10.2478/v10048-008-0019-6

A Virtual A/D Converter Testbench for Educational Purpose – Development and Results

P. Struhovský, O. Šubrt[1], J. Hospodka, P. Martinek

Department of Circuit Theory, Faculty of Electrical Engineering CTU Prague, Technická 2, 166 27 Prague, Czech Republic
E-mail: Struhovsky@centrum.cz, [hospodka,martinek]@fel.cvut.cz
[1]ASICentrum, Novodvorská 994, 142 21 Prague, Czech Republic, E-mail: Ondrej.Subrt@asicentrum.cz

This paper deals with a new concept of virtual testing engine for analogue-to-digital converters (ADCs). The whole system consists of program procedures to extract the most important ADC errors expressed in terms of integral and differential non-linearity (INL and DNL). The developed testbench is especially suitable for educational purpose because of modular conception of the system. The proposed testing engine is implemented in Maple™, bringing an ideal possibility to make a complex system for the simulations of ADC at the virtual level as well as at the circuit level. The system is a part of a complex environment using the Servo-loop and the Histogram method, combining their features so as to obtain high level of versatility. However, in this paper we concentrate only on the results from the Servo-loop method. The Servo-loop solution proposed here employs an effective search algorithm and improves convergence properties resulting in a significant reduction of the simulation time.

Keywords: A/D Converter testing, behavioural modelling, Servo-Loop method, Integral and Differential Non-linearity

1. Introduction

THE MAIN TASK of testing analogue-to-digital converters is to obtain high performance ADC with the short response time and high resolution. The ADC performance is expressed in terms of integral and differential non-linearity (INL and DNL) and depends on many parameters present in the analogue design part. According to the test setup, the existing performance extraction methods can be classified into closed-loop or open-loop category [1]. The virtual testing engine proposed in this article uses one of the widely used ADC performance extraction methods [2], the Servo-Loop, as a representative of the closed-loop category. The system part of the virtual testing engine is programmed in the software Maple™, allowing the algorithm execution and bringing a great advantage in the possibility of subsequent ADC modelling. The background of the conventional implementation of Servo-Loop method is also described here, pointing out not only its strong features but also disadvantages. Finally, some improvements of the weak points are presented, proposing a novel Servo-Loop implementation.

2. Subject and methods of servo-loop implementation

The core of a standard Servo-Loop implementation [2], [3] is a feedback loop realized across the ADC under test, analogue integrator and digital comparator blocks. Subsequently, the algorithm searches for the code transition level of a given code, performing a conventional linear search. An inversely-proportional relationship between the accuracy and the number of iterations should be pointed out. Usually, the algorithm performance is acceptable for the direct testing of low resolution A/D converters. However, to speed-up the verification time for high performance ADCs, a more efficient search algorithm would be advisable.

Referring to the above-mentioned advantages and weak points of the standard Servo-Loop implementation, our work proposes an innovative concept of joining the Servo-Loop method into a compact testing engine offering improvements in the performance extraction algorithm. The device under test is incorporated in the form of a behavioural ADC model and the testing engine outputs its extracted integral and differential non-linearity. To verify the proposed virtual testing engine, we use a serial ADC model preceding the bit-by-bit conversion algorithm corresponding to 1-bit successive approximation scheme [4]. The behavioural ADC model (Fig.1) consists of three main blocks: multiplier-by-two, comparator and summator. In the first step, the input voltage u_{in} is multiplied by two (the result is denoted as residuum R_i) and compared with the reference voltage u_{ref}. If the value of residuum R_i is greater than reference voltage u_{ref}, the converted bit b_i is "1", otherwise the bit b_i is "0". The algorithm starts with the most significant bit (MSB) and continues down to the least significant bit (LSB). The above-mentioned approach can be generally used for an arbitrary number of bits.

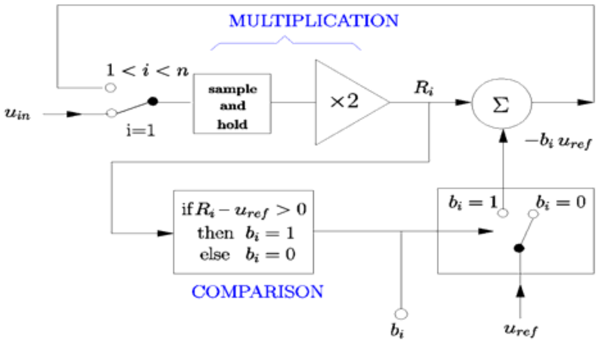

Fig.1 ADC model for verification of testing engine.

