Abstract Modeling and Simulation Based Selective Estimation

In Proceedings of Design and Diagnostics of Electronic Circuits and Systems Conference

DDECS 2011

Index

Abstract Modeling and Simulation Based Selective Estimation

Yaseen Zaidi, Sumit Adhikari and Christoph Grimm
Institute of Computer Technology
Vienna University of Technology
A-1040 Vienna, Austria
Email: {zaidi, adhikari, grimm}@ict.tuwien.ac.at

***Abstract*—SystemC AMS offers high abstraction and simulation speed through models of computation and language features such as static scheduling, constant time stepping, linear solver and dataflow paradigm. We demonstrate that such rich expressiveness can render non-ideal behavior in system level description. Design exploration and refinement from system level down to cycle accurate or circuit level is also demonstrated. The main contribution is that the fine grain characterization can start at system level design.**

I. Introduction

Abstraction has been at the forefront in software and digital hardware design domains. However, with the unfolding of SystemC AMS, the analog and mixed signal extensions of SystemC, abstraction in analog design, too, has settled in. Conventional HDL-AMS and SPICE simulations are marked by time step granularity, iterative computation and complexities of non-linear solvers (convergence, stability, stiffness). To design an analog simulation platform that performs at higher levels of abstraction, the working group of SystemC AMS conclusively settled for a light-duty and fast simulation environment, supported by adequate language semantics for greater modeling coverage [1]. The adequacy rests on years of research on abstraction [2] (time, data, communication and computation), synchronous data flow [3], static ordering of computation [4] and use of time tagged tokens [5] rather than propagation of a clock signal. Such automata and semantics are built on top of esteemed SystemC standard, which, in turn draws from the power of concurrency, hardware data types and C/C++ constructs. Although the SystemC AMS standard by no means replaces the necessity and accuracy of a descriptive AMS language, SystemC AMS however can give meaningful insight to specific analog behavior. Illustrated in Fig. 1 the freedom of expressiveness of SystemC AMS is through Linear Signal Flow (LSF), Electrical Network (ELN), Laplace Transform Function (LTF) and Timed Data Flow (TDF) Models of Computation (MoC). Further, when the need of descriptive simulation is confirmed, SystemC AMS simulator's open architecture and synchronization layer [8] can be used to interact with a descriptive simulator.

In Section II we show that non-idealities can be modeled in SystemC AMS at very high abstraction and in Section III we further discuss system level estimation when a host descriptive simulator is used to cosimulate refined models with the coarse SystemC AMS models.

II. Model based Estimation of Non-Idealities

The output signal quality of a Programmable Gain Amplifier (PGA) in a signal processing path affects the device performance, particularly when the gain requirement is high. If the gain bandwidth of the proposed OPAMP can be predicted, the number of gain stages required can be easily determined using gain-bandwidth equation. But the performance of such a derivation can only be estimated by simulating a model which can incorporate analog non-ideal effects. We model the non-idealities in SystemC AMS described by analytical equations.

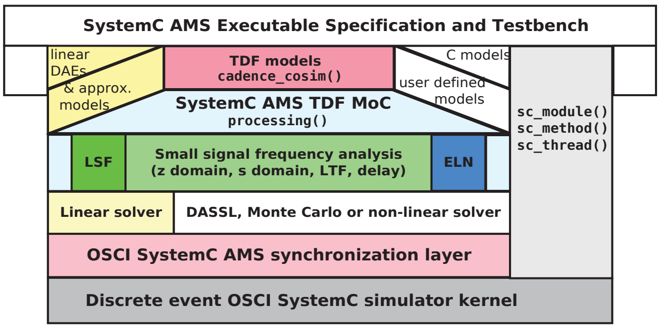

SystemC AMS Executable Specification and Testbench
linear DAEs & approx. models
TDF models
cadence_cosim()
C models
user defined models
SystemC AMS TDF MoC
processing()
sc_module()
sc_method()
sc_thread()
LSF
Small signal frequency analysis
(z domain, s domain, LTF, delay)
ELN
Linear solver
DASSL, Monte Carlo or non-linear solver
OSCI SystemC AMS synchronization layer
Discrete event OSCI SystemC simulator kernel

Fig. 1: Openness and expressiveness of SystemC AMS facilitates user extensibility

A. OPAMP Gain Bandwidth

The continuous time transfer function of a 1^{st} order system can be written as

$$H_{OL}(s),\ H_{CL}(s) = \frac{A_{OL}}{1+\frac{1}{2\pi F_{OL}}\,s},\ \frac{A_{CL}}{1+\frac{1}{2\pi F_{CL}}\,s} \tag{1}$$

where A_{OL} is the gain and F_{OL} is the 3 dB frequency of the open loop system and A_{CL} and F_{CL} are the corresponding quantities for a closed loop system.

