A/D Converter Design Evaluation Engine in MGC Environment

[1]J. Židek, [1,2]O. Šubrt, [1]P. Martinek

[1]Department of Circuit Theory FEE CTU Prague, Technická 2,
166 27 Prague, Czech Republic,
[2]ASICentrum Novodvorská 994, 142 21 Prague, Czech Republic
Email: zidekj1@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 to educate students in working with modern CAD tools. 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. A part of article deals with the example of simple Flash ADC testing.*

Keywords: A/D Converter, Design Verification, CAD Tools, Educational Purpose

1. 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 in 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 methods. This method is, however, hardly applicable for the ADC circuit simulation as it takes too many simulation steps for a given accuracy.

Procedures from the second group (referred to as closed-loop) create a reasonable compromise regarding the simulation requirements; therefore, they are good candidates to be used during circuit-level ADC simulation. The basic method is the standard Servo-Loop algorithm [5], but for much shorter simulation time Improved Servo-Loop Algorithm can be used [1] meeting 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 co-operation with analog and mixed-signal circuit simulators (e.g. Eldo, Spectre, Advance MS, etc.) up to full transistor-level complexity without the need of any other computational or post-processing software.

2. 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.

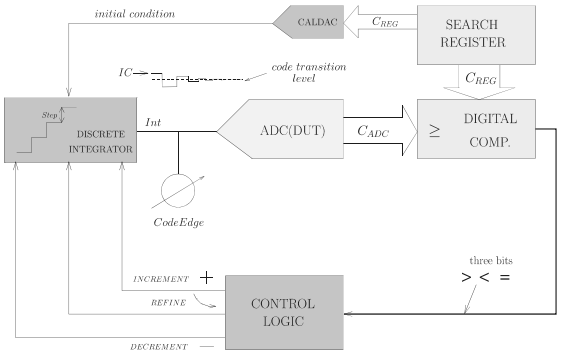

Fig. 1. Block scheme of Improved Servo-Loop

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.

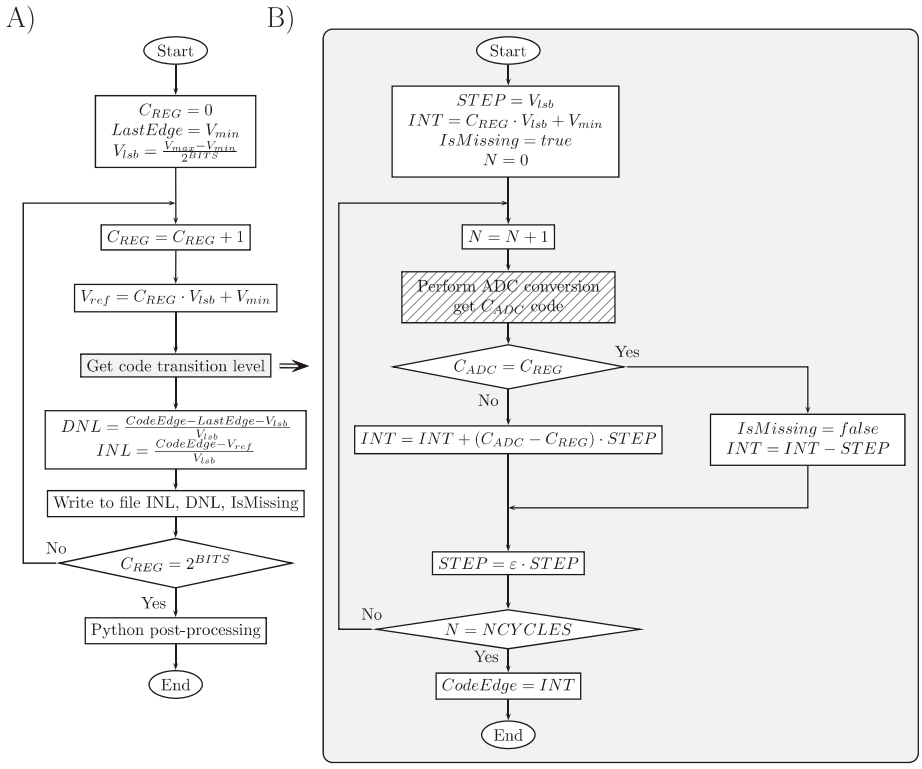

Fig. 2. Algorithm flow chart

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 CREG variable representing the actual code for which the transition level has to be found. Based on the CREG value, the V_{ref} is calculated which is then 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. If the converter output code C_{ADC} is not equal to C_{REG} value the discrete integrator output INT is changed; its incrementation or decrementation dependents on the actual C_{ADC} and C_{REG} relationship. 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. During the first loop cycle, the STEP value equals to 1LSB and it is continually refined to converge to the desired code transition level. At this point, it is important to note that the lower step transition level definition is applied [5]. The STEP size refinement by $\varepsilon<1$ constant is done at the end of each cycle. Once the C_{ADC} code equals to C_{REG} during the whole search procedure, the IsMissing boolean variable is reset; it 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 (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

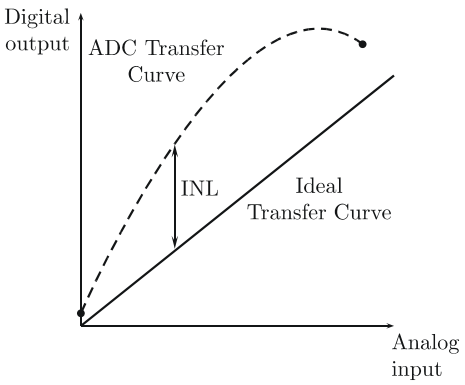

Fig. 3. Basic INL computed by Eldo

next processing in Python script language. The algorithm terminates when the set of code transition levels is complete.

