AFFINE ARITHMETICS APPLICATIONS TO SIGNAL PROCESSING

Răzvan MIALTU[1], Mircea BODEA[2]

Affine arithmetic represents a background mathematics theory for multiple applications, including, recently, for error estimation for signal processing systems. This paper introduces affine assertions as a concept for mixed-signal circuits using an original environment for semi-symbolic simulation. The original approach is compared with system level Monte-Carlo analysis, results showing an increase factor of ca. 20 for functional coverage.

This paper summarizes the developed solution for the electronic design automation software area application of the static analysis methods in real numbers finite memory systems representation associated errors.

Keywords: affine arithmetics, SystemVerilog assertions, mixed-signal systems

1. Introduction

Integrated systems designed for conditioning and processing electrical signals have to comply with a desired behavior within specified limits. In the case of digital systems, usage of a strong methodology based on automated tools for synthesis and place and route together with high fabrication fault coverage through the use of state of the art established tests ensure failure rates under 1 ppm for the final products reaching the market. For analog circuits and for mixed-signal integrated systems, technology variation impact is amplified as local deviations of single parameters may produce non-linear interactions.

The state-of-the-art methodology for producing a robust design, capable of resulting in a high yield product on the market, is based on statistical methods for sampling the infinite space of possible values for a large set of input parameters, results being computed for this representative population and then aggregated into the solution estimation. The smallest sampling population is used in corner simulations, where technology parameters are set identically for all microelectronic devices present in the design to values that represent combinations of upper and lower side specification limits. An enlarged sampling

[1] Eng., TA, Depart. of Devices, Circuits and Electronic Apparatus, University POLITEHNICA of Bucharest, Romania, e-mail: razvanmialtu@yahoo.com

[2] Prof., Depart. of Devices, Circuits and Electronic Apparatus, University POLITEHNICA of Bucharest, Romania, e-mail: mircea.bodea@dcae.pub.ro

population is used with Monte-Carlo simulations as, in this case, each device in the system is characterized by its own set of random values for the technology parameters constrained to their own specifications. The finite computation resources determine these methods to represent a bottleneck of the design and verification flow, thus increasing the projects associated risk. Pure numerical simulations (e.g. transient runs, DC analysis, AC analysis, Monte Carlo, technology corners) are characterized by a soft limitation that impacts their overall performance when employed for verification and more specifically to system validation: data provided on specification parameters does not encompass any information about uncertainty sources that contribute to the results.

Integrated circuits hardware design and software development use an extended set of formal methods to ensure the product compliance with specifications. These validation techniques rely on mathematical identity proof and their implementation with electronic design automation software applications assumes symbolic interpretation of available data resulted from successive steps of the development flow, thus solutions complexity increases with vocabulary and syntax (e.g. use of formal techniques is more natural for digital systems than for analog components).

This paper introduces a novel simulation framework enabling an improved methodology for signal processing systems validation based on the mathematical theory of affine arithmetic. The original use of state-of-the-art SystemVerilog [1] digital assertions, originally extended to special conceived affine signal classes, enables semi-symbolic simulations, where a single simulation provides supplemental information on deviations from the nominal case.

This paper is divided into chapters as follows. Initially, the background mathematical theory of affine is introduced from a requirement driven view. Next section is focused on the presentation of the novel concept of assertions for affine represented signals. The implementation details for a specific signal processing system architecture mathematical novel models (Mathworks Simulink and SystemVerilog) are part of Section 4, while the final two sections include results of simulations focusing on comparison between the two methods to characterize system parameters variation for providing the novel validation methodology an initial proof of concept.

2. Affine Arithmetic Theory

Designed systems performance is limited by technology, concept and deviation from mathematical abstraction. By technology limitation, the convention herein is to understand the state-of-the art physics associated with a certain system class, e.g. integrated electronics systems functionality is based on silicon semiconductor phenomena. Concept limitation refers to human civilization understanding of problems and general solutions to them, e.g. computer architectures have in common processing core accessing via buses data found in memory components.