When an open loop system operates in closed loop configuration, the product of gain and bandwidth are related through

$$A_{OL}\ F_{OL} = A_{CL}\ F_{CL} \tag{2}$$

The closed loop output response of an OPAMP can be described in terms of known open loop A_{OL}, F_{OL} parameters

$$H_{CL}(s) = \frac{A_{CL}}{1+\frac{1}{2\pi\frac{A_{OL}\ F_{OL}}{A_{CL}}}\,s} \tag{3}$$

Given A_{OL}, F_{OL} and A_{CL}, we model the transfer function of an OPAMP as a PGA, using Eq. 3 and LSF MoC: `(Acl,1.0,1.0/(2.0*M_PI*Aol*Fol/Acl))`.

B. OPAMP Saturation and Supply Noise

Realistically, the output of the PGA cannot have any value dictated by Eq. 3 as the output cannot exceed supply voltages. This poses a need of limiting the output of the PGA to supply voltages. Limiting the PGA output can be easily implemented using a TDF MoC which simply uses `if-else` statements for supply-rail voltage monitoring. Monitoring supply rail voltages also helps implementation of supply noise and leakage effect of 50 Hz component on the system. The power supply has been modelled as a separate block which uses a uniform random number to generate white noise on supply rails. The PGA uses these supply rails as input TDF ports. The choice of uniform random number suffices the need to generate the noise due to thermal excitation of charge carriers in the power supply.

C. OPAMP Gain Non-Linearity

For higher swing of the input signal, the gain of OPAMP goes non-linear which needs to be accounted as a large signal effect. An efficient analytical approximation which accounts for gain non-linearity using hyperbolic tangent function has been proposed in [7]

$$V_{out} = a \ \tanh\left(b \ V_{in}\right) \tag{4}$$

The selection of coefficients comes from DC analysis of a circuit level OPAMP. To empirically find out a and b, we simulated an existing simple circuit level OPAMP and abstractly modeled the non-linear gain as `Vout.write(a*tanh((b*Vin.read())))`.

D. OPAMP Slew Rate

The slew rate limits the rate of output signal to change faster than a specified value, which is the manifestation of large signal behavior. The typical definition of slew rate is the maximum allowed time rate of change of output voltage of the OPAMP and is described as [6]

$$SR = \max\ \left(\mid \frac{dV_{out}}{dt} \mid\right) \tag{5}$$

Eq. 5 is modelled using a simple rate comparison and limitation function in TDF.

E. OPAMP Input Referred Noise

OPAMP input-referred noise is the thermal noise due to OPAMP which when viewed as its appearance at the input of OPAMP. This effect can be easily modelled using an additive uniform white noise at the input of the OPAMP through TDF MoC.

F. Band-Gap Noise

When carefully designed, a band-gap reference output shows only thermal noise which is uniform white noise effecting the common-mode voltage of the OPAMP. We modelled the band-gap reference as a separate TDF module. The output of the band-gap appears as the TDF input of the PGA top-level. The output of the common-mode reference is then subtracted from the signal value which enters the LSF transfer function description and again added to the output of LSF transfer function description to ensure introduction of band-gap noise and prevention of amplification of common-mode voltage.

G. PGA Top Level Integration and Simulation

With the effects described through Sections II-A to II-F, we integrate the non-ideal behavioral blocks to form a high level PGA model shown in Fig. 2.