Python Extension for Result Postprocessing

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, Best-straight-line, End-Point-Corrected, Offset compensated, Mean compensated).

The Fig. 4 and Fig. 5 refer to each INL description. The Basic INL curve is evaluated directly in MGC in coincidence with the equation in Fig. 2A) and with Fig. 3. The meaning of the definitions can be observed directly from Fig. 3 to Fig. 5; detailed explanation can be found e.g. in [3].

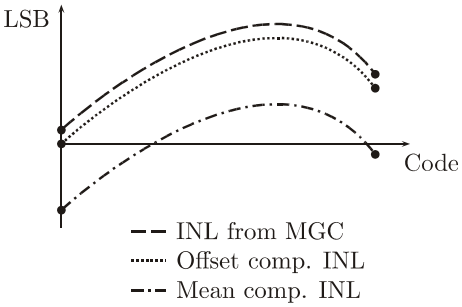

Fig. 4. Offset and Mean comp. INL

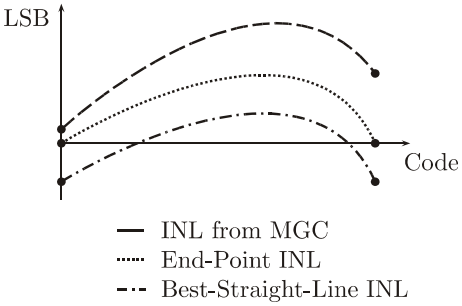

Fig. 5. Best-Straight-Line and End Point Corrected INL

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 [1], the algorithm resolution depends also on the ε variable:

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

Where Δ_N^{LSB} is the maximum possible error, ε is the refinement constant and finally, $NCYCLE$ is the number of iteration steps.

3. Environment Implementation

As it was mentioned above, the algorithm was implemented in Verilog-A. Algorithm block diagram is in Fig. 6. 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 it is C_{ADC}) and the reference value (C_{REG}). This value is led to the input of Step control block computing an appropriate size of the next step. The last block is the discrete integrator with the built-in initial condition; the condition is loaded to the comparator output when reset is at zero level. Each block works only at the time, when its clock signal is active. It is advantageous due to effective usage of simulation time. Clock signal is

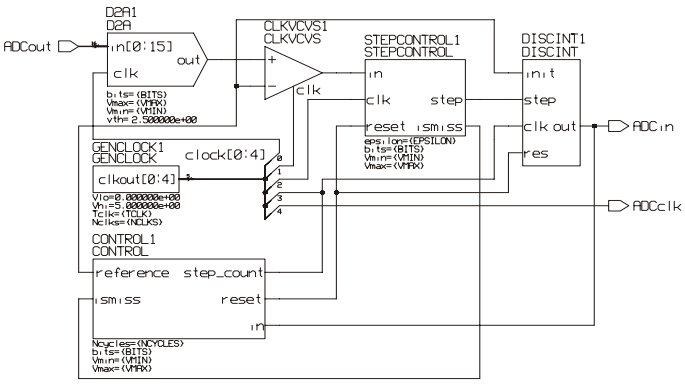

Fig. 6. MGC implementation scheme.

generated by GENCLOCK. The function of CONTROL matches to the A) part of Fig. 3.

4. 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 resistor chain and ideal comparators. Value of each resistor in chain is 1kΩ execpt R1, R32, R64, ..., R256, which have value 1.5kΩ. In Fig. 7 are displayed DNL and INL for illustration. This shows extrema case of one error source, which can be observed with our implementation.

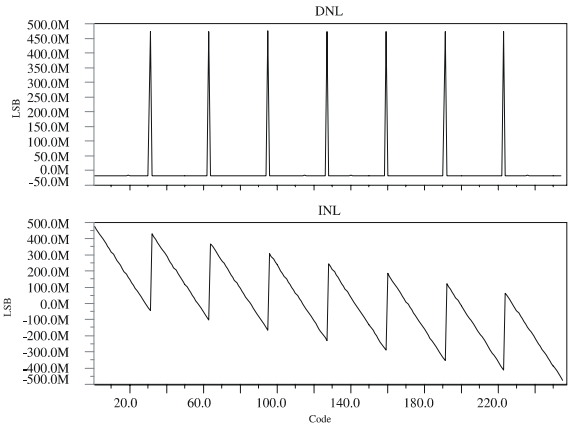

Fig. 7. INL and DNL illustration graph.

5. Conclusion

This work presents an innovative approach to the extraction of ADC performance, suitable for educational purpose. The Servo-Loop unit presented was written as a versatile program module and is suitable for co-operation with any analog simulator supporting behavioral (Verilog-A) device models. The next significant advantage of the Servo-Looper 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. This means that the netlist with extracted parametrs from layout can be used for simulation with our algorithm. This brings nearly the same results as DNL and INL measuring on real chip with no extra time for chip production.

Acknowledgements

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. In proceedings of 15th IMEKO TC4 Symposium on Novelties in Electrical Measurements and Instrumentation, 2007.

[2] Geelen G., A 6b 1.1GSamples CMOS A/D Converter, ISSCC 2001, 128-129, 438.

[3] M. Burns, G. W. Roberts, An Introduction To Mixed-Signal IC Test and Measurement, Oxford University Press, New York, 2001.

[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, 84- 89.

[5] 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.