IEEE
AUSTRIA SECTION

TU WIEN
TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Institute of Electrodynamics, Microwave
and Circuit Engineering

EMCE

TAGUNGSBAND

Austrochip

Workshop on Microelectronics

2011

austriamicrosystems

infineon

NXP

INHALTSVERZEICHNIS

System Level Robust Communication System Design using Extended SystemC AMS Building Block Library

Jiong Ou, Florian Schupfer, Christoph Grimm
Institute of Computer Technology
Vienna University of Technology
Vienna, Austria
email: {ou, schupfer, grimm}@ict.tuwien.ac.at

***Abstract*—Today, communication systems typically consist of analog hardware, digital hardware and software which are functionally interwoven. Analog circuits usually have limited accuracy and are quite sensitive against harsh environment. To guarantee the robustness of such systems becomes challenging design task. The consideration for the robustness has to be brought to the design at the beginning of the design process. On the other side, simulation and modeling of radio frequency communication systems usually tends to be slow in standard simulation environments due to the high frequencies. The module implementation is furthermore a time and effort costing task.**

In this paper, a SystemC AMS model-based design approach is combined with Affine Arithmetic to solve these two problems. A modeling example is shown in the latter part of this paper to illustrate the concept of the proposed design approach.

***Keywords*-SystemC AMS, affine arithmetic, modeling, simulation**

I. INTRODUCTION

Nowadays, communication systems are usually embedded mixed signal systems(E-AMS), which have digital HW/SW functionally interweaves with analog and mixed-signal blocks such as RF interfaces, power electronics, sensors and actuators. Unfortunately, analog circuits have only a limited accuracy and are quite sensitive to harsh environment. The increasing integration makes HW/SW systems much cheaper, and an established strategy is to use HW/SW systems to compensate low accuracy of analog circuits. Therefore, an efficient modeling and analysis approach has to be developed, so that the relationship between uncertainties and behavior of systems can be figured out at early design stage. Based on the found relationships, the compensation can be developed to increase the robustness of the whole system. In this paper, an Affine Arithmetic based semi-symbolic method is used to model the uncertainties of the E-AMS systems. Sets of uncertain values are modeled by sets of ranges enclosing all of the deviations and additionally labeling them with symbols for keeping track of the range correlations throughout the computation process.

On the other side, simulation of signal processing systems and communication systems combined with AMS functions usually tends to be slow in standard simulation environments as the high frequencies lead to very small time steps and therefore huge calculations. Furthermore, the modeling itself needs a serious investment in time, as the system designer has to create most parts of his system from scratch. The SystemC-AMS extensions offer modeling in Timed Data Flow (TDF) which allows faster simulations as the scheduling is done in advance. Besides, a pre-developed building block library is provided to relieving the designer from their detailed module implementation and giving him more time to address and analysis the system-level aspects in his design. The library is generic enough to cover a significant number of E-AMS communication system building blocks and therefore can be used for a various applications.

The paper is organized as follows: Section II presents a review of the relevant literature. Section III gives a brief introduction to SystemC AMS and Affine Arithmetic. Section IV provides an introduction to the pre-implemented building block library. Section V proposes a building block library based approach for designing robust communication systems. Section VI depicts an illustrative example to show the usage and efficiency of the proposed design approach. Section VII concludes the paper and identifies future work.

II. STATE OF THE ART

Recently several semi-symbolic system analysis techniques based on Affine Arithmetic have emerged in the academic field [6], [3]. They use ranges, defined by the Affine Arithmetic, to model and simulate parameter uncertainties in conservative and non-conservative systems. One special application is to estimate the optimal bit widths of a system to suffice a given system quality. Errors introduced during quantization, rounding and truncation operations are modeled by ranges and resulting system quantities are calculated [1]. Although the most published work concentrates on the system level also transistor level circuits are simulateable when solving the non-linear differential equations by using Affine Arithmetic [3]. [5] uses semi-symbolic simulation to analyze the convergence behavior of control loops in presence of uncertainties. [2] additionally enhances the semi-symbolic simulation for simulating non-linear analog circuits and obtaining refinement information to improve the system quality. The problem of over approximation is addressed in recent works where the Affine Arithmetic is modified to also provide exact results for multiplication operations which avoids the additional approximation terms [9].

