A Powerful Extension of Servo-Loop Method for Simulation-based A/D Converter Testing

Ondřej Šubrt[1,2], Pravoslav Martinek[1], Carsten Wegener[3]

[1] *Department of Circuit Theory, Faculty of Electrical Engineering CTU Prague, Czech Republic*
[2] *ASICentrum, Prague, Czech Republic*
[3] *Infineon Technologies AG, Munich, Germany*
e-mail: Ondrej.Subrt@asicentrum.cz, Martinek@fel.cvut.cz, Carsten.Wegener@infineon.com

Abstract- This paper deals with the virtual testing environment for analog-to-digital converters (ADCs) employing a novel and powerful extension of the Servo-Loop method [1], [4]. We build an improved version of the Servo-Loop targeted to full transistor-level circuit simulation of static integral and differential ADC non-linearity. In comparison with the conventional implementation, the Servo-Loop version proposed was enhanced by an effective search algorithm. The algorithm was implemented as a versatile Servo-Looper tool written in Verilog-A language which is suitable for direct co-operation with most of the analog and mixed-signal simulators used in industry. The prospective advantage of our approach is the fact that the implementation in Verilog-A creates an ideal opportunity to build a complex environment comprising the virtual testing engine as well as the DUT in the form of circuit-level ADC design or its behavioral model. At this point, the powerful capabilities of the proposed Servo-Looper tool were successfully confirmed by a large simulation set performed on the ADC behavioral model and the full custom ADC design example. The paper presents the most significant results of the ADC simulation procedure.

Keywords- Mixed-signal, analog to digital converter, integral and differential nonlinearity, static ADC test, behavioral modeling

IWADC'07 topic - 2.2 ADC testing, 2.4 ADC/DAC modeling

I. Introduction

The performance of an A/D converter is usually expressed by *integral* and *differential non-linearity* (INL and DNL), being the most important metrics of the ADC transfer characteristic. The extraction of ADC design performance is a challenging task. Especially, in high-resolution A/D converters, the INL and DNL extraction is complicated by small error magnitudes present in the analog design part as well as by large number of digital states given by the number of ADC bits. According to the test setup, the existing performance extraction methods can be classified into *open-loop* or *closed-loop* category [1]. The *code-density* method [2], [3] is a typical representative of the open-loop category, producing an estimation of the code transition levels from the histogram of a large set of recorded ADC output codes. In our work, we propose an approach for performance extraction suitable for *full transistor-level simulation* of the integrated circuit design. In this case, statistical processing (e.g. a histogram processed by the code-density method) is not suitable because of the enormous simulation time requirements. Instead, the *Servo-Loop*, belonging to the closed-loop extraction methods is the primary candidate to perform this task. The main reason for

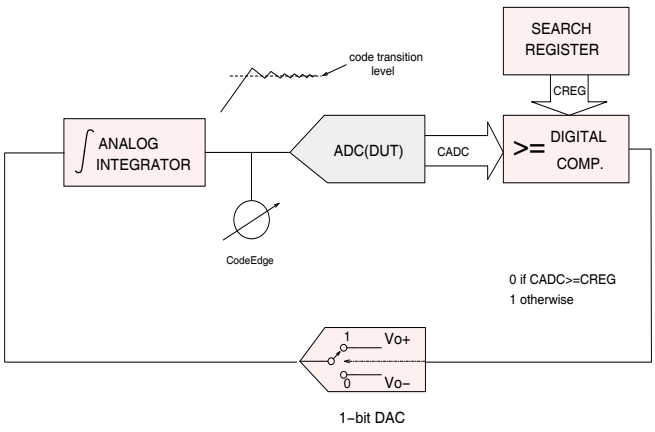

Fig. 1 Standard Servo-Loop implementation.

this choice is the inherent property of the Servo-Loop method, processing a single capture of the ADC response for a given small number of iteration steps. The main principle of a standard Servo-Loop implementation [4] is that a feedback loop is accomplished across the ADC under test, *analog integrator* and digital comparator blocks. This conventional Servo-Loop arrangement is depicted in Fig. 1. Subsequently, the algorithm searches for the code transition level of a given code, performing a conventional *linear search*. The Servo-Loop method is widely used for direct A/D converter test and measurement. However, to implement this method in a simulation environment of advanced IC design tools, improvements of the standard implementation are strongly advised as to increase the performance and efficiency of the search algorithm.

