An Innovative Verification Approach For Nyquist-rate A/D Converters – Algorithm and Implementation

J. Židek [1], O. Šubrt [1,2], P. Martinek [1]
[1] Department of Circuit Theory FEE CTU Prague,
Technická 2, 166 27 Prague, Czech Republic
[2] ASICentrum,
Novodvorská 994, 142 21 Prague, Czech Republic
E-mail : zidekj1@fel.cvut.cz, Ondrej.Subrt@asicentrum.cz, martinek@fel.cvut.cz

Abstract:
Environment for testing static parameters of analog-to-digital converters is presented in this article. It is a novel concept of powerful engine suitable for design and verification of generic type ADCs in Mentor Graphics IC Studio software. The source code of each block of the design is written in Verilog-A which offers relatively effortless portability on different design systems (e.g. Cadence). The core of our proposal is based on Servo-Loop with improved search algoritm [1]. The simulation outputs are curves of static INL and DNL. Here, we focus mainly on algorithm and implementation of testing interface.

INTRODUCTION

Integral (INL) and differential (DNL) non-linearity are two of basic parameters of A/D converters. The ways of their measurement can be divided into two groups. Algorithms belonging to the so-called open-loop category are advantageous for production test. Best known member of open-loop methods is the histogram method. Procedures from the second group (referred to as closed-loop) create a reasonable compromise regarding the simulation requirements. The basic method is the standard Servo-Loop algorithm [5]; however, much shorter simulation time requirements are taken by applying the Improved Servo-Loop method [1] which meets the same performance specifications.
Recent works in this field are mostly oriented either to measurement level or behavioral model simulations employing mathematical software such as Maple or Matlab. The environment proposed in our article is built up completely in Verilog-A and therefore it can be used in a direct cooperation with analog and mixed-signal circuit simulators (e.g. Eldo, Spectre, Advance MS, etc.) up to full transistor-level complexity with no need of any other computational or post processing software.

IMPROVED SERVO-LOOP ALGORITHM

In Fig. 1, block scheme of the proposed Servo-Loop system is outlined. Our Servo-Loop implementation is based on this scheme, suggesting significant improvements against the basic approach [5]: effective usage of the discrete-time integrator, application of the initial condition and refinement of the integrator step. There are two major ways how to implement this algorithm and both of them are dedicated for different class of applications. The first approach concerning the on-chip testing environment can be realized by discrete integrator (e.g. switched capacitor one) and several logic blocks. This method is good for final chip measurement with limited possibility of structure change.
The second concept of testing environment is completely integrated into IC design software. The disadvantage of the second approach is much longer time used for the verification (based on "virtual measurement") against the first one, but the time spent for ADC design and evolution is much shorter. Implementation is also simpler, because environment can be written only at behavioral level; gate-level synthesis is therefore not required.

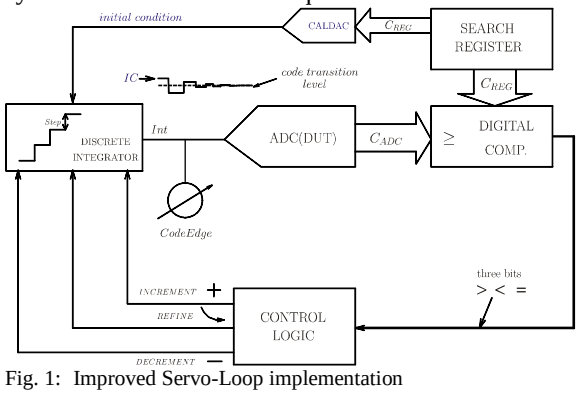


Fig. 1: Improved Servo-Loop implementation

Algorithm Principle and Definitions

The flowchart of the proposed novel algorithm variant is depicted in Fig. 2A). Here, V_{min} and V_{max} are the minimum and maximum ADC input voltages representing the full scale range. BITS is the number of ADC bits, i.e. the output word width. The LastEdge variable stores the value of the previous code transition level, see further explanation below. Finally, V_{lsb} is the code width expressed in term of the input voltage, i.e. the analog input increment

