Analog Mixed Signal Extensions for SystemC

White paper and proposal for the foundation of an OSCI Working Group (SystemC-AMS working group)

Karsten Einwich	Fraunhofer IIS/EAS	Karsten.Einwich@eas.iis.fhg.de
Christoph Grimm	Univ. Frankfurt	Grimm@cs.uni-frankfurt.de
Alain Vachoux	EPFL Lausanne	Alain.vachoux@epfl.ch
Natividad Martinez Madrid	FZI Karlsruhe	martinez@fzi.de
Felipe Ruiz Moreno	FZI Karlsruhe	ruiz@fzi.de
Christian Meise	Univ. Frankfurt Professur TI, Continental Teves	meise@ti.informatik.uni-frankfurt.de

Abstract

SystemC supports a wide range of Models of Computation (MoC) and is very well suited for the design and refinement of complex digital systems from functional downto register transfer level. However, for a broad range of applications, interactions of the digital part and algorithms on the one hand and analog parts and continuous-time environment on the other hand are very complex. Therefore, is it essential to consider the analog parts within the design process of Mixed-Signal System on Chip solutions.
Because simulation performance is especially for the analog parts very crucial, different analog MoC must be introduced to allow the use of the most efficient solver for the considered level of abstraction. In this white paper we describe possible areas of application and requirements for mixed-signal extensions for SystemC, and propose the foundation of a “Mixed-Signal Working Group”.

1. Introduction and Motivation

SystemC 2.0 provides a very flexible way to model the communication of systems by specifying a model for communication and synchronization in a channel and providing an interface for the channel, which can be used independently from a specific realization of a channel. This is generic enough to describe systems using various models of computation, including static and dynamic multirate dataflow, Kahn process networks, communicating sequential processes, and discrete events.
However, the discrete approach to modeling and simulation is not applicable to systems, whose behavior is specified by differential equations. Note, that the differential equations can also be given by a netlist, or by transfer functions, for example. Such systems are called *analog systems*. For many classes of systems, the analog parts will become more dominant in respect of performance parameters, power consumption, silicon cost and yield. Analog systems are simulated by mathematical methods that compute a solution for the underlying set of differential and algebraic equations. For simulation of analog systems, a *solver* computes a solution of the differential and algebraic equation, and uses numerical integration techniques to compute signal values for a time interval. For combined discrete/continuous simulation, the discrete event simulators are coupled and synchronized with the analog solver.
In the following, we describe areas of applications to be considered by the analog extensions, and formulate design requirements. Furthermore, we describe a solver integration concept that meets the formulated requirements.

2. Areas of application and Requirements

Areas of application

Analog and mixed-signal systems can be found in many applications. The requirements for modeling and simulation depend both on the area of application and the level of abstraction of the model. Considering that SystemC is used rather for system-level design, the proposed mixed-signal extensions should as well focus on system-level design. On system-level, analog models are used for defining executable specifications of:

- Signal processing functions
- Hierarchical structures (netlists)
- Abstract behaviours (equations)
- The environment of embedded systems

We plan to consider system-level design in the following application domains for the definition of the requirements:

- Signal processing dominated applications (telecommunications and multimedia)
- RF (Wireless communication) applications
- Automotive and power electronics

These application domains have different requirements for modeling and simulation of analog and mixed-signal systems.

Requirements for signal processing dominated applications:

- Sampled systems with a fixed time step.
- Description as directed signal flow graphs (no conservative semantics in analog parts).
- Graph nodes are linear dynamic functions(e.g. transfer functions) or static non linear functions.
- Equations to be solved are in explicit form (i.e. in the form y=f(x)) and can be in most cases ordered such that unknowns can be computed in sequence. If it is not possible to find such an ordered set of equations then are algebraic loops (or coupled equations) and iterative techniques have to be used.
- Description in the frequency domain.

Requirements for RF/wireless applications:

- Possible tight interaction between event-driven (control) and analog parts (e.g. automatic gain control or adaptative filtering).
- Ability to simulate the baseband behaviour

Requirements for automotive and power electronics applications:

- Stiff non-linear systems.
- Multi-discipline systems (e.g. mechanical, electrical, fluidic parts).
- Real-time behavior. This means that the model should execute in time steps which are constrained by an maximum execution time border

Requirements for Integration in SystemC

SystemC has a very generic and flexible concept for modeling systems, which is based on a layered structure of the class library. Basically, SystemC consists of a generic discrete event simulator kernel, templates for modeling communication, and synchronization in different MoCs and concrete data types for modeling digital hardware. The mixed signal extensions shall allow the SystemC framework to support analog MoCs in a generic and flexible way and to provide an efficient synchronization layer between (possibly only some) event-time and continuous MoCs.