I. Proposed Novel Servo-Loop Arrangement

In this section, we propose a powerful extension of the Servo-Loop method to accelerate its convergence speed and reduce the total extraction time. In Fig. 2, the proposed new arrangement of the Servo-Loop method is shown, generally usable for simulation-based extraction of ADC INL and DNL. In comparison with the standard approach depicted in Fig. 1, the following innovations were suggested. First, instead of a conventional continuous-time analog integrator, a discrete-time version is used (denoted as *discrete integrator*). The next significant difference against the standard implementation is that an initial condition (implemented by the *CALDAC block*) is applied to the integrator output. In comparison with the standard approach from Fig. 1, the proposed novel implementation has two prospective advantages.

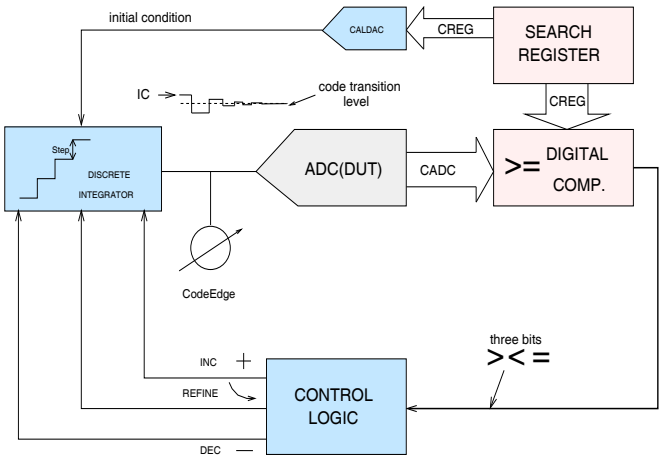

Fig. 2 Proposed Servo-Loop implementation.

1. **Discrete integrator applies a priori known step values to the ADC input signal.** From these values, the corresponding transition levels can directly be established (with no need to check or measure what the ADC input signal is).
2. **Convergence process is significantly accelerated** because of two reasons. First, an initial condition has been applied to the ADC input, based on the estimation provided by CALDAC block. During the first iteration cycle, the difference between the C_{ADC} and C_{REG} codes is therefore reduced, speeding up the feedback search for an equilibrium point. A small initial step of the integrator output can be therefore chosen, typically less than 1 LSB. The second reason of the convergence speed-up arises from the fact that the integrator step value is being continuously refined during the iteration process – for detailed description, refer to the next section.

II. Target Servo-Loop Algorithm

Figure 3 shows the details of the algorithm performed by the proposed Servo-Loop arrangement. In the following, an 8-bit ADC is assumed, characterized by the reference value, denoted as *Ref* in Fig. 3.
In the beginning, the appropriate system variables are initialized. Beside those already mentioned in the previous section, a new variable called *IsMissing* is added. The value of this boolean variable is initialized with true, and remains so if the code under test is missing in the ADC transfer characteristic.
The algorithm works as follows. Execution of the main cycle is conditioned by the decrementation of N_{cycle} variable, determining the number of iterations (e.g. $N_{cycle} = 10$). Inside the main cycle, the ADC conversion is performed first, resulting in the C_{ADC} output code. The obtained code is then compared with the C_{REG} value. If $C_{ADC} \geq C_{REG}$, the integrator output *(Int)* is decremented, otherwise it is incremented in order to find the lower transition level for code C_{REG}. In the algorithm, the amount by which the Int value is changed depends on the absolute difference between C_{REG} and C_{ADC} codes. If the codes are equal, the *Step* value is simply subtracted from Int. In the case of non-zero difference, the Int

value is changed by Step·(C_{REG}-C_{ADC}). Note that the C_{REG}-C_{ADC} term can be either negative or positive, indicating a decrement or increment, respectively. The multiplication by term $C_{REG} - C_{ADC}$ term represents another action as to achieve quick Servo-Loop convergence. The IsMissing variable is set to false when C_{ADC} equals at least once the value of C_{REG} during the iteration cycle. This implies the fact that the C_{REG} code is not missing on the ADC transfer characteristic. The last action item of the main cycle is to refine the Step variable. Its value is multiplied by a *damping factor* $0 < \varepsilon < 1$ as to optimize the Servo-Loop convergence.