III. SYSTEMC AMS AND AFFINE ARITHMETIC

SystemC AMS uses C++ based language constructs to model and simulate analog and/or mixed-signal systems [10]. Its main *Model of Computation* (MoC) is *Timed Synchronous Dataflow* (TDF) which is also used for the semi-symbolic simulation. It is a timed version of the original *Synchronous Dataflow* (SDF) which allows to precalculate the schedule of process executions. This characteristic offers a high simulation performance in combination with a powerful modeling expressiveness. On the other hand the C++ based nature of SystemC AMS allows easy integration of additional libraries, like the Affine Arithmetic library in our case. This extensibility makes SystemC AMS an efficient choice for the ever increasing functionality of the range based approach.

Affine Arithmetic is a semi-symbolic technique describing ranges by a nominal central value and a superimposed sum of interval valued partial deviations [7]. Affine Arithmetic bases on the original concept of Interval Arithmetic [8] but enhances it with symbolic range identifiers to overcome the dependency problem preventing the usability of the Interval Arithmetic. This correlation enhancement provides Affine Arithmetic the flexibility to be applied in realistic systems. A feedback loop, for instance, is simulateable considering the correlated summation and is not ending up in a considerable overapproximation the Interval Arithmetic would result [4]. For modeling purposes, each affine expression represents the influence of independent sources of uncertainty by a sum of partial deviations $x_i\epsilon_i$. The symbol ϵ_i represents the range $[-1, 1]$ which is scaled by the deviation value x_i. Affine expressions are referred to by $\tilde{}$ in the following.

$$\tilde{x} = x_0 + \sum_{i=1}^{length} x_i\epsilon_i \qquad \epsilon_i \in [-1, 1] \tag{1}$$

$$\tilde{x} \pm \tilde{y} = (x_0 \pm y_0) + \sum_{i=1}^{length} (x_i \pm y_i)\epsilon_i \tag{2}$$

$$c\tilde{x} = cx_0 + \sum_{i=1}^{length} cx_i\epsilon_i \tag{3}$$

Equation 1 gives the composition of an Affine Arithmetic symbol which is also referred to as Affine Arithmetic Form. The number i of partial deviations correlates with the sources of uncertainty which affect this particular quantity. Equation 2 and 3, specify the so called affine operations which result in exact solutions. All other operations can solely be solved by approximating the exact result, which is performed by computing the approximation solution and adding an additional $x_{i+1}\epsilon_{i+1}$ to enclose the remaining residual.

IV. THE BUILDING BLOCK LIBRARY

The implemented building block library, which is compliant to the recently released OSCI SystemC AMS 1.0 standard, focus on communication and radio frequency systems, especially on signal sources, basic analog arithmetic components, AD/DA converters, filters and channels. The library is available under open source license from the Vienna University of Technology, and can be soon downloaded for free from www.systemc-ams.org. The most important building blocks available in the library are introduced in this section.

A. Signal Sources

Oscillator is one of the most common and important signal sources of communication systems. The building block library provides a sine signal generator, where the phase jitter and amplitude jitter are modeled by using Affine Arithmetic. Let $y(t)$ be the output of the module. It is described by the following function:

$$y(t) = \tilde{A}sin(\omega t + \tilde{\varphi}) \tag{4}$$

while $\tilde{A}$ and $\tilde{\varphi}$ are Affine Arithmetic symbols, which define the nominal value and deviation of the amplitude and phase jitter, respectively. As sine operator is not defined for Affine Arithmetic, it is not allowed to put any Affine Arithmetic symbol, like $\tilde{\varphi}$, directly into the bracket of the sine operator. Therefore, an approximation has to be applied when calculating $y(t)$ from Equation 4. When assuming that the phase noise is sufficient small, Equation 4 can be approximated as follows:

$$y(t) \approx \tilde{A}(sin\omega t + \tilde{\varphi}cos\omega t). \tag{5}$$

Besides the sine wave generator, various signal generators are provided by the building block library.