corresponding to 1LSB code change of an ideal ADC with the same analog input range as the DUT.
In Fig. 2, the initial variables are set immediately after start. The main algorithm cycle is executed for each transition level, i.e. 2^{BITS}-times for the whole set of the ADC codes. The looping statement is ensured by incrementation of C_{REG} variable representing the actual code for which the transition level has to be found. Based on the C_{REG} value, the V_{ref} is calculated and then is used for INL computation. The next step is the most important part of the algorithm formed by the transition level search procedure; it is detailed in grey box in Fig. 2B). First, the initial values of the internal variables are set and after that, single ADC conversion is performed. The discrete integrator output INT is changed in dependence on relationship

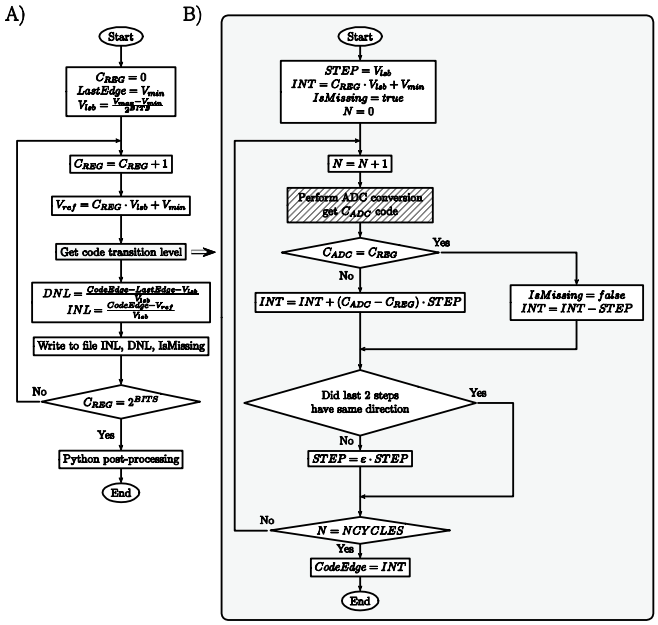


Fig. 2: Algorithm flow chart - A) Complete diagram, B) Detailed view of code transition level computation

between converter output code C_{ADC} and C_{REG} value. Here, the $C_{ADC} - C_{REG}$ term ensures a quick convergence action in case that the actual C_{ADC} is too far from C_{REG} target. At this point, it is important to note that the lower step transition level definition is applied [3]. The improvement of convergence suggested by us is as follows. The STEP size refinement by $\varepsilon < 1$ constant is not done at the end of each cycle as in [1]. Only if the last two iteration steps do not have the same direction, STEP size refinement is done. Thanks to that, $\varepsilon < 0.5$ can be used without loosing certainty of convergence.
IsMissing boolean variable indicates that the appropriate C_{ADC} code is present on the ADC transfer characteristic. The extracted code transition level value is outputted to the main algorithm cycle (Fig. 2A), the DNL and INL are then calculated. The INL and DNL data, together with the IsMissing variable are written to separate files for the next processing in Python script language. The algorithm terminates when the set of code transition levels is complete.

Python Extension for Result Post processing

Here, it is necessary to notice that the Verilog implementation in MGC software has one specific feature. The file writing subsystem adds unwanted additional lines into the output file together with the useful data. Therefore, it is impossible to format the file in compliance with the EzWave input format. That is why a script in the Python language was used. The proposed implementation can evaluate five types of INL representation (Basic, Offset compensated, Mean compensated, End-Point-Corrected, Best-straight-line). The computation of these dependencies is very simple.

$$INL_{Offset}[C] = INL[C] - INL[C_0] \qquad (1)$$

$$INL_{Mean}[C] = INL[C] - \frac{1}{N}\sum_{i=1}^{i=N} INL[i] \qquad (2)$$

$$INL_{End-Point}[C] = INL[C] - (k \cdot C + q) \qquad (3)$$

where $k = \frac{INL[C_{\max}] - INL[C_0]}{C_{\max} - C_0}$ and

$$q = INL[C_0] - C_0 \frac{INL[C_{\max}] - INL[C_0]}{C_{\max} - C_0}$$

$$INL_{Best-SL}[C] = INL[C] - (k \cdot C + q) \qquad (4)$$

where k and q are got from INL by linear least square algorithm.