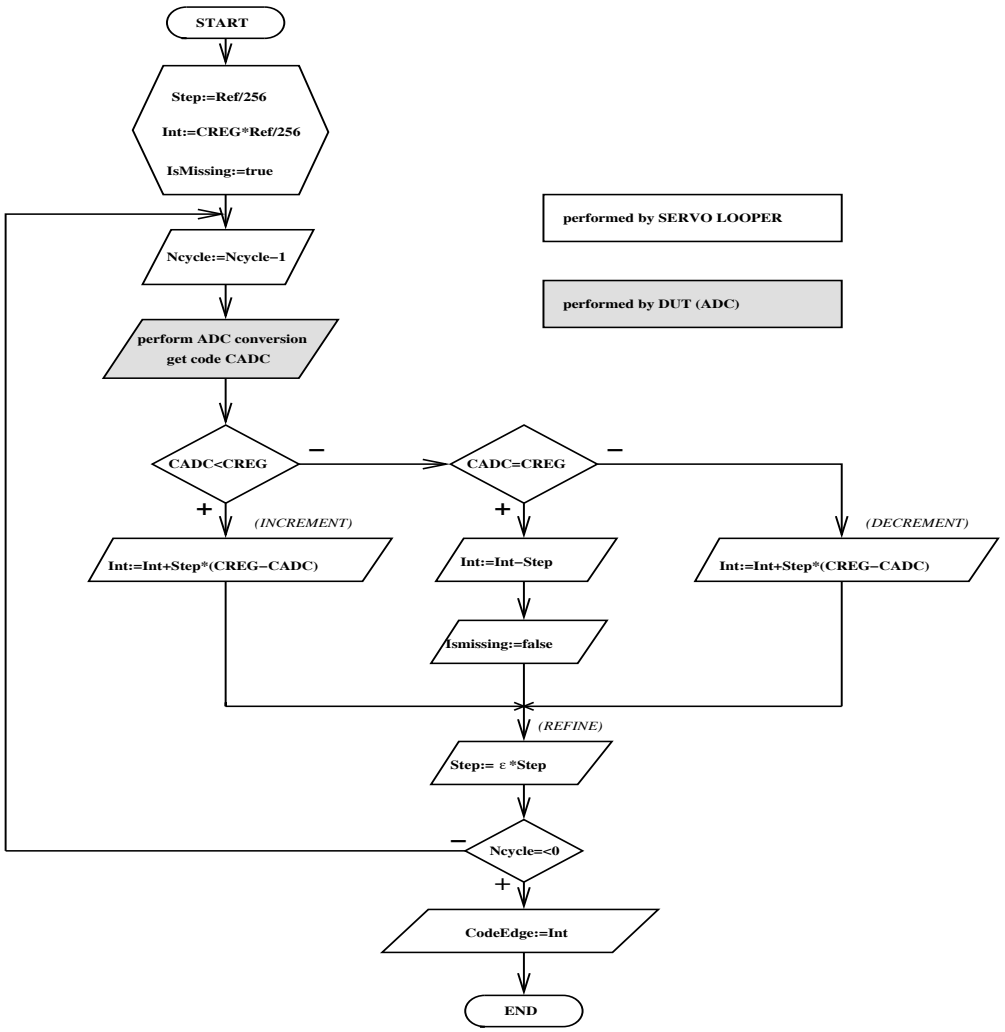

Fig. 3 Target ServoLoop algorithm.

A. Accuracy Issues

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 this paragraph, a simple formula for estimating the accuracy of code edge values is presented, with respect to the proposed Servo-Loop implementation.

The code edge values after particular iteration steps are then (assigning $X_C(0)$=Int as initial value):

$$\begin{aligned} X_C(1) &= \mathrm{Int} \pm \mathrm{Step} \\ X_C(2) &= (\mathrm{Int} \pm \mathrm{Step}) \pm \epsilon\,\mathrm{Step} \\ &\cdots \\ X_C(N_{cycle}) &= \mathrm{Int} + \mathrm{Step} \sum_{i=0}^{Ncycle-1} (-1)^{s(i)} \epsilon^i \end{aligned} \tag{1}$$

where the exponent s(i) is a series of zeroes and ones denoting a signature of the convergence process:

$$\{s(i),\ s(i) \in \{0,1\}\}_{i=0}^{N_{cycle}} \tag{2}$$

Let us define the *absolute accuracy* Δ_N as a difference between the code edge values of the last two iteration steps:

$$\Delta_N = X_C(N_{cycle}) - X_C(N_{cycle}-1) =$$

$$= \text{Step}\left(\sum_{i=0}^{Ncycle-1} (-1)^{s(i)} \epsilon^i - \sum_{i=0}^{Ncycle-2} (-1)^{s(i)} \epsilon^i \right) = \ \ldots \ = \underline{\text{Step} \cdot \epsilon^{Ncycle-1}} \tag{3}$$

Given in LSB, the accuracy can be expressed as:

$$\Delta_N^{LSB} = \epsilon^{Ncycle-1} \tag{4}$$

Example

To demonstrate the accuracy of the proposed Servo-Loop implementation, a comparison against conventional linear search method is given below.