In our paper, the influence of two major non-ideality sources is considered in the form of the gain and offset errors. The gain error is produced by a non-ideal multiplication by two, where the multiply-by-two block (labelled as "x2" in Fig.1) has a factor of not exactly two. The offset error arises in the comparator block where the comparison process exhibits a non-zero voltage offset. It could be shown that the greater the gain or offset error is, the greater is also the integral non-linearity [7]. For further simulation purpose, the gain and offset errors are expressed in ppm.

A. Extraction of ADC Design Performance

In our work we present an innovative approach developed on assumptions discussed in [5], [6] which significantly accelerates the loop convergence and reduces the number of iterations. The implementation is depicted in Fig. 2 and works as follows. First, the searched code is fed into the code register, converted to a voltage value by an arbitrary DAC (not shown in the Figure) and stored into the cumulative-sum (cumsum) circuit. The voltage value is then converted to the digital number in the ADC. The output from the ADC is compared with the code register value. Digital comparator output is set to "1" in the case that the ADC output value is higher than the word from code register. Regarding to the state of comparator output, positive or negative incrementation is set. Note that the size of incrementation step decreases in each iteration. The loop iterates n-times for each searched code, i.e. the number of runs is $n \cdot 2^N$ where N is the number of ADC bits.

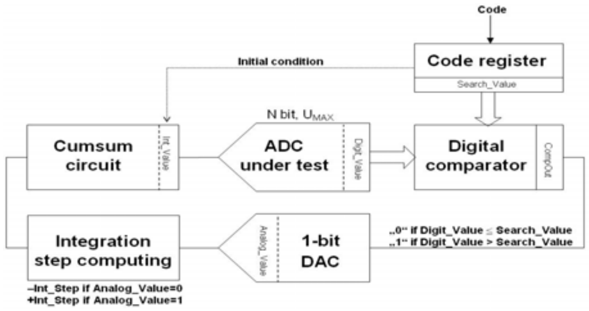

Fig.2 Block diagram for Servo-Loop Method.

B. Algorithm Background

To determine the INL and DNL characteristic of the ADC, we define a straight line $V_{ideal}(c)$ as the reference. The Servo-Loop algorithm extracts the upper code transition V(c), i.e. the voltage level, where the output spends half the time at codes greater or equal to c, to coincide with the ideal line:

$$V_{ideal}(c) = V(c_0) + (c - c_0) \cdot V_{lsb} \tag{1}$$

where

$$V_{lsb} = \frac{V(c_{fs} - 1) - V(c_0)}{c_{fs} - c_0 + 1} \tag{2}$$

Based on (1) and (2), we compute the INL and DNL deviation from the ideal characteristic as:

$$INL(c) = \frac{V(c) - V_{ideal}(c)}{V_{lsb}} \tag{3}$$

$$DNL(c) = \frac{V(c) - V(c-1)}{V_{lsb}} \text{-1} \tag{4}$$

After N_{cycle} iterations, the code edge value is returned from the algorithm. A question arises about the accuracy (resolution) of the returned value as a function of the given number of iterations. In [5], a simple formula for estimating the accuracy of code edge values was derived:

$$\Delta_N^{LSB} = \varepsilon^{Ncycle-1} \tag{5}$$

where Δ_N^{LSB} is the *relative accuracy* (expressed in LSB) denoting the difference between the code edge values of the last two iteration steps. To demonstrate the accuracy of the proposed Servo-Loop implementation, a comparison against conventional linear search method is given. Performing the *linear search*, the number of cycles required for corresponding accuracy is:

$$N_{LIN} = \left\lceil \frac{1}{\Delta_{LIN}^{LSB}} \right\rceil \tag{6}$$

where $\lceil\ \rceil$ denotes the upper nearest integer. As a result of (6), *for 0.01 LSB accuracy, 100 iteration cycles are needed.* In the *proposed Servo-Loop algorithm*, the number of cycles to obtain the accuracy Δ_N is given by a re-arrangement of (5):

$$N_{cycle} = \left\lceil 1 + \frac{\log(\Delta_N^{LSB})}{\log(\varepsilon)} \right\rceil \tag{7}$$

Subsequently from (7), setting the damping factor to e.g. ε = ⅔, it follows that *for 0.01 LSB accuracy only 13 cycles are needed.* The difference between the two methods becomes even more significant when higher accuracy levels are required, e.g. Δ_N^{LSB} =1 mLSB.

