Microelectronics Journal 38 (2007) 805–815

Microelectronics Journal

www.elsevier.com/locate/mejo

Generic discrete–continuous simulation model for accurate validation in heterogeneous systems design

Faouzi Bouchhima[a,b,*], Gabriela Nicolescu[a], El Mostapha Aboulhamid[c], Mohamed Abid[b]

[a] *Ecole Polytechnique de Montréal, Succ. Centre-Ville, C.P 6079, Montréal, Canada*
[b] *Ecole National d'ingénieurs de Sfax, Route Soukra, BP w 3038, Sfax, Tunisia*
[c] *Université de Montréal, Succ. Centre-Ville, CP 6128, Montréal, Canada*

Received 5 April 2006; accepted 2 April 2007
Available online 31 May 2007

Abstract

Heterogeneous systems combining several technologies will potentially overcome several performances and applicability limitations, as well as providing new useful features. However, their design is currently confronted to important challenges. One of the key challenges is the integration of pre-built components specific to different application domains (e.g. electrical, mechanical, optical, etc.). In this context, new CAD tools are mandatory to offer a global view of the designed systems and to enable their overall validation. This paper presents a generic discrete–continuous simulation model for an accurate global validation in heterogeneous systems design. This model enables to use powerful tools for the discrete and continuous domains. Solutions are proposed for model's layers implementation in the case of SystemC and Simulink simulators. The continuous–discrete simulation tool is proposed. This tool integrates the two previous simulators by automatically generating global simulation model instances. The evaluation of the simulation model was performed using an illustrative application.

Keywords: Continuous/discrete simulation model; Synchronization model; Matlab–simulink; SystemC

1. Introduction

Modern systems like mixed-signal systems and real-time controllers integrate discrete and continuous components. Nowadays, systems on chip are the main drivers for convergence of multiple technologies. In 2001, FRAM and FPGA were integrated on chip then follow in 2003 MEMS and chemical sensors; in 2005 electro-optical; and in 2006 electro-biological. The 2006 ITRS update edition [1] highlights that "the development of new technologies that are placed side-by-side to a digital design in a silicon die presents a whole set of new challenges. Examples are MEMS, electro-optical devices and electro-biological devices. These new components will require modeling of both the interface between the digital portion and the non-digital components and a proper abstraction of the non-digital system behavior in order to still be able to verify the digital portion of the system". These systems will be ubiquitous in communications, automotive, medical and other domains.

The global validation of these systems requires new techniques enabling reusability, high abstraction levels and simulation accuracy from a time point of view. Currently, in order to respect the tight constraints of time-to-market, cost and performance, the heterogeneous systems are designed by reusing pre-designed components. Thus the design flow may be characterized by two main aspects: (1) building components that may be reused and (2) helping in their integration. This type of design represents an important challenge; one of the key issues being the integration of the pre-built components specific to different application domains (e.g. electrical, mechanical, optical, etc.) [2,3]. Nowadays, designers build different components to be integrated by using powerful existing tools specialized for a specific application domain (e.g. SystemC for the electronic digital part, Matlab/Simulink for the mechanical

*Corresponding author. Ecole Polytechnique de Montréal, Succ. Centre-Ville, CP 6079, Montréal, Canada.
E-mail address: faouzi.bouchhima@polymtl.ca (F. Bouchhima).

0026-2692/$ - see front matter
doi:10.1016/j.mejo.2007.04.001

part, etc.) and they often prefer to keep using their current tools. Consequently, new CAD tools for heterogeneous systems must be based on global simulation models defined independently from specific languages or simulators. This permits the integration of best and adequate existing tools in order to exploit their full capabilities. To reach this requirement, the co-simulation technique is used in this work, even though it decreases the simulation speed, it allows the designer to describe each, the continuous and the discrete model, in a specific and appropriate language. The co-simulation model must be generic (independent from languages or simulators) in order to respect the mentioned requirement. Some difficulties in the definition of a continuous/discrete co-simulation model are (1) the heterogeneity of the definitions of the concepts manipulated by the discrete and the continuous components and (2) the need of continuous/discrete communication and synchronization. This is eased by using co-simulation interfaces. These interfaces have a great influence on the accuracy and the performance of the global simulation. Their automatic generation is very important, since their design is time consuming and an important source of errors.