Mixed-signal systems due to their nature of processing values that are time-continuous and value-continuous have intrinsic errors both in analog and in digital domain. Starting from imperfect modeling of circuit behavior in simulators due to incomplete understanding of phenomena or due to equation simplifications for computational resources reasons, or from error injection in simulation due to computation with a software on a digital machine storing information in limited size registers and ending with silicon transistors parameter variations, signals in the final product differ from their expected values. A robust design is achieved when the differences do not accumulate, thus keeping real values within certain limits from the expected ones.

Range arithmetic was introduced [2] to provide a complete mathematical formalism for the study of errors propagation in a digital computing machine, i.e. a computer processor, having as motivation the necessity of providing accurate computation results, with a known documented degree of accuracy. Interval arithmetic set is composed of a superset of real numbers, each of its elements representing the whole set of real values found lying between its two endpoints.

Recursive mathematic functions or feedback control loops show the insufficiency of interval expressions, where they fail to converge.

Numbers in affine format [3] represent values determined by a set of variables, thus being similar with a multi-variable function $f(x_1, x_2..x_d)$. Taylor's theorem stipulates that a one can find a series $T(x_1, x_2..x_d)$ approximating the function within a defined accuracy, the remainder being given by the same theorem:

$$T(x_1,..,x_d) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_d=0}^{\infty} \frac{(x_1-a_1)^{n_1}..(x_1-a_1)^{n_1}}{n_1!..n_d!} \left(\frac{\partial^{n_1+..+n_d} f}{\partial {x_1}^{n_1}..\partial {x_d}^{n_d}} \right) (a_1,..,a_d) \tag{1}$$

equivalent to

$$T = f(x_L) + \frac{\partial f}{\partial x}(x - x_L) = f(x_L) + \frac{f(x_H) - f(x_L)}{(x_H - x_L)}(x - x_L) \tag{2}$$

or written as

$$T = \sum_{k=1}^{d} \left[f(x_L^k) + \frac{f(x_H^k) - f(x_L^k)}{(x_H^k - x_L^k)}(x - x_L^k) \right] \tag{3}$$

and expressed as a sum of

$$T=\sum_{k=1}^{d}\left[f(x_L^k)\right]+\sum_{k=1}^{d}\left[(f(x_H^k)-f(x_L^k))\frac{(x-x_L^k)}{(x_H^k-x_L^k)}\right] \tag{4}$$

where $f(x_L^k)$ are the values in whose vicinity the function is computed.

From a mathematical point of view, numbers of affine arithmetic are defined as

$$\tilde{x}=x_0+x_1\varepsilon_1+x_2\varepsilon_2+\cdots+x_n\varepsilon_n \tag{5}$$

with the number $\tilde{x}$ uniquely defined by a set $x_0, x_1, x_2..x_n$ of known floating-point numbers and by a set $\varepsilon_1, \varepsilon_2..\varepsilon_n$ of symbolic variables whose value respect

$$-1<\varepsilon_i<1 \tag{6}$$

where ε_i is called noise symbol, thus the influence of each noise symbol being quantized by its derivative value $x_1, x_2..x_n$.

Considering an ideal value defined by no influence of noise components, i.e. $\forall i \in N^* x_i = 0$, the projections of the affine variable on $\mathbb{R}$ of maximum and, respectively, of minimum value, shall be found at same distance to the ideal value. The total deviation for an affine variable from its central value has the metric defined by

$$rad(\tilde{x})=\sum_{k=1}^{n}|x_k| \tag{7}$$

corresponding to the difference to the worst cases values, i.e. all the terms contributing in a singular error direction (either positive or negative).