Meaning of variables in equations (1) to (4) is: C is code for which is INL computed, C_0 and C_{max} are the first and last code for which is INL defined.
The Fig. 3 refers to each INL description. The Basic INL is labeled INL in previous equations and is evaluated directly in MGC in coincidence with the equation in Fig. 2A).

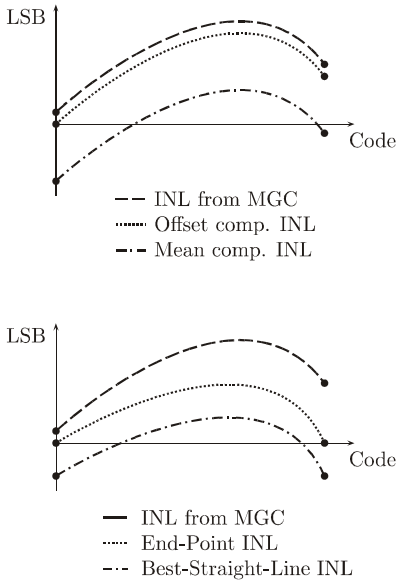
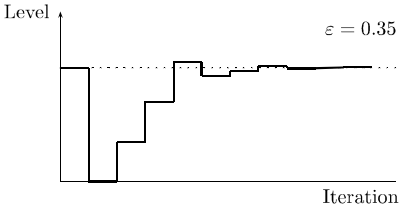


Fig. 3: All variants of INL output

System Accuracy

The algorithm resolution is one of the most important parameters. In our implementation, the algorithm can effectively change accuracy by using two parameters. The first parameter is the number of iterations NCYCLES and as it is thoroughly discussed in [5], the algorithm resolution depends also on the ε variable. The optimum value of ε to meet the convergency requierement is not unique. We verified this statement by empirical tests in Matlab. These tests showed that the best value of ε is between 0.24 and 0.35.
Influence of ε and NCYCLES is well understandable from Fig.4 and Table 1. shows maximal error of code edge measurement for ε=0.35.



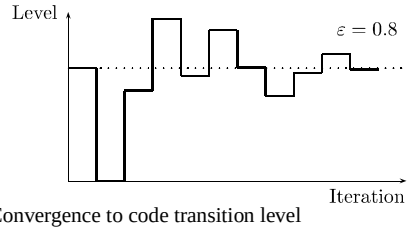


Fig. 4: Convergence to code transition level

Table. 1: Maximum error vs. number of cycles

NCYCLES	*Maximum error [LSB]*
5	0.23
10	0.043
15	0.0053
20	0.0012
25	0.00020
30	0.000025

ENVIRONMENT IMPLEMENTATION

As it was mentioned above, the algorithm was implemented in Verilog-A. Algorithm block diagram is in Fig. 5. Output word from ADC DUT of maximum size of 16 bits is connected to the block labeled as D2A. This block converts the digital signal to a form which can be easily processed by Verilog-A. The next block is the voltage controlled voltage source outputting the difference between the input value (in principle C_{ADC}) and the reference value (C_{REG}). The difference is led to the input of the Step Control block, which computes an appropriate size of the next step. The last block is the discrete integrator (DISCINT) with the built-in initial condition; the condition is loaded to the comparator output when reset is at zero level. Each block is sensitive for input signal only at the time when its clock signal has a rising edge. It is advantageous due to effective usage of simulation time. Clock signals are generated by GENCLOCK in sequence of bus indexes. The function of CONTROL matches to the A) part of Fig. 2.
As an illustration, we provide below an example of DISCINT Verilog-A code:

```
`include "disciplines.h"
module DISCINT(clk, step, init, res, out);

input init, step, clk, res;
electrical init, step, clk, res;
output out;
electrical out;

real aout;

analog begin
   if (V(res) == 0.0)
      aout = V(init); //inicialization
   @(cross(V(clk)-2.5, +1)) begin  //at clock
      if (V(res) != 0.0)
         aout = aout - V(step);
   end;

   V(out) <+ aout;
```

```
end
endmodule
```

