Top-Down System Level Design Methodology Using SpecC, VCC and SystemC

Lukai Cai UC, Irvine
Paul Kritzinger ASA, Motorola
Mike Olivares ASA, Motorola
Daniel Gajski UC,Irvine

There appears to be an increasing trend towards the use of the C/C++ language as a basis for the next generation modeling tools and platform methodology to encompass design reuse. However, even with this convergence, industry is suffering the pain that there is no one tool or a complete tool flow methodology that can implement a top-down design methodology from C to silicon .

In this paper we suggest a top-down methodology from C to silicon. In our methodology, we focus on methods to make the design flow smooth, efficient, and easy. The proposed methodology is a pure top-down methodology. We developed our design methodology by using SpecC [1], VCC[2], and SystemC[3]. We choose SpecC, VCC and SystemC because they are all C-related and each have strong support in at least one field of design. Our proposal for a methodology is based on our experiences of attempting to model the JPEG encoder with SpecC, SystemC and VCC, and one internal project, attempting to implement architecture exploration for MPEG encoding and decoding using VCC.

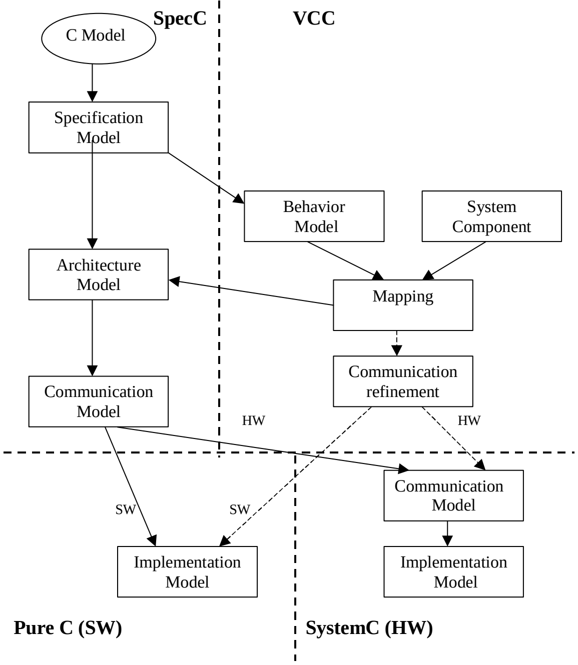

Figure 1 Top-Down Design Flow Using SpecC, VCC and SystemC.

As shown in Figure1, we propose combining the design flows of SpecC and VCC and adding SystemC as a back-end to either SpecC or VCC.

SpecC methodology is a top down methodology[1]. It provides four well-defined levels of abstraction (models) and a well-defined method for moving down these successive levels. Increasing architectural refinement takes place with each level [1]. In terms of assisting in this process, SpecC today provides a performance profiling tool at the specification level, and a model refinement tool to help in the conversion from specification level to architectural level. We use SpecC for behavior exploration (where a behavior is defined as a collection of functions) system modeling, and model refinement.

VCC[2] is a Behavior/Architecture co-design and design export tool. With VCC, the performance of design can be estimated after mapping specific behaviors to specific architectural components. Furthermore, VCC can be used to implement communication refinement. We use VCC for architecture exploration.

SystemC is a C++ class library that can be used to create a cycle-accurate model for software algorithms, hardware architectures, and interfaces, related to system-level designs[3]. SystemC's co-simulation and synthesis tool can help to generate RTL level design model from the behavior level, thus allowing completion of the design.

We see the potential of success for a SpecC to SystemC refinement methodology. Based on the result of architecture exploration from VCC, the SpecC specification model will be refined to a SpecC architecture model, using the SpecC refinement tools. By following SpecC methodology this architectural model will be refined to a SpecC communication model. The SpecC communication model is then translated into a SystemC communication model.

The SpecC to SystemC refinement and synthesis method is a language-based methodology. Thus, it is consistent and straightforward. IP can be modeled at different levels and IP reuse, IP exchange, and IP integration are all possible. Like SpecC, the SystemC language enables tools to have a common framework for interoperability. This allows users to utilize the best "point tool" solution for the implementation of the methodology. Since both languages use C as the underlying technology, interoperability can be achieved. This method quickly converts the C model to an implementation, resulting in decreased design cycle time.

Reference:

[1] D. Gajski, J. Zhu et al. "SpecC: Specification Lanugaeg and Design Methodology" Kluwer Academic Publishers, 2000
[2] Cadence, "VCC2.1 Production Documentation"
[3] www.systemc.org