B. Basic Analog Arithmetic Components

Analog arithmetic components like adder, multiplier, integrator and etc. are common frond-end components of communication systems. Therefore, analog arithmetic modules, which are able to deal with Affine Arithmetic signals, are provided by the building block library.

The operator integration is originally not defined for Affine Arithmetic. It is calculated in the library by applying the following numeric approximation:

Let $x(t)$ and $y(t)$ be the input and output of the integrator module,

$$y(t) \approx x(t) * \Delta t + y(t-1) \tag{6}$$

while Δt represents the simulation time step.

C. Filters

Filters are inevitable components of communication systems. SystemC AMS provides a Laplace Transfer Function (LTF) module for modeling filters generally. However, the module is not applicable to Affine Arithmetic signals. Therefore, a new Laplace Transfer Function module has to be implemented. We propose to apply the following numeric approximation for LTF module implementation. Principally, the implementation is based on the finite difference approximations method of numerical differentiation (Equation 7).

$$y'(t) = \lim_{h \to 0} \frac{y(t) - y(t-h)}{h} \approx \frac{y(t) - y(t-h)}{h} \tag{7}$$

h in the equation corresponds to the time step of the simulation.

It is assumed, that Laplace Transfer Function has the following general form:

$$H(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + ... + b_0}{a_n s^n + a_{n-1} s^{n-1} + ... + a_0} \tag{8}$$

Let $x(t)$ and $y(t)$ be the input and output of the module, Equation 8 is equivalent to the Equation 9 in the time domain.

$$\sum_{i=0}^{n} b_i x^{(i)}(t) = \sum_{i=0}^{n} a_i y^{(i)(t)} \tag{9}$$

When applying Equation 7 to Equation 9, the output of the LTF module $y(t)$ can be approximately calculated by following the Equation 10.

$$y(t) = [(-1)^i \sum_{i=0}^{n} \sum_{j=0}^{n} k_{j(n-i)} b_j h^{(n-j)} x(t-i) + (-1)^{(i+1)} \sum_{i=1}^{n} \sum_{j=0}^{n} k_{j(n-i)} b_j h^{(n-j)} y(t-i)] / (\sum_{j=0}^{n} k_{jn} b_j h^{(n-j)}) \tag{10}$$

The coefficients $k_{j(n-i)}$ is calculated as follow: $(k_{0n}, k_{0(n-1)}, ..., k_{00})$ equals (1,0,...,0). $(k_{in}, k_{i(n-1)}, ..., k_{i0})$ is calculated by using an assistant vector m, which has the element value: $m_n = 0, (m_{n-1}, ..., m_0) = (k_{(i-1)(n-1)}, ..., k_{(i-1)0})$. k_{il} is calculated by adding corresponding elements of k and m: $k_{(i-1)l} + m_l$, while $i \in [0, n]$.

D. AD/DA Converter

Analog to Digital(AD) and Digital to Analog(DA) converters build interfaces between digital and analog subsystems. Therefore, the design of appropriate AD/DA converters is an especially important issue when developing communication systems. An AD converter module and a DA converter module are provided by the building block library. The effect of dynamic reference voltage variation is modeled by using Affine Arithmetic variables.

1) DA converter: The variation of reference voltage has a linear effect on the output of the DA converter. Let the reference voltage be represented by an Affine Arithmetic variable $\tilde{V_{ref}} = V_{ref\_nom} + (-V_{ref\_dev}, +V_{ref\_dev})$, while V_{ref_nom} and V_{ref_dev} represent the center value and the deviation range of the reference voltage, respectively. The output of the DA converter can be represented by the Equation 11.

$$Output = Input_{(10)} \frac{\tilde{V_{ref}}}{2^N} \tag{11}$$

while $Input_{(10)}$ and N are the decimal representation and the bit width of the input binary signal.