Two affine forms may share a set of common noise symbols $\{\varepsilon_{i1}, \varepsilon_{i2}..\varepsilon_{im}\}$ thus indicating a partial dependency or correlation between the two quantities. A pair of intervals (x, y) in $\mathbb{R}^2$ space constraints possible values to a rectangular region denoted by $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$, whereas in the case of an affine forms, pairs sharing at least one symbol determine $\mathbb{R}^2$ definition domain to be constrained to a smaller polygon. As a generalization, m affine forms depending on n noise symbols determine a set S of possible joint values for the corresponding quantities in the form of a center-symmetric convex polytope that is a parallel projection into $\mathbb{R}^m$ of the hypercube U^n.

The group law of affine forms, the addition, manifesting closure, associativity, existence of an identity element and generation of an inverse element, is linearly defined according to representation of any affine form as a sum of noise symbols and derivatives products. Therefore, linear operations results are part of affine forms group as equation:

$$\tilde{z}=\alpha\tilde{x}+\beta\tilde{y}+c \tag{8}$$

with $c \in \mathbb{R}$, is easily converted to

$$\tilde{z} = \alpha x_0 + \beta y_0 + c + \sum_{k=1}^{d} (\alpha x_k + \beta y_k) \varepsilon_k \tag{9}$$

where one can recognize an affine form with components univocally determined by the summing terms. With the observation that real numbers are a subset of affine numbers, addition and any operation that generates an affine number result is said to be an affine operation. Non-affine operations can generate non-affine numbers when their terms are affine themselves, e.g. the multiplication represented by

$$\tilde{z} = \tilde{x}\tilde{y} \tag{10}$$

and defined by summation of products resulting from all symbols combinations. While multiplication with the ideal value, a real number in itself, generate affine form terms, multiplication of products that include symbols shall generate quadratic symbols, not included in affine numbers set.

Consistency of affine arithmetic, i.e. closure, when non-affine operations are employed, is ensured through results approximation with a new affine number. For the multiplication result, the last term of equation

$$\tilde{z} = x_0 y_0 + \sum_{k=1}^{n} (y_0 x_k + x_0 y_k) \varepsilon_k + z_{n+1} \varepsilon_{n+1} \tag{11}$$

is replacing the quadratic components from the exact formula with a new noise symbol and a new corresponding weight. The approximation procedure ensures closure with the drawback of inclusion of an additional noise term and an intrinsic overestimation.

3. Mixed-Signal Assertions

Current signal processing integrating systems complexity leads to increased effort associated to design and verification activities. The solution found for decreasing the time to market of a product containing at least one integrated digital system is usage of automation procedures. These are not restricted to extensive scripting and other methods used for software testing, specific methodology making use of hardware verification language special structures generating constrained randomization of input stimuli and providing flexible checking with employment of assertions.

Randomization of stimuli for a digital system is used for creating sets of values for all inputs ports in a way that statistically will cause the internal finite state machines to transition between most of their states, desirable through all, in various sequences. Introduction of constraints into the randomization equations promises that only valid stimuli are provided as inputs, thus reducing the number of false errors and optimizing resources. At the other end, errors are discovered automatically through the use of assertions, as long as a specification is in place. This specification dictates the way outputs shall behave when certain signal values are provided at the system inputs. Insertion of assertions is not limited to system output ports, as grey box testing provides a superior confidence level.

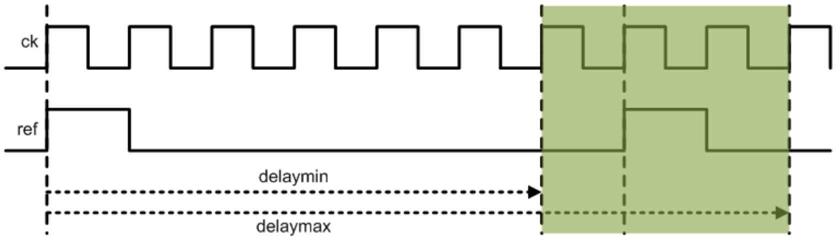


Fig. 1. Basic digital assertion for events succesion.