TABLE I: Signal Path Parameters

Sampling Frequency	1 MHz
$V_{supply}(+)$	1.2 V
$V_{supply}(-)$	0 V
Reference Voltage	0.5 V
OPAMP Open Loop Gain	60 dB
OPAMP Open Loop Band-width	10 kHz
OPAMP Slew Rate	20 V/μs
OPAMP Input Referred Noise	100.0 μV
Bang Gap Reference Noise	100.0 μV
Power Supply Noise	100.0 nV
Power Supply 50 Hz Component	0.0 V

A sinusoidal of 50 Hz is used for exciting the system. Major parameters used during simulation are given in Table I. The OPAMP when used as an unity gain buffer configuration shows the quality of the output accounted in Fig. 3a. The result shows that the noise ground is at -20 dB and the 3^{rd} harmonic is at 0 dB, that is, effectively, the output of the buffer is 16-bit clean. The requirement of PGA gain of 80 dB instantly suggests two stage PGA, but the performance of such a suggestion needs to be estimated through simulation. This has been done instantiating two OPAMPs in cascade with each bearing 40 dB gain. The result is shown in Fig. 3b. The FFT result in red is for the single stage output when operated with a gain of 80 dB, which is even higher than the open loop gain configuration of the OPAMP and is completely impractical but it gives a nice reference for comparison showing insufficiency for single stage high-gain and wide-band operation. The FFT plot in blue is the two stage PGA output with each PGA bearing a closed loop gain of 40 dB, showing the signal quality of such a choice. For the chosen configuration the noise ground is high (at 55 dB) and the 3^{rd} harmonic is at 70 dB showing the loss in signal quality for such high gain stages although the configuration meets our requirement.

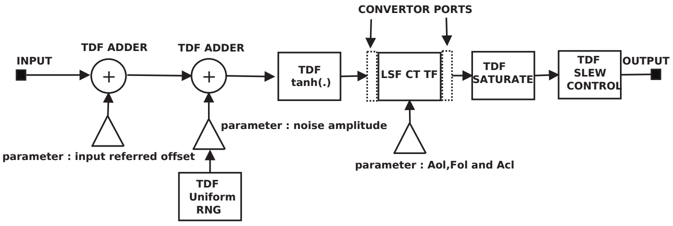
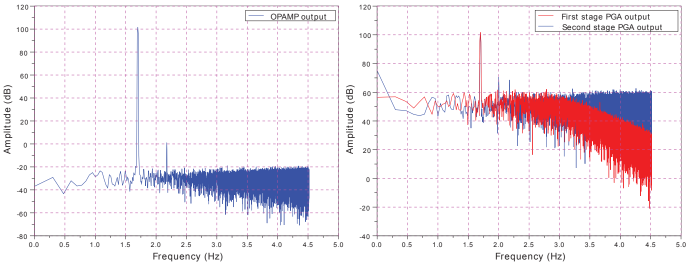


Fig. 2: OPAMP macromodel



(a) Output of a single stage PGA as unity gain buffer at input sinusoid of 50 Hz, 470 mV

(b) Output of a two stage PGA at input sinusoid of 50 Hz, 47 μV

Fig. 3: Simulating non-idealities in an OPAMP at system level

III. Simulation based Refinement and Estimation

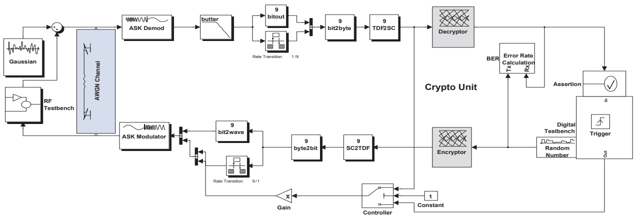


Fig. 4: ASK transceiver modeled in SystemC AMS, crypto unit is cosimulated as DES in RTL with Cadence IUS

We now show results of SystemC AMS to Cadence IUS cosimulation. The main idea is that at low rates the TDF node to be cosimulated follows the scheduling semantics of a TDF cluster and the synchronization by SystemC AMS synchronization layer illustrated in Fig. 1. The cosimulation wrapper hosted inside a TDF module will be executed if the port rate (required number of data samples) is met. The execution of the wrapper does not change the balance equation or the TDF topology, rather the interface is a user C++ function to the TDF execution semantics. Therefore, the synchronization is autonomous such that it requires no middleware software. This synchronization schemes works best at architectural level, at which the simulation speed is kept high by processing low number of tokens such as the example of Fig. 4. The simulation data of Cadence simulator is accessed as C data type via VHDL Programming Language Interface (VHPI). Further details of the cosimulation interface can be found in [8].

