-6, 7-9 June 2010

[9] “IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters”, IEEE Std 1241-2000

[10] M. Mahoney, "DSP-Based Testing of Analog and Mixed-Signal Circuits", IEEE Computer Society Press, ISBN 0-8186-0785-8, 1987.

[11] S. Max, "IEEE Std 1241: the benefits and risks of ADC histogram testing," Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, vol.1, pp.704-709, 21-23 May 2001
[12] F. Azaïs, S. Bernard, Y. Bertrand, M. Renovell, "Implementation of a Linear Histogram BIST for ADCs", in proc. of Design, Automation and Test in Europe (DATE) Conference, pp.590-595, 2001
[13] J. Ren, J. Feng, H. Ye, "A Novel Linear Histogram BIST for ADC", 9th International Conference on Solid-State and Integrated-Circuit Technology (ICSICT), pp.2099-2102, 20-23 Oct. 2008
[14] M. da Glória Cataldi Flores, M. Negreiros, L. Carro, and A.A. Susin "INL and DNL Estimation Based on Noise for ADC Test", IEEE Transactions on Instrumentation and Measurement, Vol. 53, no.5, pp. 1391- 1395, Oct. 2004
[15] A.J. Ginés, E.J. Peralías and A. Rueda, "An Adaptive BIST for INL Estimation of ADCs without Histogram Evaluation", IEEE 16th International Mixed-Signals, Sensors and Systems Test Workshop (IMS3TW), pp. 1-6, 7-9 June 2010
[16] B.K. Vasan, D.J. Chen, R.L. Geiger, "ADC Integral Non-Llinearity Testing with Low Linearity Monotonic Signals", IEEE Instrumentation and Measurement Technology Conference (I2MTC), pp.1-5, 10-12 May 2011
[17] E. Korhonen, C. Wegener, J Kostamovaara, "Combining the Standard Histogram Method and a Stimulus Identification Algorithm for A/D Converter INL Testing With a Low-Quality Sine Wave Stimulus", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 57, no.6, pp.1166-1174, June 2010
[18] M.A. Jalon, E. Peralias, "ADC Non-Linearity Low-Cost Test Through a Simplified Double-Histogram Method", Journal of Electronic Testing: Theory and Applications, Vol .26, Issue 1, pp 47-58, 2010
[19] M.J. Barragan, D. Vazquez, A. Rueda, " Analog Sinewave Signal Generators for Mixed-Signal Built-in Test Applications", Journal of Electronic Testing: Theory and Applications, Vol. 27, Issue 3, pp. 305-320, June 2011
[20] V. Kerzérho, V. Fresnaud, D. Dallet, S. Bernard, and L. Bossuet, "Fast Digital Post-Processing Technique for INL Correction of ADC: Validation on a 12-bit F&I ADC," accepted for publication in IEEE Transactions on Instrumentation & Measurement, 2010