Digital assertions, currently state-of-the-art, are limited to a certain set of expressions that are relevant for digital systems (e.g. check of time intervals between two active states, as shown in Fig. 1), representing in fact a subset of expressions that SystemVerilog language is able to provide.

Last years have known the spawning of activities for integrating assertions, both Property Specification Language [4] and SystemVerilog [5] flavored, into mixed-signal systems pre-silicon verification, extending a digital system design state-of-the-art methodology to analog domain. The efforts have been not only directed to providing checkers for analog models developed in the extended family of hardware description languages (VHDL, Verilog and their extensions), but also for integrating assertions directly at SPICE netlist level [6] with the help of *pragmas*.

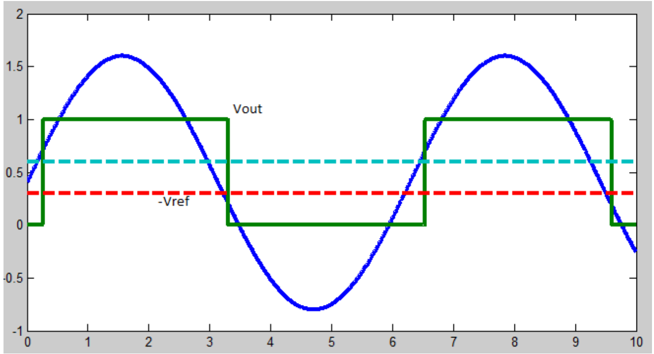


Fig. 2. Typical assertion on the analog signal of a comparator with hysteresis.

Analog assertions [5], due to signals continuous nature both in time and in value, represent a more complex version of the digital counterparts, with an extended set of usual operators. A PSL assertion for a simple comparator with hysteresis, with its behavior shown in Fig. 2, shall be written in the form *hyst_comp_pos: assert always “V(in) > V(ref)” -> “V(out) >= V(Voref)”;* and *hyst_comp_neg: assert always “V(in) < -V(ref)” -> “V(out) <= -V(Voref)”;* i.e. based on a hysteresis of a value double than that of *V(ref)*, the comparator output voltage shall be set to the voltage corresponding to correct logic[7].

As the novel environment implements an arithmetic that extends the mathematic operators associated to real numbers, associated classes are included to represent stencils both for data and for operators. Classes are necessary as simple type extension provide neither basic operators functionality nor necessary memory automatic management.

Affine arithmetic is introduced for mixed-signal systems study as a method to analyze the parameters variation on overall performance and whether a product sample under the effect of technology and environment condition corners shall still exhibit functionality within specification limits. The benefit that affine arithmetic promises in respect to classical Monte-Carlo and corner simulations approaches is an optimized result-to-resources ratio, but one can profit from this advantage only if simulation data processing is automated as well.

As explained previously, full automation for integrated electronic systems verification is implemented through two techniques: (1) constrained randomization and (2) assertions. These ensure that for a range of inputs valid (properly constrained) and representative (properly randomized) the system-under-test shall exhibit a functionality a-priori documented (characteristic determined by output signals being asserted accordingly).

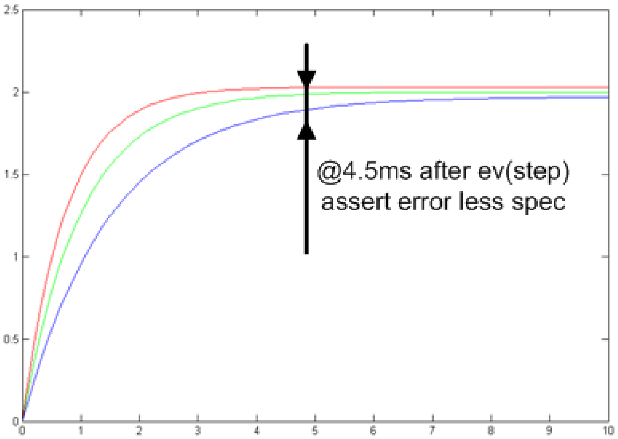


Fig. 3. Typical affine signal assertion on a noise symbol.