Fig. 4 depicts executable specification of a transceiver. All blocks are connected by signal flow whereas the data is time tagged tokens that are equi-separated in time. The analog blocks are Amplitude Shift Keying (ASK) modulator and demodulator. A digital block is crypto unit. A controller is software block which selects the cryptographic scheme. The entire description is abstract e.g. the behavior of crypto unit captured in one line of code by adding a parity bit: `parval8=parval.range(7,0).xor_reduce()`. System performance is evaluated in bit error rate calculation block.

A. Simulation Based Estimation

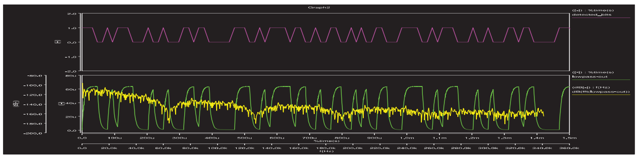


Fig. 5: Encrypted bits detected as output from a lowpass filter

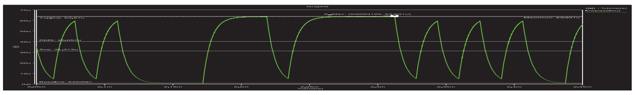


Fig. 6: Average (31 μV) and RMS (40 μV) amplitudes

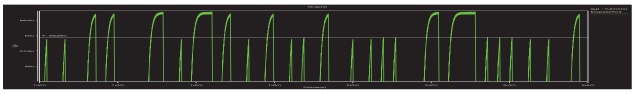


Fig. 7: Maximum amplitude swing

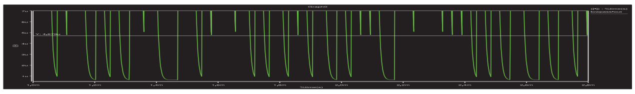


Fig. 8: Minimum amplitude swing

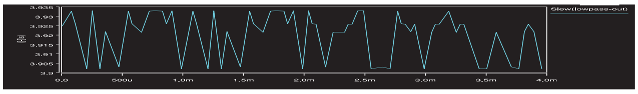


Fig. 9: Amplitude slew rate

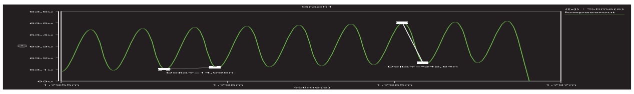


Fig. 10: Ripple measurement in amplitude peak

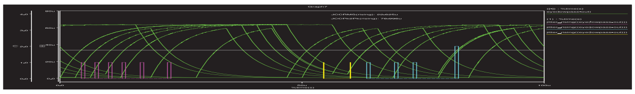


Fig. 11: Jitter in rise

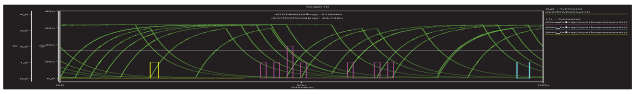


Fig. 12: Jitter in fall

The specification in Fig. 4 is heterogeneous and simulates in a few msec. We directly refine and replace the abstract crypto unit block by cosimulation of VHDL softcore of 64-bit Data Encryption System (DES) and make amplitude measurements on the encrypted ASK signal. Fig. 5 shows the output of low pass signal, FFT of the output signal in yellow and equivalent digital representation of the bits in pink. In Fig. 6 we measure max amplitude at nearly 64 μV, lowest drop at 643 nV and an average value around 31.3 μV. Figs. 7 and 8 show a maximum vertical peak to peak variation of about 2.5 μV and a minimum vertical peak to peak variation of about 4 μV. The variation in amplitude as time progresses is measured in Fig. 9. The passband ripple in Fig. 10 is 342 nV and peak to peak delay of 166 nsec. The ripples can be adjusted by designing for an appropriate ripple factor ϵ, specified as $20 \log_{10} \frac{1}{\sqrt{1+\epsilon^2}}$. Figs. 11 and 12 show eye plots in lowpass detected bits both in rising and falling jitter edges respectively. For rising edge the RMS jitter is 22.625 μ and the peak to peak jitter 79 μ. We estimate Bit Error Rate (BER) using a scaling factor α related to jitters as

$$Jitter_{PP} = \alpha \; Jitter_{RMS} \; , \; BER = \frac{1}{2} \; erfc \; (\sqrt{2} \; \alpha) \tag{6}$$