3. Basic MoC integration concept

Layered approach

The requirements for modeling and simulation in the application domains formulated above are very different, and even partially contradictory. To meet the different requirements, an extendable and generic concept is necessary. However, the complexity should be limited. To meet these oppositional requirements, we propose a layered approach, which is similar to the approach of standard SystemC. The layered approach is shown in figure 1. On top of the existing standard SystemC kernel, we plan to realize a synchronization layer. The synchronization layer provides a method to couple and synchronize different analog simulators (solvers).

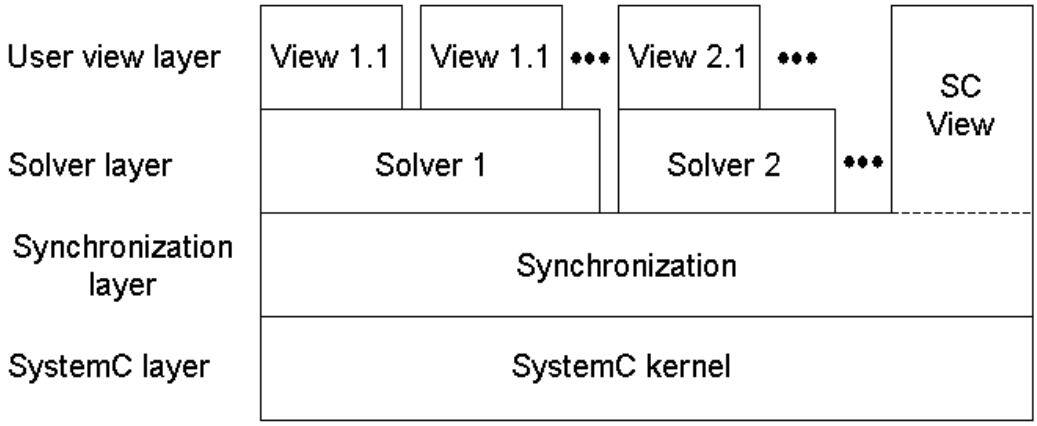

Figure 1: Layered approach for mixed-signal extensions.

The synchronization layer should be unique for analog extensions. Thus all communication between different MoC takes place via the synchronization layer. On top of the synchronization layer, different MoCs can be realized by different analog solvers.

Properties of the Synchronization Layer

The synchronization layer is the most critical part of the mixed-signal extensions. This layer should encapsulate the MoCs as far as possible, and should allow the connection of blocks modeled with different MoCs. To achieve this encapsulation we propose the following restrictions:

Increasing time only, no backtracking: In the simulation of analog and mixed-signal systems, it is a common technique to synchronize different simulators only when this is really needed. This can lead to a situation, where an event from one simulator occurs, which is in the past for other simulators. Then, a rollback mechanism could set the simulation time back to a point in the past. However, this requires the ability to save a state and to restore the saved state. This is not supported by SystemC, and the synchronization layer therefore can only increase the time.

Directed signal flow: In the considered areas of application, the signal flow between modules is directed. Therefore, on synchronization layer the realization of a directed signal flow is sufficient.

Taking into account the different domains of application, we plan to provide a generic synchronization mechanism, which can work with different methods for selecting time steps:
For simulation of signal processing systems, constant time steps usually are precise enough: Such systems are mostly over sampled with known and constant time steps. Such a synchronization should lead to a high simulation performance.

For the simulation of control systems, e. g. in the automotive domain, systems are nonlinear and often stiff systems. The use of constant step width would lead either to a large error, or to a very bad performance. Therefore, we plan to allow variable step sizes.

SC View/Description layer

This layer represents a direct connection to the synchronization layer for those already existing SystemC descriptions. Thus, the compatibility is assured.

Solver layer

A solver provides a certain MoC, e.g. a solver for a linear DAE-system, or another solver for a non-linear system, or a solver optimized for power electronics. The solver must be able to interact with the synchronization layer. As input the solver gets the equation system in a solver specific form e.g. as constant matrices or matrices with method pointers.

Static MoCs

Another class of MoC that has to be supported is the static analysis (i.e. AC/Noise/DC analysis) of analog descriptions and descriptions with an analog specification like digital filters. There is a coupling between static MoCs and other MoCs in the equation set up phase only. The coupling between static MoCs and the event-driven MoC of SystemC only consists in defining consistent stable states at their references (possibly in the Synchronization layer). Especially for digital modules with an analog specification an optional implementation can be necessary.

View/Description layer

The interface layer provides the solver with the equation system. This equation system can be e.g. generated from a network using the modified nodal analysis or from a behavioral representation like transfer function or state space system. The same interface can be useful for different solvers (e.g. linear / nonlinear DAEs); nevertheless, the realization must take into account that the mapping to the solver layer is different.
At least the following interfaces should be part of an analog and mixed-signal SystemC extension library:

- *Netlist interface:* This interface should be common to all underlying MoCs.
- *Equation interface:* This interface should allow a user to formulate behavioral models or functional specifications in a direct way as a differential algebraic equation.