Performing the *linear search*, the number of cycles required for corresponding accuracy is $N_{LIN} = \left[\frac{1}{\Delta_{LIN}^{LSB}} \right]$ where [ ] denotes the higher nearest integer. *Thus 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 re-arrangement of (4):

$$N_{cycle} = \left[1 + \frac{\log(\Delta_N^{LSB})}{\log(\epsilon)} \right] \tag{5}$$

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 =0.001 LSB.

B. System-level implementation

The algorithm described in the previous paragraph was implemented into the virtual testing environment as follows. Individual system blocks are included either as analog circuit schematics exported to SPICE netlists or as mixed-mode using behavioral description in Verilog-A language [5]. At the top-level, system blocks of both types are instantiated into the SPICE netlist and simulated in ELDO analog circuit simulator from Mentor Graphics [8]

The process of ADC performance extraction is managed by the Verilog–A program unit called *ServoLooper*, implementing the following functions:

- **initialization and clock signal generation** for the ADC under test
- **generation of the ADC input signal** (optionally current or voltage) and capturing the ADC output code
- **performing iteration process** and return the transition level for the code under test or the IsMissing value if the code does not appear
- **simulation data operations** – writing the extracted code edge values into text files, implementing *sweep mode* to obtain the whole transfer characteristic, featuring *restart option* to salvage or reinvoke existing simulation.

III. Simulation results

This section deals with the ELDO simulation results of the ADC environment introduced in the previous section. In order to verify the facilities of the proposed approach, *intrinsic performance of the ServoLooper unit* is extracted in paragraph A. For this purpose, an idealized ADC model was created

(in the form of a circuit-similar behavioral model with all error parameters cancelled). Next in paragraph B, typical error mechanisms were assigned to the ADC model, using a first-order behavioral description. Finally, in paragraph C, *Switched-Current ADC performance is extracted* from the "real" transistor-level circuit.

A. Intrinsic ServoLooper performance

Table 1 displays the intrinsic performance results of the ServoLooper unit, characterized by the *residual non-linearity parameters* INL_{res} *and* DNL_{res}. A comparison to the theoretical assumption is given, calculating the relative accuracy upon the formula (4).

N_{cycle}	**Simulated performance max. absolute value [LSB]**		**Calculated performance [LSB]**
	INL_{res}	DNL_{res}	
10	$< 2.3 \cdot 10^{-3}$	$< 2.7 \cdot 10^{-3}$	$2.6 \cdot 10^{-3}$
25	$3.2 \cdot 10^{-5}$	$3.4 \cdot 10^{-5}$	$5.9 \cdot 10^{-5}$
40	$1.0 \cdot 10^{-7}$	$< 1.5 \cdot 10^{-7}$	$1.4 \cdot 10^{-7}$
Conditions: *ServoLooper* C_{REG}=1 to 255, *ADC model:* $t_{recovery}$=5 µs, t_{stream}=160 µs, I_{ref}=127.66 µA, *ELDO simulation options:* abstol=1e-16, reltol=1e-08, itol=1e-08, HMAX=100ns			

Table 1 Intrinsic ServoLooper performance.

In Table 1, the $t_{recovery}$ parameter means the ADC recovery time after a conversion, t_{stream} is the length of the output ADC stream representing the serial output word, I_{ref} is the ADC reference current. From the table it is obvious that the theoretical formula makes a good estimation for the simulated values. The CPU simulation time required for the full set of 255 codes was t_{CPU}=1 hour, running on 2 GHz PC.

B. ADC model performance

This paragraph presents the simulation results of the ADC model in conjunction with the proposed Servo-Looper unit. The ADC model arrangement used for this simulation is depicted in Fig. 4. Basically, the model describes one-bit successive approximation algorithm where two basic error sources characterized by the *gain* and *offset* error, are incorporated. In Fig. 5, the simulated INL contribution from the gain error and offset error is shown, demonstrating the typical signature of a multiplicative error mechanism [6]. With respect to the intrinsic performance discussed in paragraph A, the Servo-Looper was able to correctly evaluate non-linearity contribution from a gain error level downto about 10 ppm (setting the variable N_{cycle}=25). According to the fact that a behavioral ADC model was simulated, the CPU simulation time requirements were the same as in paragraph A.