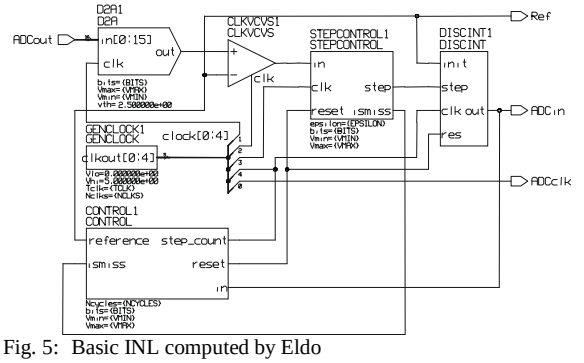


Fig. 5: Basic INL computed by Eldo

SIMPLE FLASH ADC TESTING EXAMPLE

This section presents simulation result of the Flash ADC in conjunction with the proposed Servo-Loop unit. The ADC is the basic 8-bits realization with a resistor chain and ideal comparators. Thermometer-to-Binary decoder is designed as a detector of the most significant logical ”one”. Comparators and Thermometer-to-Binary decoder are realized only as a behavioral model in Verilog-A. This approach is chosen, because of simulation time consumption result relevancy. As we focus mainly on the testing algorithm issues, we target to maximize the ratio between the verification environment and DUT contribution to the simulation time. The value of resistors in chain is 1k by default. In the following simulation set, linear superposition principle is checked for the sum of INL performance contributors versus the sum of individual input of all of them. The simulation results, when R_1, R_{32}, R_{64}, . . . ,R_{256} are changed to 1.5k, can be seen in Fig.6a. The $\varepsilon = 2/3$, NCYCLES = 19 and default Eldo parameters was chosen for these simulation.

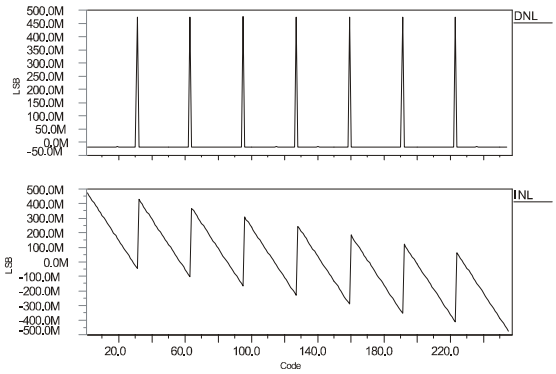


Fig. 6: DNL and INL for R1, R32, R64, …, R256 equal to 1.5kΩ

INL and DNL results from nine simulations, where only one resistor from R1, R32, R64, …, R256 set is changed to 1.5kΩ, are subtracted from the data depicted in Fig.7. The final result can be considered as a *residue error* of the algorithm under the above-mentioned parameter values and it is shown in Fig.7.

In this paragraph, we illustrate the all-code simulation time of above mentioned system (Servo-loop unit and 8-bits Flash ADC). The accuracy obtained from a cal-