4. Relations to other OSCI working groups and commercial tools

For the realization of the synchronization layer, which connects different analog MoCs, we have similiarities to the synchronous dataflow model proposed by the “Kahn Process Network and Synchronous Dataflow” WG. The synchronous dataflow model has the potential to speed up the synchronization layer proposed in section 3, because no scheduling is required at simulation time. Nevertheless, for the realization of the mixed signal extensions, some further features like a much closer connection to the time are required. Therefore the definition of the synchronization layer has to be synchronized with the dataflow working group. For the integration of static MoCs the results of the API working group are essential. To set up the equation system this MoC needs to explore the structure (the instances and there connections).
Commercial tools for simulation of analog and mixed-signal systems are most notably Saber, Eldo, Spectre, different VHDL-AMS/Verilog-AMS simulators and Matlab/Simulink. Possible analog extensions to SystemC are not intended to replace mature simulators for circuit level design. The focus of the intended extensions is more on system level design, but should focus on abstract modeling and MoCs that permit the fast and efficient simulation Due to it’s open architecture, a coupling of circuit level simulators such as Saber or Spice is feasible on top of the synchronization layer

5. Proposed Objectives, Class Design Issues

Language design goals

The SystemC libraries to support the analog and mixed-signal systems outlined in this document should satisfy the following objectives:

- Simulation performance by abstraction and restrictions
- Technical feasibility with adequate effort
- Security (error detection/localization; non ambiguity)
- Modularity (independence of modules/encapsulation, scalability, IP assembly)
- Manageable complexity (clear and limited number of interfaces to other MoCs)
- User understandable algorithm
- Mechanism to include lower levels of abstractions, expandability, flexibility
- Use of common netlist format (extension/subset of existing SystemC format)

Class design issues

In order to realize the extension library proposed in this white paper, we plan to take the following actions:

1. Definition of a synchronization mechanism: How do we coordinate and synchronize the time advancement in the analog solvers? Here, we must ensure, that no solver simulates too far into the future (and might have to be set back to a simulation time in the past), and that all simulators get appropriate inputs (no deadlocks).
2. Definition of MoC: Which is the underlying MoC which computes system behaviour from the behavior of the single solvers (e. g. static dataflow). How do we ensure or improve numerical stability and convergence of the overall simulation?
3. Backplane: Formulation of an interface that allows the realization of the synchronization mechanism.
4. Netlist format definition and specification of equation interface.
5. Definition of analog datatypes
6. Design methodology: How can the extension library be integrated in existing design flows, how can we make model creation easier and more efficient, etc.

Validation

1. Identification of test cases from targeted application domains.
2. Evaluation of applicability and performances of the mixed-signal extensions.

Documentation

1. Functional Specification document.
2. SystemC-AMS library documentation (classes + synchronization layer).
3. User Guide.

Dissemination

1. Trainings, seminars, papers, conferences.

6. Development roadmap

We propose to realize the mixed-signal extensions in basically three phases, each one achieving one level of implementation (level capabilities are additive):

Phase 1 / Level 1

- Main application domain targeted is signal-processing dominated applications.
- Supported analog MoCs are linear DAEs (Differential Algebraic Equations). Explicit form of equations. Behavior as a sequence of assignments.
- Library of dedicated functions (e.g. transfer functions).
- Structural composition (netlists). Library of primitive elements (e.g. R, C, L, sources).

- Time-domain simulation. Limited static AC/DC capabilities.
- Synchronization with static dataflow MoC. Only constant time steps. No time backtracking. Only "weak" mixed-signal description (e.g. control or status signals).

Phase 2 / Level 2

- Main application domain targeted is RF/wireless applications.
- Supported analog MoCs are nonlinear DAEs. Implicit form of equations. Behavior as true simultaneous statements.
- Support of initial conditions.
- Static MoC prototype.
- Synchronization with variable time steps.
- Equation-based description (non-linear equations to be solved interactively)

Phase 3 / Level 3

- Main application domain targeted is automotive applications.
- Specialized analog MoCs (e.g. for power analysis and mechanical problems).
- Synchronization between analog MoCs.
- Generic concept for synchronization with other MoCs

7. Conclusion and proposal

Analog and mixed-signal extensions for SystemC are necessary in a broad range of applications, for example in telecommunication, multimedia and automotive applications. In order to cover the very different requirements, an analog and mixed-signal extension library should be structured into a synchronization layer, which is independent from the solvers to be used, and solver/description-layers. Such a library could significantly improve the generality of the SystemC approach and would widen the scope of SystemC to a broad range of applications, where SystemC nowadays is focused on pure discrete hardware/software systems.
Therefore, we propose, that there should be an “OSCI SystemC-AMS Working Group”. This working group should consider this white paper as its initial brief.