Compared to the standard solution [2], [3], the asset of our Servo-Loop implementation is the following:

- Cumsum circuit applies *a priori* known values to the input signal. Therefore, there is no need to check the integrator output as it is done in the standard implementation.
- Convergence process is assisted by an initial condition and by adaptive step refinement of the cumsum block. At this point, adjustment of the step size helps to accelerate the search for the loop equilibrium point as there are less iterations needed to maintain the same INL accuracy. The search complexity is changed from linear to logarithmic.

3. Results

In the following, we introduce the computational results of the Servo-Loop implementation with the ADC model. The number of bits was set to N=12 and the offset and gain errors were adjusted to 500 ppm or 1000 ppm, respectively. Fig.3 and Fig.4 present the INL plot for the gain and offset errors. As an inherent property of the ADC architecture, the local extrema of INL occur at binary-weighted code values.

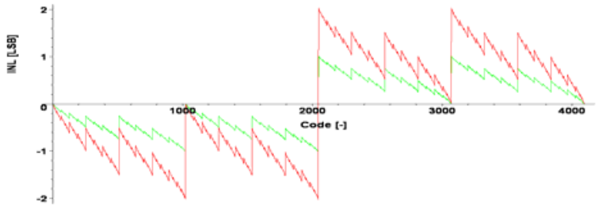

Fig.3 Servo-Loop INL plot with 500 ppm and 1000 ppm gain error.

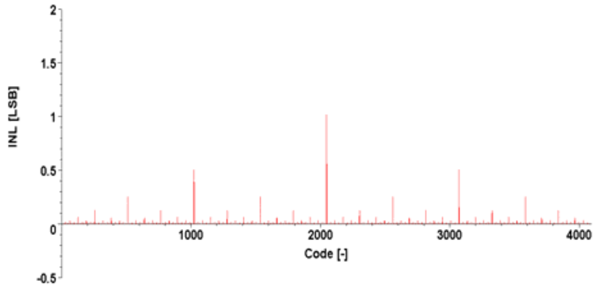

Fig. 4a Servo-Loop INL plot with 500 ppm offset error.

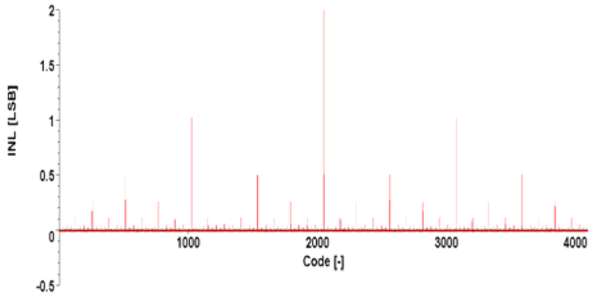

Fig.4b Servo-Loop INL plot with 1000ppm offset error.

The verification of linearity in the sense of error scaling [7] is shown in Fig.5. It is interesting to note that the shape of the linearity difference in case b) corresponds to an equivalent error mechanism of additive nature present in the ADC model and/or virtual testing environment.

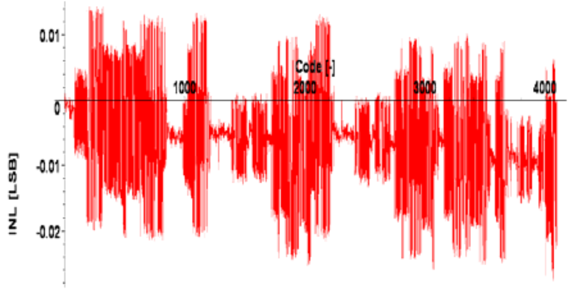

Fig.5a Servo-Loop linearity difference generated by gain error.

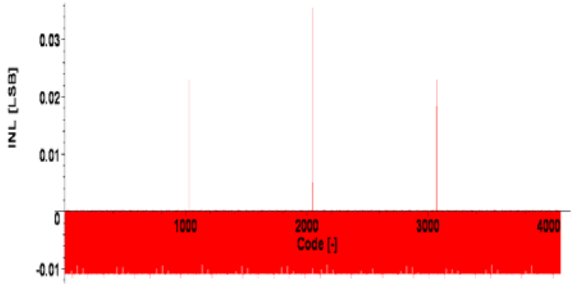

Fig.5b Servo-Loop linearity difference generated by offset error.

Accuracy of the whole Servo-Loop algorithm can be defined as the noise floor taking the form of a residual INL for the case if an ideal ADC is tested. In our case, we evaluated the INL by the zero offset and gain errors. At this point, interesting is the zoomed residuum depicted in Fig.6, giving rise to additional errors of multiplicative nature.