This paper proposes a generic architecture of a discrete–continuous simulation model for an accurate global validation in heterogeneous systems design. Solutions are provided to implement this model in the case of SystemC and Simulink simulators. This paper presents the continuous–discrete simulation (CODIS) tool for the automatic generation of the global simulation model (co-simulation model) instances. The proposed model is evaluated using an illustrative application.

The rest of this paper will be organized as follows: Section 2 describes the related work. Section 3 introduces the continuous and the discrete simulation models and the main concepts involved in both models. Section 4 defines the synchronization model, and introduces and details the generic simulation model. Section 5 details the simulations interfaces and their implementation. In Section 6, the CODIS tool is introduced. Section 7 gives experimentation results and accuracy analysis. Finally, Section 8 concludes the paper.

2. Related work

In the past years, although research in the discrete/continuous simulation area has accelerated, it is still behind its digital counterpart. A short overview about related work and existing tools is given in this section.

In [4], Tahaway et al. present a simulation model based on VHDL and ELDO. Their model is at the physical level and is based on the lock-step approach with a fixed time step. Other works use the notion of co-simulation backplane to integrate mixed-signal simulators. In [5], the authors develop a collection of functions (backplane) that allow simulators to share data during simulation. They integrate an open source Spice simulator and a logic simulator. In a similar way, the authors in [6,7] propose a co-simulation environment based on Xyce (a SPICE parallel simulator) and SAVANT (a parallel VHDL simulator). The interfacing is enabled by C++ classes containing methods for signal conversion and data exchange between simulators.

The Nexus-PDK environment proposed by Celoxica [8] supports co-simulation of cycle accurate C, C++ and Handel-C models with SystemC, MATLAB/Simulink, VHDL and Verilog simulators. All the models integrated in this environment are discrete. Similar approach is adopted by Active-HDL [9].

Other approaches propose the utilization of a unique language for the specification of the overall system. Some of these languages may be obtained by extension of well-established languages. Illustrative examples are VHDL-AMS [10], Verilog-AMS [11] and recently, SystemC-AMS [12] extending, respectively VHDL, Verilog and SystemC for mixed-signal systems design. The first trials, contributing to the deeper understanding of the problems in using SystemC and VHDL for mixed-signal systems design are presented in [13,14]. In [15], a based SystemC framework that supports signal processing-dominated applications is proposed. The synchronization between the synchronous dataflow and linear continuous time is using fixed time step. Another framework is proposed in [16], the authors presented a mixed-signal simulation to simulate an analog-to-digital data converter. The framework includes C++ mixed-signal modules. They implement a virtual clock for the scheduling of the analog blocks to avoid multiple executions of them due to the SystemC scheduler.

Modelica [17] is a language for the design of heterogonous systems. Several commercial simulation tools are based on it (e.g. Dymola, MathModelica). This language provides a set of libraries for several application domains (electrical, thermal, etc.). However, the concept of discrete events is difficult to manipulate in this language. Ptolemy [18] is a java-based environment that considers heterogeneous systems represented as a set of components whose interaction styles are governed by computation models implemented as “domains”. It provides a unified infrastructure to assure hierarchical composition of these computation models.

In summary, a number of attempts were proposed for mixed-signal simulation. They require abandoning well-established efficient tools for specific domains [10–12,18]. Moreover, most of the proposed heterogeneous approaches are application-specific extensions or specific for a pair of simulators [4,6,8,15].

In comparison with these existing works, the contributions of this paper consists in (1) proposing a generic model for an accurate continuous/discrete simulation (independent from languages and simulators); (2) providing implementation solutions in the case of SystemC and Simulink simulators; (3) introducing the CODIS tool which can automatically produce instances of the global simulation model; and (4) presenting an accuracy and performance analysis using an illustrative application.