Combining HIFSuite and SCNSL for smart device integration and simulation

Alessandro Danese\textsuperscript{2}, Franco Fummi\textsuperscript{1,2}, Valerio Guarneri\textsuperscript{2}, Michele Lora\textsuperscript{2}, Graziano Pravadelli\textsuperscript{1,2}, Francesco Stefanni\textsuperscript{1}

\begin{itemize}
  \item \textsuperscript{1}EDALab s.r.l., Italy, email addresses: name.surname@edalab.it
  \item \textsuperscript{2}University of Verona, Italy, email addresses: name.surname@univr.it
\end{itemize}

Introduction

The main design issue of smart devices is their high degree of heterogeneity, due to the simultaneous presence of multiple domains and extra-functional properties, together with the traditional system functionality. This makes design and simulation very challenging, even because heterogeneity implies that the functionality is not the only dimension that must be considered at validation time. Other properties, such as power consumption or thermal dissipation, are critical to ensure correctness of the final product and to correctly estimate its behavior. This makes component integration and simulation key phases in the design and verification process of smart devices. Thus, to efficiently master smart device design, it is fundamental to be aware of design issues and to know how to solve them through innovative tools and methods, which allow integrating all the components of a smart device into an efficient and flexible simulation platform. We addressed such issues by means of the combined use of HIFSuite tools and SCNSL to obtain a homogeneous and fast SystemC/C++ model of a smart device through the compositions of heterogeneous components. An Open Source Test Case (OSTC) has been defined to show the potentiality of the proposed methods and tools.

HIFSuite

HIFSuite (http://www.hifsuite.com/) is a set of tools to speed up the integration of heterogeneous components inside a virtual platform. The core of HIFSuite is the Heterogeneous Intermediate Format (HIF), an intermediate language able to capture the semantics of different Hardware Description Languages (HDLS) and different abstraction levels. By exploiting its application programming interface (API), many tools for manipulation and verification of HIF descriptions have been implemented.

HIFSuite mainly consists of the following tools:

- **Front-end translators**: for mapping HDL descriptions into HIF. Currently there are converters from Verilog, Verilog-AMS and VHDL to HIF.
- **Back-end tools**: for mapping HIF descriptions to actual languages. The hif2sc tool generates SystemC RTL, SystemC TLM, SystemC AMS, and C++ descriptions. Moreover, it allows to generate descriptions which use an optimized implementation of HDL data types (namely HDTLib).
- **IP-XACT translators**: for mapping an IP-XACT description to HIF, and vice versa.
- **\(A^T\)**: for abstracting HIF RTL level descriptions to HIF TLM (AT and LT) and C++ descriptions.
- **HR**: for flattening design hierarchies.
- **Analyzer**: for mapping Verilog-AMS models to SystemC-AMS descriptions.
- **DdT**: for replacing HDL data types with C++ built-in data types. Therefore, generated code runs faster, but it will not preserve details typical of HDL data types, such as multi-value logic.
- **treeComposer**: for assembling the final platform starting from a set of HIF models, representing platform components, and an IP-XACT description of their interconnections.

SCNSL

The SystemC Network Simulation Library (SCNSL – http://scnsl.sourceforge.net/) is a library that aims at extending SystemC capabilities by supporting network descriptions. SCNSL is a free and open source project released under the LGPL license. The main features and components are the following:

- **Designers** implement the functionalities inside RTL or TLM tasks. Tasks complexity can vary according to the designers’ objectives, but in complex virtual platforms they are usually implemented as a network interface connected to the other platform components.
- **Tasks** are associated with network nodes, which can communicate via channels. SCNSL provides implementation of various channel models, such as full duplex, half duplex, unidirectional and two models for wireless channels (namely shared and delayed-shared).
- **Network simulation** can be started quickly and easily customized since SCNSL provides many utility classes, such as network protocols, traffic generators, tracing utilities, environment models, etc.. Custom models can be easily implemented too.

The OSTC case study

The demonstrator relies on a case study, which is an open source test case (OSTC) implemented within the context of the FP7-ICT-288827 SMAC EU project: smart components and smart system integration (http://www.fp7-smac.org).

The OSTC represents a widely heterogeneous device, composed by aggregating IPs from different domains, such as digital hardware, analog and RF components, embedded software, MEMS, etc. (Fig. 1).

It is divided in functional and non-functional parts. The interconnection of the functional parts provides the description of the overall behavior of the platform, while non-functional components are used to model parameters such as energy and power consumption. The application implemented by the test case aims at exploiting completely the heterogeneity of components composing the platform. In particular, OSTC exchanges information with the surrounding world by sensing the external acceleration applied to the physical system and it communicates with a RF wireless component. Moreover, a serial interface is provided to program the system or to retrieve data from it. Possible use scenarios for this system are, for example, fall detection in the assistance of elderly people or impact detection in automotive applications.

The OSTC is used to show how heterogeneous components, available at different abstraction levels and for different domains, can be integrated to create a homogeneous SystemC/C++ model suited for efficient simulation at transactional level.

\begin{tabular}{|c|c|}
  \hline
  Verilog & Memory \\
  \hline
  SystemC & Controller \\
  \hline
  SystemC & UART \\
  \hline
  SystemC & XYAxis \\
  \hline
  SystemC & RF & Transceiver \\
  \hline
\end{tabular}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Fig1.png}
\caption{The SMAC OSTC case study.}
\end{figure}