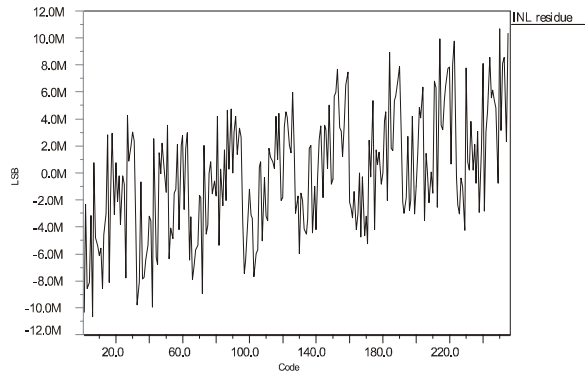


Fig. 7: INL residue (ε=2/3 and NCYCLES=19)

culation is given too. The worst-case extraction error for 10 iterations is approximately 26 mLSB; this can be furthermore reduced to only 140 nLSB after 40 iterations. This difference is highlighted by the fact than the global elapsed time varies from 40 seconds (for 10 iterations) only to 1 minute and 54 seconds (for 40 iterations). It is computed by the equation (1) with no respect to accuracy options of Eldo simulator (abstol=$1\cdot10^{-16}$, reltol=$1\cdot10^{-08}$, itol=$1\cdot10^{-08}$ and vntol=$1\cdot10^{-08}$). The simulation ran on PC with E8400@3.00GHz processor.

Fig. 6 shows the DNL simulation result as a demonstration of the resistor values deviation. Backgrounded by [7], we identified that the major component formatting the shape of the curve depicted in Fig. 6 follows the shape of Walsh-Rademacher function. Since the extrema of these functions are a priori known, this in fact can simplify the so-called *test point selection* procedure – see below in Section 5.

FURTHER DEVELOPMENT -USAGE OF LEMMA

As it is reported in [6], the so-called *LEMMA method* (Linear Error Mechanism Modeling Algorithm) can be used to streamline the simulation and measurement of ADC. This algorithm underlies the sensitivity analysis over a set of the fundamental device non-idealites referred to as *error sources*. In our flash ADC case, the error sources can be represented e.g. by the comparator offset and the deviation from nominal resistor values. If error magnitudes are small enough (meeting the requirements of linear modeling), it is possible to use principle of superposition and scaling of error sources. Nonlinearity of converter response can be decomposed to a sum of linearly independent error sources, which are formed in the so-called *ambiguity group*. This approach allows shortening the simulation time of the converter response as a matter of test points quantity reduction. Here, the test points are selected on the criterion based on the evaluation

of dominant error sources. Implementation of this advanced approach is being prepared in nowadays.

CONCLUSIONS

This work presents an innovative approach to the extraction of ADC performance, suitable for both full transistor-level and behavioral simulation. The Servo-Loop 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 multiprocessor 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. The main motivation for building the Virtual testing engine was the fact that the choice and availability of similar software tools is problematic. The next enhancement of our work aided by the LEMMA algorithm will bring simulation acceleration of converters specified by a complex model such as full transistor-level description.

ACKNOWLEDGMENTS

The work has been supported by the grant GACR 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 work has been also supported by the research program MSM6840770014 of the Czech Technical University in Prague. ICstudio®, Advance MS®, Design Architect® , Eldo® and EZwave® are registered trademarks of the Mentor Graphics Corporation. Maple® is a trademark of the Waterloo Maple, Inc., Matlab® is a trademark of the Mathworks, Inc.

REFERENCES

[1] Šubrt, O., Martinek, P., Wegener, C. *A Powerful Extension of Servo-Loop Method for Simulation-based A/D Converter Testing*, 15th IMEKO TC4 Symposium.

[2] Geelen, G. *A 6b 1.1GSamples CMOS A/D Converter*, ISSCC 2001.

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

[4] Estrada, P., Maloberti, F. *Virtual Test Bench for Design and Simulation of Data Converters*, Proc. IEEE Conf. BMAS, Behavioral Modeling and Syst., Santa Rosa, California 2002

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

[6] Wrixon, A., Kennedy, M. P., *A Rigorous Exposition of the LEMMA Method for Analog and Mixed-Signal Testing*, IEEE Transactions on Instrumentation and Measurement, October 1999, Vol. 48, No. 5.

[7] Z. Růžička, *Walsh function and ways of their generation*, [in Czech] Elektrorevue 2004/65-17.12.2004 internet magazine [online] [cit. 2009-10-29]. Available from WWW: <http://www.elektrorevue.cz/clanky/04065/index.html>.