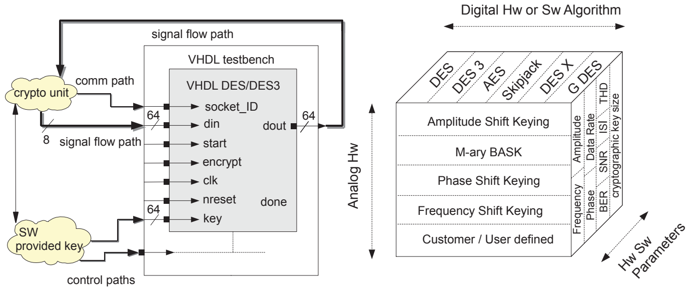
For rising edge trigger the estimated BER is 1.5×10^{-12} due to lower peak to peak jitter and for falling edge trigger (steeper roll off and higher bound of Gaussian distribution) the BER is estimated to be 2.15×10^{-18}. The jitter and BER budget can be estimated for varying signal lengths though there is slight variation in α. The designer can further estimate system performance by experimenting with other figures of merit such as the insertion loss, carrier frequency and threshold voltage for detecting high and low bits.

B. Abstraction in Control, Communication and Signal Flow

TDF MoC embeds own solver as a user defined function while the dataflow in TDF cluster is time tagged tokens. Both of these features permit for an abstract interface between SystemC AMS and Cadence. The abstraction concepts also hold in connecting both the simulations. Illustrated in Fig. 13a, the signalflow path only handles simulation data, while RTL model signals clk, $start$, $encrypt$ etc. are local to the model. On the system side (SystemC AMS) the data is 8-bit but this data is encrypted as 64-bit on circuit side (RTL). The communication path is active when the system simulator initiates cosimulation or the cosimulator finishes its simulation. The software controller via the control path selects the type of the cryptography RTL algorithm to be cosimulated.

C. Designer's Degree of Freedom

The designers degree of freedom begins with estimating performance parameters. In case the performance is not up the mark, the



(a) Abstraction in the two simulators' interface prevents tight coupling and synchronization

(b) Analog, digital and software refinement in one-go

Fig. 13: Co-analysis thru high simulation speed, abstract interfaces and loose synchronization at system level

designer can change the model altogether. For example, the designer has the leverage of experimenting with modulation techniques (FSK, PSK) if certain performance measurements (noise, distortion, SNR) are unacceptable under ASK. If the prime dynamic figures appear feasible, the designer can continue with further estimation under the same modulation scheme e.g. signal quality, amplitude, frequency, phase or improving signal detection in the receive path (top in Fig. 4) with various lowpass filter configurations. Optionally, based on poor detection of the demodulated signal, the designer can correct the gain at the modulator input. For better intersymbol interference, RTL models of cryptographic algorithms with different key sizes can be cosimulated. Fig. 13b illustrates this degree of freedom for the system designer who can explore the design space by model choices and estimating parameters in all domains. Such variation in design exploration is possible because of high simulation speed, rich expressiveness in MoCs and little programming effort.

D. Execution Performance

The computation time for 20 cosimulation instances are shown in Fig. 14 for several models. The latency is inherent of distributed computing, typically it is in the order of hundreds of msec. These times can be reduced by using shared memory as opposed to sockets and also by skipping exhaustive steps of compiling and elaboration i.e. simulating only pre-compiled and pre-elaborated models.