The fundamental affine assertion is represented by a simple checker of a signal defined affine term within a certain time interval, as shown in Fig. 3. The formal expression can be explained as the verification of the measure of the impact a variation of a signal path precedent block parameter has at an ulterior location of that signal path. Formally, the checker equivalent in algebraic analysis is the determination of the variation of the range from the co-domain of a composition of functions when certain functions parameters may have any values within specified ranges.

Considering

$$y = f(x_L, t) + \sum_{k=1}^{n} f_k(x_L)\, \varepsilon_k \tag{12}$$

as the result of input

$$x = x_0 + \sum_{k=1}^{m} x_k \varepsilon_k \tag{13}$$

signal processing, with standard notations for affine terms explained above, where the signal processing is expressed by

$$y = f(x, t) \tag{14}$$

with *t* a real value, representing time, one can express the inclusion assertion as:

$$|f_k(x_L)| < lim_k\ for\ \forall t \in [t_1, t_2] \tag{15}$$

Next section shows the implementation of specific system verification language environment this formal affine assertion is used in.

4. Models for the System under Test

Two mathematical models were developed for a mixed-signal integrated system representing a silicon die comprising of an area sensitive to an external electro-magnetic field, i.e. sensitive element, and of an associated signal processing circuitry. The integrated system function is high-accuracy detection of position and speed variations for a wheel in rotation generating a periodic signal. The principle applies both for external electric fields and for external magnetic fields, as the sensitive area can be manufactured as integrated capacitive silicon structures on the circuitry die or as magnetic sensitive areas.

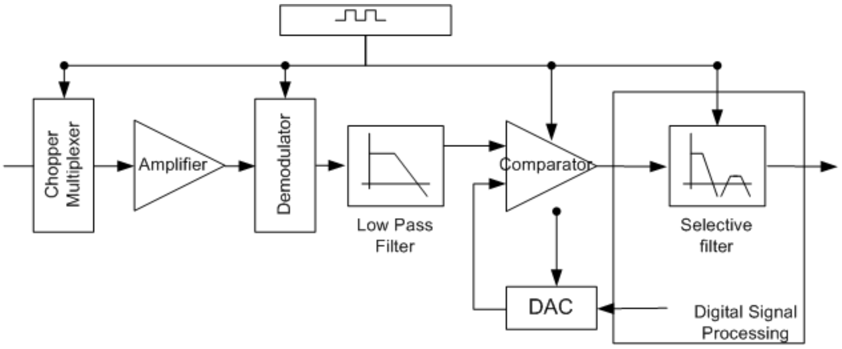


Fig. 4. System-unde-test block diagram.

As seen in the block diagram of Fig. 4, the analog sub-system is rejecting 1/f noise through the means of a modulation / demodulation scheme characteristic to chopper architecture. An up/down counting analog-to-digital converter is employed for converting the demodulated signal after an anti-aliasing and ripple rejection analog low pass filter. Further signal processing, algorithmic in nature, is implemented in the digital domain.

SystemVerilog novel implementation of the enumerated concepts, initially presented in [8], may be divided into three categories: (a) implementation of an environment with associated structures for data storing and data processing, representing the foundation layer; (b) a system model based on description of individual blocks connected in a data flow and (c) a test-bench structure that provides input stimuli for the device-under-test and includes the checking mechanisms for ensuring functionality according to specification.

According to semi-symbolic simulation methodology, affine data requires intrinsic integration with the model fabric. Accordingly, a library was developed following two directives: (1) object oriented design shall be used for all elements, thus signals, wires and blocks being all instances of specific classes; (2) affine forms are instantiations of the fundamental data class; and (3) communication is realized similar to transaction level model abstraction.

In addition to the model developed for capturing device-under-test behavior, a standard pre-silicon verification testbench was created having the particularities described by the next paragraphs.