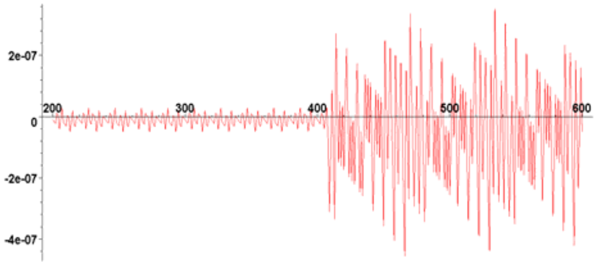

Fig.6 Zoomed residual INL for Servo-Loop.

Table 1 shows an overview of the total simulation time and magnitude of the residual INL, compared for various ADC resolutions. An ideal ADC model is used for this simulation, with zero magnitudes of all error sources. The parameters of the used PC are: Intel Core2 CPU T5600 @1,85GHz, 1GB RAM.

ADC resolution n_{bit}	All-code INL simulation time t_{sim}	Residua INL_{res} [LSB]
8	3.6 sec	$3.4\cdot10^{-8}$
10	15.9 sec	$3.8\cdot10^{-7}$
12	1 min 20 sec	$6\cdot10^{-7}$
14	6 min 13 sec	$2\cdot10^{-6}$

Table 1 All-code INL Computation Time and Residual.

4. Discussion and conclusions

This work proposes a virtual testing engine for A/D converters applying the Servo-Loop method. Particular improvements were taken on the testing engine so as to accelerate the convergence and increase the algorithm efficiency. As concerns the system implementation, the extensive use of Maple™ allows to create circuit-similar ADC models with higher level of complexity because of the availability of built-in packages for symbolic circuit analysis. The extended performance of the virtual testing engine is documented on a simulation example of a 12-bit ADC and can be further extended. The next development of the testing environment will be focused on extraction of dynamic ADC errors.

The first asset of our Servo-Loop implementation compared to the standard solution [3] is that the cumsum circuit applies *a priori* known values to the input signal. Therefore, there is no need to check the integrator output as it is done in the standard implementation. The second advantage is that the convergence process is assisted by an initial condition and by adaptive step refinement of the cumsum block. At this point, adjustment of the step size helps to accelerate the search for the loop equilibrium point as there are less iterations needed to maintain the same INL accuracy. The search complexity is changed from linear to logarithmic.

ACKNOWLEDGEMENTS

The work has been supported by the grant GAČR 102/07/1186 from the Grant Agency of the Czech Republic, carried out in co-operation with the Department of Circuit Theory Faculty of Electrical Engineering, CTU Prague, and the ASICentrum Prague. The work has also been supported by the research programme MSM6840770014 of the Czech Technical University in Prague. Maple™ is a trademark of Waterloo Maple Inc.

REFERENCES

[1] Capofreddi, P.D. & Wooley B.A. (1999). The Efficiency of Methods For Measuring A/D Converter Linearity. *IEEE Transactions on Instrumentation and Measurement* 48 (3), 763-769

[2] Burns M. & Roberts G.W. (2001). *An Introduction to Mixed-Signal IC Test and Measurement*. (pp. 447-481). Oxford University Press, New York

[3] The Institute of Electrical and Electronics Engineers, Inc. (2000). IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters. IEEE Std. 1241-2000. New York.

[4] Nairn D. (1993). *Algorithmic and Pipelined A/D Converters*, in: C. Toumazou, J. B. Hughes, N.C. Battersby (Eds.), Switched-Currents an analogue technique for digital technology (pp. 304-322). United Kingdom, Peter Peregrinus Ltd.

[5] Šubrt O. (2005). *Application of SI Technique in A/D Conversion and Its LEMMA-aided Design Evaluation*. Doctoral Thesis, Faculty of Electrical Engineering, CTU Prague, Czech Republic

[6] Šubrt, O., Martinek, P. & Wegener C. (2007). A Contribution to Advanced Extraction Methods for Static ADC Non-linearity, In: Proceedings of the IEEE Conference IMTC, 1-3 May 2007 (CD-Rom). Warsaw, Poland.

[7] Carrol, B., Wegener C. & Kennedy M.P. (1999). Lemma-ADC: The Linear Error Mechanism Modelling Algorithm Applied to A/D-Converters. In: Proceedings of the IEE Conference ADDA 1999 (pp. 145-148), United Kingdom