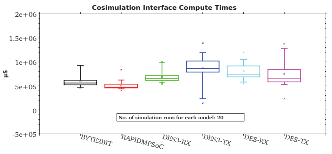


Fig. 14: Temporal deadlock in TDF nodes during cosimulation

E. Limitation and Outlook

The future release of SystemC AMS is planned to support variable time stepping which would enhance non-linear and non-ideal behavioral modeling. Further, currently no commercial simulator hosts analog procedural interface, although both VHDL-AMS and Verilog-AMS language working groups plan standards for analog procedural interface. The cosimulation framework cannot perform C level access of analog quantities. Therefore, we cannot couple analog signals at circuit simulator to the system simulator. However, analog circuit detail is usually wrapped in a digital description such as top level VHDL or a testbench and since both VHPI and Verilog Procedural Interface (VPI) can access real valued discrete time quantities, we can link any discrete time signal from the circuit simulator to the signal flow path between the two simulators. Future version of the cosimulation interface shall address analog VHPI or VPI.

IV. Conclusion

SystemC AMS, as both a C/C++ based system design language and a high level simulator, has been shown to model finer and specific behavior through its diverse MoCs. Although the language does not provide circuit design constructs for the notions such as cycle accuracy by a clock, boolean equation solver, adaptive stepping, non-linear solver, analog operations (slew rate, crossing, wave smoothening), time derivatives and integrals, it's C++ modeling capabilities and MoCs can still render similar behavior at much higher level of abstraction and simulation speed. Estimation of OPAMP non-idealities at high simulation speed has been demonstrated in exploring the design space and determining the number of stages required for particular gain. This estimation is largely based on algebraic equation driven modeling. Further, the SystemC AMS simulator's open architecture and synchronization layer can interface with a hardware description language based dedicated analog and mixed signal simulator where HDL based analog and mixed signal modeling capabilities are an absolute must. To this end, Cadence IUS has been coupled to replace an abstract model (parity checker) in the executable specification of an ASK modem, by cosimulation of RTL DES model. The advantage of using coupled simulator with SystemC AMS is that finer models already captured at much detail (HDL/HDL-AMS) can be cosimulated with much abstract models (SystemC AMS). Therefore the simulation speed cost is incurred at blocks which are simulated as fine grain whereas the overall high level description simulates much faster since it uses fixed time stepping, TDF modeling paradigm and timed tokens. The time penalty for cosimulation is about one second for DES/DES3 models shown as a case study. The overall cosimulation is used to characterize amplitude in the ASK modulation as a test case while other specific design parameters can be similarly experimented at the high simulation speed as SystemC AMS language is developed to quickly realize virtual architectures, explore design space and early estimation before actual low level designing begins.

References

[1] A. Vachoux, C. Grimm and K. Einwich, "SystemC-AMS requirements, design objectives and rationale," Design, Automation and Test in Europe Conference and Exhibition, 2003, pp. 388- 393, 2003.

[2] A. Jantsch, *Modeling Embedded Systems and SoCs: Concurrency and Time in Models of Computation*, Morgan Kaufmann Pub., 2004.

[3] E. A. Lee, and D. G. Messerschmitt, "Synchronous Data Flow," Proceedings of the IEEE , vol. 75, no. 9, pp. 1235- 1245, Sept. 1987.

[4] E. A. Lee, and D. G. Messerschmitt, "Static Scheduling of Synchronous Data Flow Programs for Digital Signal Processing," Computers, IEEE Transactions on , vol. C-36, no. 1, pp. 24-35, Jan. 1987.

[5] R. Alur and D. L. Dill. "A Theory of Timed Automata", Theoretical Computer Science 126:183-235, 1994 (preliminary versions appeared in Proc. 17th ICALP, LNCS 443, 1990, and Real Time: Theory in Practice, LNCS 600, 1991).

[6] Sumit Adhikari and Christoph Grimm, "Modeling Switched Capacitor Sigma Delta Modulator Nonidealities in SystemC-AMS", Specification Design Languages, 2010. FDL 2010. Forum on, 14-16 Sept. 2010.

[7] Abdelfattah,K. and Razavi,B. "Modeling Op Amp Nonlinearity in Switched-Capacitor Sigma-Delta Modulators", Custom Integrated Circuits Conference, 2006. CICC '06. IEEE, pp. 197-200, 10-13 Sept. 2006.

[8] Y. Zaidi, C. Grimm and J. Haase, "Fast and Unified SystemC AMS - HDL Simulation," Specification Design Languages, 2009. FDL 2009. Forum on, pp. 1-6, 22-24 Sept. 2009.