According to Equation 3, the deviation of the output signal is $(-\frac{Input_{(10)}}{2^N} V_{dev}, +\frac{Input_{(10)}}{2^N} V_{dev})$.

2) AD converter: The variation of the reference voltage also has a direct effect on the resolution of an AD converter. Basically, an AD converter with N bits wide output signal has a resolution $RES = V_{ref}/2^N$, which is the smallest increment that this converter can resolve. In other words, all input signals which have a value in the range $[i{*}RES, (i{+}1){*}RES)$ will be converted to the digital word $i_{(2)}$, while $i_{(2)}$ is the binary representation of the integer i. When considering variation of reference voltage, the range has to be modified to $[i * (RES + V_{ref\_dev}/2^N), (i+1) * (RES - (V_{ref\_dev}/2^N)))$. This means, that only signals which have a value that lays in the modified range can be converted correctly. All other signals may result in error outputs due to the variation of the reference voltage.

E. Channels

In order to model the communication environment, the building block library provides two basic channel models. These are the attenuation channel module(ATTN) and the Additive White Gaussian Noise channel module(AWGN). Suppose that the input and output signal of the channel module are $\tilde{x}(t)$ and $\tilde{y}(t)$, respectively. The channel module behavior can be represented by the following equations:

$$\tilde{y}(t)_{ATTN} = \tilde{G} * \tilde{x}(t) \tag{12}$$

$$\tilde{y}(t)_{AWGN} = Gaussian\ Noise + \tilde{x}(t) \tag{13}$$

while $\tilde{G}$ is the attenuation of channel modeled by using Affine Arithmetic.

V. PROPOSED DESIGN APPROACH

As shown in Fig.1, the proposed design approach consists of three main parts, which are system modeling, system analyzing and compensation. The flow allows designer to start their work from building executable system model based on the design specification. With the help of the building block library described in section IV designers can build up the system model efficiently and easily. As functionalities and possible uncertainties are already modeled in the building block library, the only work to do in this step is to select the suitable modules. And then set the parameters of the instantiated module appropriately to achieve the expected system behavior.

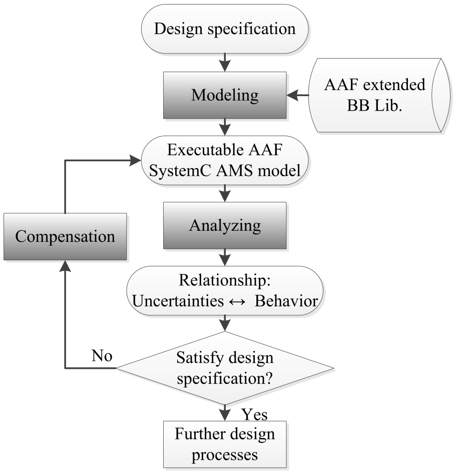

Figure 1. Proposed design flow

In the next step, a SystemC AMS simulation environment is provided to estimate the behavior of the modeled system. Signals in analog subsystems are represented by using Affine Arithmetic symbols. Based on the simulation results, the behavior of the system with uncertainties can be estimated and the impact of certain uncertainties to the system behavior can be analyzed.

If the simulation results show that the design specification is not satisfied or errors occur on certain position, compensations have to be inserted into the system before proceeding to further design processes. we classify the compensation measures into two groups, which are passive compensations and native compensation. Measures which decrease uncertainties by adding or replacing components are considered as passive compensations. A simple example is to use quartz oscillator instead of RC oscillator to reduce the amplitude, as well as frequency jitter of an oscillator block. In contrast to that, native compensations compensate uncertainties from one module by using uncertainties or characteristics of another module in the system. For instance, when converting an analog signal to a digital signal, a range of analog value will be converted to a single digital value, which compensates a certain degree of uncertainties.

