EDA tools and methods required for formal system level representation

By Mario Diaz Nava, Laurent Maillet-Contoz STMicroelectronics and Adam Morawiec, ECSICorporation

Nowadays the electronic systems industry observes an ever increasing customer demand for innovative products integrating multiple heterogeneous functions. Such systems become more and more complex but can nevertheless be implemented due to the advances in nanometer technology enabling the integration of hundreds of millions of transistors in a single chip. However, the system and microelectronic companies, in order to remain competitive or in critical cases to survive, need to acquire and continuously expand their design capabilities and dispose of manifold expertise to design and integrate these complex systems – and this at the pressure of lowest possible cost in an extremely limited time frame.

In this context, companies mainly differentiate in their design capabilities based on designs tools, methods, standards and flows to provide unique efficient design services and competitive products.

Systems-on-Chip (SoCs) are contributing to the implementation of systems with ever increasing complexity. It is now commonly understood that traditional design techniques are no longer suited, and there is an obvious need for new design flows starting at system level.

System Level Design Challenges

As this is well known now, systems become more and more complex, due to several factors:

» Increase of the number of transistors integrated on one chip (Moore's Law).
» Increase of the heterogeneity of components, by the integration of RF, analog, digital blocks, and involving multi-processor systems in charge to run huge software stacks, co-operating to control hardware blocks, but also offering significant parts of the algorithmic part of the system.
» Difficulties for the integration of subsystems coming from different providers (internal first, but also third parties).
» Reuse of hardware and software components in different contexts.
» Testing the system implies (re)testing of all subsystems involved.

This creates new challenges in term of design flows, to master the complexity, increase the reliability of the system, while addressing time to market constraints as well. Whereas the design at RTL is now well understood and stable in terms of design processes, System Level Design (SDL) is still in its infancy. The techniques associated to this level provide lots of advantages, but are at the moment limited to experts, who have developed their own, and usually in-house expertise, methods and tools. SDL techniques still have to mature to be deployed widely in the industry by targeting several objectives:

» Moving one step forward in the definition of methods and tools to capture the overall system functionality, and provide efficient and proven means to refine the system representation.
» Offering formal methods to assess the validity of system properties at different levels of abstraction, and to provide means to check that refinement of the system has not introduced any issue, if the refinement itself can not be proven as correct-by-construction.
» Improving the high level synthesis methods to provide effective and general purpose tools, able to deal with regular C/C++ code without limiting themselves to a very constrained subset of these languages.
» Completing the standardization effort, in some areas standardization process has been already carried out. For example, the SPIRIT consortium defines the design and IP exchange formats, or the OSCI consortium defines standard memory-mapped bus APIs. Other areas need to be standardized addressing higher levels of abstraction to capture the overall functionality of the entire system, before hardware/

Newsletter edacentrum Probeauszug
BestellenSie sich den kompletten Artikel
über newsletter@edacentrum.de

edacentrum, Hannover, April 2009