The model of state-of-the-art digital assertions is used for developing affine specific checkers. Considering affine assertions as introduced in the previous section, an extension of standard assertion formalism is possible within the limits of SystemVerilog Language Reference Manual. While no restriction is imposed on assertions available for binary content in a mixed-signal design, coding for affine assertions is extended from digital methodology.

As such, an immediate assertion can be extended to impose a limit for certain error components from a signal (e.g. the amplifier gain variation induced error, seen after the filter, shall not extend over a specified percent of least significant bit typical voltage value at the analog-to-digital converter comparator input pins). The assertion is written as *assert ((comp.in1[gainindex]) <0.7) $display("A023 Ok. Gain delta.")* with 0.7 being 20% of the LSB typical value in mV. A comprehensive error budget can be generated and asserted in this way at any intermediate point of the signal chain, including at the output port.

Concurrent assertions rely on dynamic signals time dimension and on relations between signals from this perspective, with an assertion written as *(Offset>0)&&$rose(chopclk'event)|=>(Offset<0).* Other assertions are inserted as properties to validate and to verify the robustness of proposed design and the existence of guard bands between specification limits and the parameters of worst case samples.

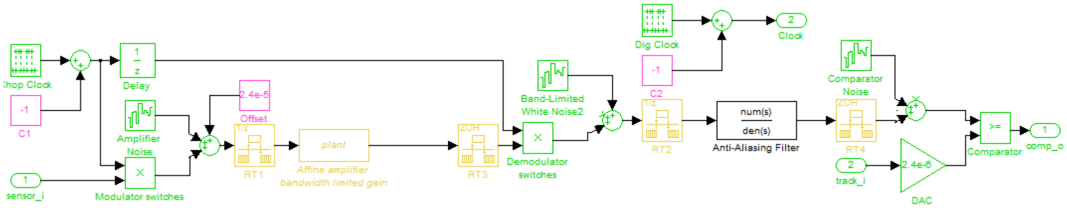
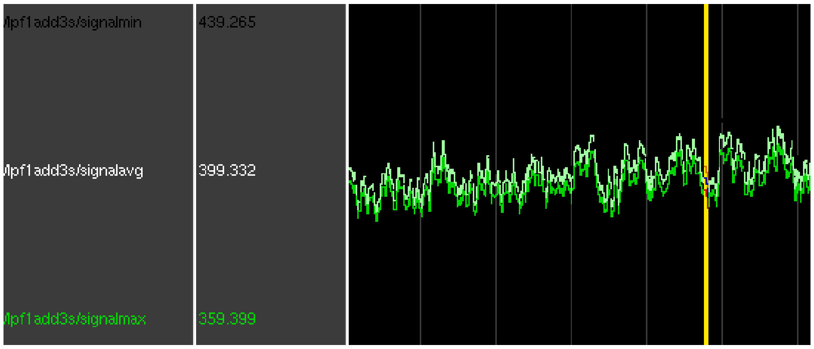


Fig. 5. Analog signal path Simulink reference model

As reference, a Mathworks Simulink model, shown in Fig. 5, was developed according to the state-of-the-art techniques using components that are found in the basic library and the Robust Control Toolbox employing variation in uncertainty form with the usage of Uncertain State Space blocks.

5. Simulation Benchmarking

The library of SystemVerilog models for components and for the underlying signal mechanism has been simulated with Mentor Graphics Modelsim v. 5.7 on Fedora Linux workstation, while results with the reference model have been obtained in Mathworks Matlab R2012a on Windows 7 workstation.



/lpf1add3s/signalmin 439.265

/lpf1add3s/signalavg 399.332

/lpf1add3s/signalmax 359.399

Fig. 6. Simulation results on the reference SystemVerilog model

Considering a 10 kHz input signal, with chopper clock at 250 kHz, 5 signal periods are simulated in 4 minutes with affine environment (Fig. 6), compared to 84 minutes for a Monte-Carlo regression with 6 runs (Fig. 7).