VI. ILLUSTRATIVE MODELING EXAMPLES

A simple transceiver system is modeled and analyzed in this section to illustrate the concept of the proposed design approach. The structure of the transceiver system is shown in Fig. 2. The transmitter system first converts a serial stream of random distributed binary digits into two parallel streams. A digital analog converter with two bit wide input port is then used to convert the parallelized bit streams into analog values, which will be furthermore modulated by using a four level amplitude modulation. The modulated signal transmits then through an AWGN channel and is received by the receiver, where it is demodulated and translated back into a digital signal. The needed modules shown in Fig. 2 are all provided by the building block library. Therefore, only module instantiations are needed to build up the system model. Grey blocks in Fig. 2 stand for modules with uncertainties. There are five uncertainties modeled in this system model, which are reference voltage variation from DA converter, amplitude jitter from the two sine generators and phase jitter from the two sine generators. The rest modules of the system model are supposed to be ideal.

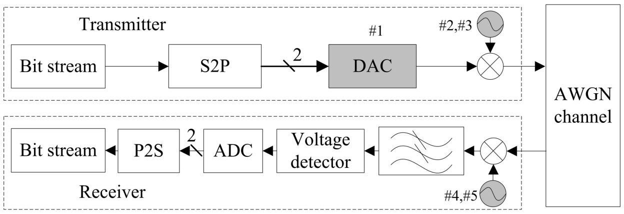

Figure 2. Transmitter Structure

Based on this simulation model, system behavior can be estimated by using SystemC AMS simulator. Table I shows the parameters and their values used in our experiments. f, A and φ stand for the frequency, amplitude and phase of the carrier signal, respectively. V stands for the reference voltage of the AD/DA converter. The footnote nom and dev respect to the nominal value and deviation range. Values represented by symbols means, that they are changed for different analysis purposes. Besides, a Butter worth low pass filter with a cutoff frequency of 15 MHz is modeled by using the Laplace Transfer Function module described in section III.B in the receiver system.

f	A_{nom}	A_{dev}	φ_{nom}	φ_{dev}	V_{nom}	V_{dev}
50 MHz	1	A	0	φ	4	0

Table I
PARAMETERS FOR THE SIMULATION

Fig. 3 shows a signal which is received by the receiver. As shown in Fig. 3, the simulation, different from traditional simulations, provides a signal with its possible variation range. This is comparable to Monte Carlo simulation. However, the simulation here provides the range only in one simulation, when a large number of simulations are needed when Monte Carlo simulation is applied.

The simulation results listed in Table II show us the maximal variation range of each uncertainty (percent in comparison to its nominal value), when all other uncertainties are set to zero and system behaves correctly.

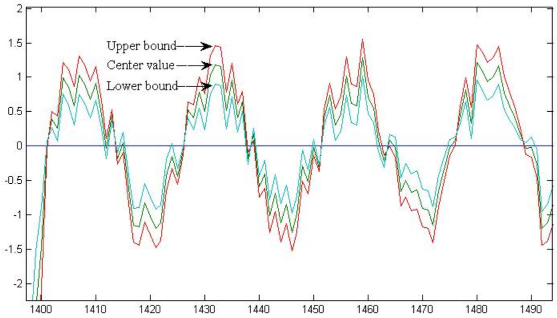

Figure 3. Transmitted signal with AWGN noise

A_{dev_tr}	A_{dev_re}	φ_{dev_tr}	φ_{dev_re}	V_{ref_dev}
10%	9.9%	100%	100%	11%

Table II
MAXIMAL RANGE OF UNCERTAINTIES

As can be read from Table II, the maximal allowed amplitude variation of sine generators in transmitter and receiver is about 10%. The maximal range of the reference voltage variation of digtial analog converter is about 11%. These uncertainties can be compensated by the quantization error of analog digital converter in the receiver system, which is considered as a kind of native compensation measure. Another important phenomenon to be mentioned here is that the phase jitter from the sine generators has almost no effect on the system behavor. It can be found according to Equation 4, the pahse jitter is basically a noise signal with the carrier signal frequency. When it passes through the low pass filter in the receiver. This noise will be removed by the filter. This explaines, why the phase jitter has a very small effect on the system behavior.

