PERFidiX: Performance Estimation and POSIX based RTOS Features for Software modeling in SystemC

Héctor Posadas[1], Eugenio Villar[1], Jesús A. Adámez[1] & Francisco Blasco[2]

[1]University of Cantabria, GIM, TEISA Dept.
E.T.S.I.I.T. Avda. Los Castros s/n, 39005 Santander, Spain
{posadash, villar}@teisa.unican.es

[2]DS2, Robert Darwin 2, Parque Tecnológico, Paterna, Spain
francisco.blasco@ds2.es

I. - INTRODUCTION

HW/SW systems development requires a design flow where both, HW and SW, must be considered together in order to allow refining the components of the system. One of the most important ways of simulating complex HW/SW systems is the use of high-level languages. Among them, SystemC is one of the most accepted languages in the designer community. However, the use of SystemC presents some limitations. The simulation of SW elements requires including the effects of a Real-Time Operating System (RTOS), and the HW platform both together. Thus, an extension of SystemC is necessary.

First, the execution of the refined SW code produces an untimed simulation. As a consequence, the system cannot be accurately co-simulated and performance estimations cannot be obtained. The effects of the target platform in the SW execution time are critical when modeling the whole system.

Secondly, SystemC does not directly support several features presented in typical RTOS. The SW refinement requires a model of the RTOS mechanisms for concurrency, scheduling, communication and synchronization. So the simulation framework has to include a RTOS specification that provides all the common capabilities in the standard operating systems.

II. - PERFIDIX

To overcome these limitations, a new tool called PERFidiX has been developed. This tool contains two modules: a library that works over the SystemC kernel, and an independent graphic interface.

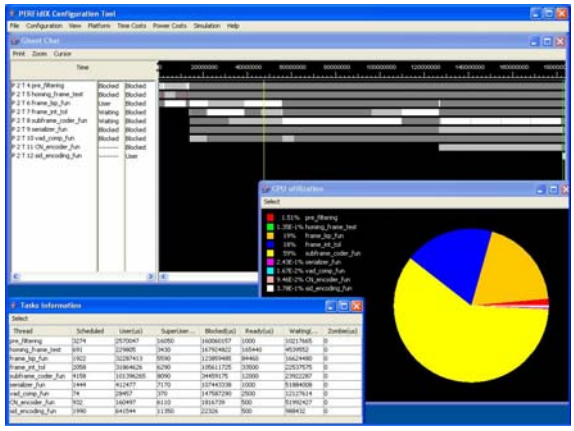
The SystemC library can be divided in two parts. First it automatically intruments the SW code to obtain execution time estimations. Dynamically, the library estimates the time cost of the SW that is been executed. After that, the estimated time is annotated at the correct points where required, not only at static predefined points. Thus, the untimed simulation is moved into a timed one.

Furthermore, the library models the features of a POSIX-compliant RTOS. Processes and threads can be scheduled using the POSIX defined priorities and policies through the POSIX standard API. Channels, as mutexes, semaphores or message queues, and POSIX signals can be also used to communicate and synchronize the system SW components.

To obtain a real, generic RTOS API, the POSIX standard has been used. Thus, the real, final SW code can be simulated accurately in the same framework where system-level models and HW components can be developed.

Summarizing, the library automatically generates timed simulations of SW components, allowing the use of the common RTOS facilities and features. As a consequence, the same code that will be introduced in the target platform can be adequately included in the system simulation.

Apart from that, a graphic interface has been also developed. This interface allows the designer to easily configure the library to model adequately the target platform and to analyze the simulation results.



III. - KEY PAPERS

[1]H. Posadas, F. Herrera, P. Sánchez, E. Villar and F. Blasco: "System-level performance analysis in SystemC", in Proceedings of the Design, Automation and Test Conference, IEEE, 2004.

[2]H. Posadas, E. Villar and F. Blasco: "Real-Time Operating System modeling in SystemC for HW/SW co-simulation", in Proceedings of DCIS, IST, Lisbon, 2005.

[3] H. Posadas, J. Adámez, P. Sánchez, E. Villar and F. Blasco: "POSIX modeling in SystemC", Proceedings of the Asia and South-Pacific Design Automation Conference, IEEE, 2006.

IV. – PERFIDIX WEB

PERFidiX will be available soon from:
http://www.teisa.unican.es/perfidix