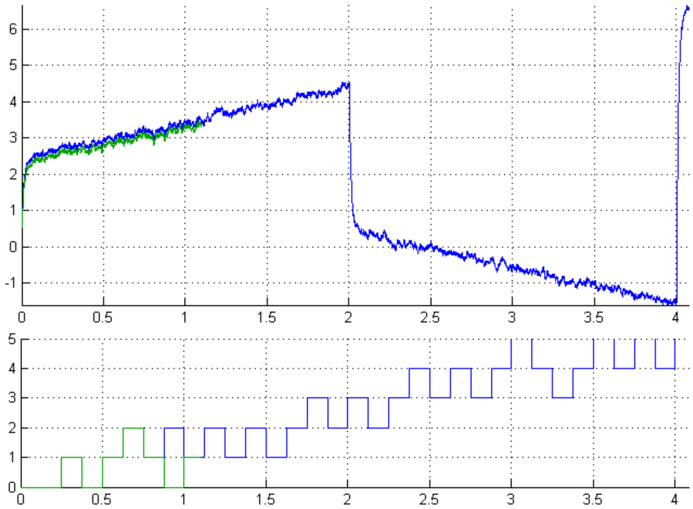


Fig. 7. Simulation results on the Simulink reference model

Benchmarking has been realized considering the scope of achieving a similar coverage factor at the assertion level, thus for the standard Monte-Carlo analysis method multiple runs are necessary, while a single run has been employed for each test case for the affine arithmetic model. Simulink simulation time is computed considering these multiple runs comprised in a script. Additionally, to minimize the influence of the tools as much as possible, batch scripting with no user interface, including no graphical plotting of the results, has been considered. While this benchmarking measure has been presented herein, a better definition of test conditions and additional effort for exclusion of external influences is necessary in the future, as at this stage the SystemVerilog model and SystemVerilog environment is at the prototype level.

6. Conclusions

Using a semi-symbolic methodology for system validation enables, as shown in this paper, the replacement of standard scripts containing a set of simulations characterized by parameters with randomized values. The major benefit is a faster concept validation (up to 20 times smaller times) and thus enabling parameter optimization at system level with direct feedback regarding overall behavior and regarding individual tolerances. Additionally to improvements in terms of required simulation time, the methodology relies on automation, i.e. inherent processing supplemental data regarding deviations, the herein introduction of affine assertions further enhancing the concept of goals validation (e.g. performance, safety related, reliability).

REFERENCES

[1] *IEEE*, "1800-2009 IEEE Standard for SystemVerilog – Unified Hardware Design, Specification and Verification Language," 2009

[2] *R.E. Moore*, "Automatic Error Analysis in Digital Computation", Technical Report LMSD-48421 Missiles and Space Division, Lockheed Aircraft Corporation,pp. 43-56, Jan. 1959

[3] *J.Stolfi, L.H. deFiguerido,* "Self-validated numerical methods and applications", 21st Brazilian Mathematics Colloquim, 1997

[4] *C.Radojicic, C. Grimm F. Schupfer, M. Rathmair,* "Verification of Mixed-Signal Systems with Affine Arithmetic Assertions", VLSI Design, vol. 2013, ID 239064, 2013

[5] *D.Nickovic, O.Maler, K. Jones, V. Konrad,* "Examples of Analog Assertions", VLSI Design, vol. 2013, ID 239064, 2013

[6] *D. O'Riordan et al.*, "PSL/SVA Assertions in SPICE", DVCON 2011, Mar 2011

[7] *D.Nickovic,* "Checking Timed and Hybrid Properties: Theory and Applications", University Joseph Fourier – Grenoble, Doctoral Thesis, Oct 2008

[8] *R.Mialtu*, "A SystemVerilog Approach in System Validation with Affine Arithmetic", in International Semiconductor Conference (CAS) 2012, vol. 2, pp. 407-410, 2012