If more than one uncertainty are available in the system, the robustness of the whole system reduce greatly. For instance, it can be estimated from the simulation results that in the case that the amplitude variations from the two sine generators are 3%, the maximal variation range of the reference voltage of DA converter is then only 5%.

VII. CONCLUSION AND FUTURE WORK

In this paper, a new design approach for system level robust communication system design is reported. The proposed design approach is based on a pre-implemented SystemC AMS building block library, which is extended by Affine Arithmetic. Various basic modules for modeling communication systems are already available in the library. With the help of the building block library, effort for building robust system model is greatly reduced. With the combination of SystemC AMS and Affine Arithmetic, relation between uncertainties and system behavior can be found efficiently. Compensations can be then applied contrapuntally.

In the next step, more types of uncertainties should be considered and implemented into the building block library. A compensation strategy should be developed, which can guide designers to apply compensations systematically. Furthermore, a framework, which represents the proposed design flow should be developed.

ACKNOWLEDGEMENTS

This work has been supported by the Austrian WWTF project MARC under contract no.ICT08_012

REFERENCES

[1] Claire Fang Fang, Tsuhan Chen, and Rob A. Rutenbar. Floating-Point Error Analysis Based On Affine Arithmetic. In *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, pages 561–564, 2003.

[2] Darius Grabowski, Christoph Grimm, and Erich Barke. Ein Verfahren zur Effizienten Analyse von Schaltungen mit Parametervariationen. In *Tagungsband GI/ITG/GMM - Workshop Modellierung und Verifikation '03*, Dresden, Mar 2006.

[3] Darius Grabowski, Markus Olbrich, and Erich Barke. Analog circuit simulation using range arithmetics. In *ASP-DAC '08: Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 762–767, Seoul, Korea, 2008. IEEE Computer Society Press.

[4] Christoph Grimm, Wilhelm Heupke, and Klaus Waldschmidt. Refinement of Mixed-Signal Systems with Affine Arithmetic. In *Design, Automation and Test in Europe 2004 (DATE âĂŹ04)*. IEEE Press, Feb 2004.

[5] Christoph Grimm, Wilhelm Heupke, and Klaus Waldschmidt. Analysis of Mixed-Signal Systems with Affine Arithmetic. *IEEE Transactions on Computer Aided Design of Circuits and Systems*, 24(1):118– 123, 2005.

[6] Wilhelm Heupke, Christoph Grimm, and Klaus Waldschmidt. Semi-symbolic simulation of nonlinear systems. In *Forum on Specification and Design Languages 2005 (FDLâĂŹ05)*, Lausanne, Sep 2005. ECSI.

[7] M.Andrade, J.Comba, and J.Stolfi. Affine arithmetic (extended abstract). In *INTERVAL '94*, St. Petersburg, Russia, 1994.

[8] R. E. Moore. *Interval Analysis*. Prentice-Hall, Eaglewood Cliffs, NJ, 1966.

[9] Huahao Shou, Hongwei Lin, Ralph Martin, and Guojin Wang. Modified Affine Arithmetic Is More Accurate than Centered Interval Arithmetic or Affine Arithmetic. In *Martin(Eds.), Lecture Notes in Computer Science 2768, Mathematics of Surfaces, Springer-Verlag*, pages 355–365. Springer, 2003.

[10] Alain Vachoux, Christoph Grimm, and Karsten Einwich. Analog and Mixed-Signal Modeling with SystemC-AMS. In *International Symposium on Circuits and Systems 2003 (ISCAS '03)*, Bangkok, Thailand, May 2003. IEEE Press.