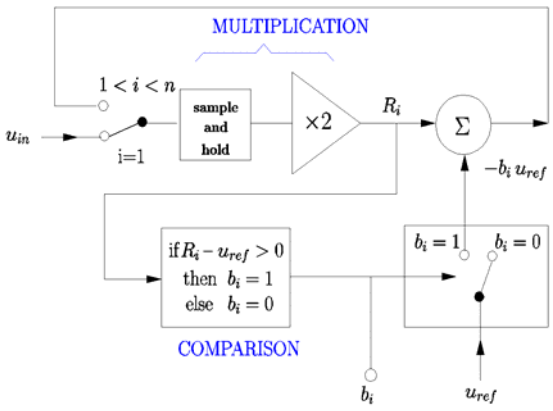

Fig. 4 Behavioral ADC model.

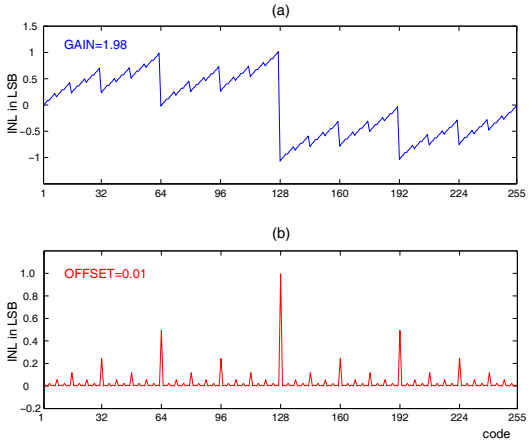

Fig. 5 INL contribution from (a)gain error, (b) offset error.

C. Transistor-level ADC performance

Table 2 displays the simulated results from the full transistor-level design of a Switched-Current ADC employing successive approximation technique [7]. The simulation setup and time requirements are also reported. It is worth mentioning that the Servo-Looper prooved a sufficient performance to extract the low-magnitude non-linearity errors, being a consequence of the robust ADC design.

Parameter	**Explanation**	**Value**
INL	Integral non-linearity – max. absolute value	0.038 LSB
DNL	Differential non-linearity – max. absolute value	0.033 LSB
t_{CODE}	Simulation time per code [1)]	5 hours
INL_{res}, DNL_{res}	Estimated residual error [2)]	$6 \cdot 10^{-5}$ LSB
Note: 1) running on 5 PCs in parallel, *ServoLooper* C_{REG} = subset of 27 codes 2) N_{cycle}=25, Temp=27 °C, typical MOS models. Other simulation conditions same as in Table 1.		

Table 2. Overview of transistor-level simulation results.

IV. Conclusion

This work presents an innovative approach to the extraction of ADC performance, suitable for both full transistor-level and behavioral simulation. The ServoLooper unit presented was written as a versatile program module for ADC performance extraction and is suitable for co-operation with any analog simulator supporting behavioral (Verilog-A) device models. In conjunction with the ELDO simulator, it also enables the multi-processor run feature. The next significant advantage of the ServoLooper module is the fact that it is capable to extract the static non-linearity of any ADC architecture, described at analog or behavioral simulation level of abstraction. The required specifications are the data validity and recovery time and the type of ADC input, either voltage or current. Here, the main motivation for creating the ServoLooper module was the fact that the choice and availability of similar software tools is problematic.

V. Acknowledgment

The work has been supported by the grant GAČR 102/07/1186 of the Grant Agency of the Czech Republic running in co-operation between the Czech Technical University in Prague and ASICentrum Prague. The research concerning data converter Design Evaluation was supported by Enterprise Ireland, Grant No. IP/2004/207. The work has been also supported by the research program MSM6840770014 of the Czech Technical University in Prague.

References

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

[2] P. D. Capofreddi, B. A. Wooley: "The Efficiency of Methods For Measuring A/D Converter Linearity", *IEEE Trans. Instrum. Meas.*, vol. 48, No. 3, pp. 763-769, June 1999.

[3] C. Wegener: *Applications of Linear Modeling to Testing and Characterizing D/A and A/D converters,* Ph. D. Thesis, University College Cork, Ireland, 2003.

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

[5] Open Verilog International, Inc.: *Verilog-A Language Reference Manual*, Version 1.0, August 1996

[6] O. Šubrt: *Application of SI Technique in A/D Conversion and Its LEMMA-aided Design Evaluation*, Ph.D. thesis at CTU Prague, Faculty of Electrical Engineering, Czech Republic, December 2005

[7] O. Šubrt, P. Drechsler: "High performance approach to algorithmic A/D converter using new types of SI memory cells", *Proc.* IFAC Workshop PDS 2003, Ostrava, Czech Republic, pp. 101-105.

[8] Mentor Graphics Corporation: *ELDO User's Manual, Software version 6.2_1*. United States, 2003