Part II

C/C++-BASED SYSTEM DESIGN

Part II of this book addresses language-based modelling and design techniques for simulation, debugging, transformation, and analysis of hardware/software embedded systems. C/C++ based design methodologies are entering productive industrial design flows especially after the IEEE standardization of SystemC. Hence, the lion’s share of contributions uses SystemC and its extensions to illustrate the scientific approach. However, articles using languages like graphical (UML) or functional languages (e.g. Haskell) were presented, covering interoperability issues and heterogeneous models of computations. Other topics were embedded software modelling or technology specific approaches, e.g., for reconfigurable computing or signal processing platforms. Efficient mechanisms for abstraction and simulation like transaction level modelling (TLM), IP-based system design or system synthesis were also covered by this workshop.

In Part II, you will find six excellent articles representing the typical spectrum of contributions to this thematic area. Hopefully, the following overview helps you to pick your favourite articles.

In CHECKPOINT AND RESTORE FOR SYSTEMC MODELS, Monton et al. present an efficient technique called checkpointing to save and restore the state of a SystemC simulation model. Rather than saving the complete model state, their approach aims at saving only essential information needed to restore the state. Models that follow some coding guidelines can reliably be stored and restored with checkpoints that are orders of magnitude smaller than the full model.

The exploration of design alternatives for multi-processor SoCs was one of the main motivations for SystemC TLM based modelling. In their paper EFFICIENT APPROXIMATELY-TIMED PERFORMANCE MODELING FOR ARCHITECTURAL EXPLORATION OF MPSOCS, M. Streubühr et al. present a framework for modelling MPSoCs at Electronic System Level (ESL) where the mapping of the application and the platform model are configured at elaboration time. The benefits for faster design space exploration are shown in an example.

The memory subsystem has a huge impact on the performance of many embedded systems. The efficient analysis of embedded memories is challenging as it requires fast and correct models. Loeb and Sauer present in their article FAST SYSTEMC PERFORMANCE MODELS FOR THE EXPLORATION...
OF EMBEDDED MEMORIES an extension to the SystemC-based System-Click framework. It allows combining different memory models to accurately reflect the performance characteristics of embedded memory. In an MPSoC example they evaluate the technique and compare it with state-of-the-art techniques.

Most embedded systems are compositions of heterogeneous components which often require combining different models of computation to efficiently describe such complex systems. The article ANOTHER TAKE ON FUNCTIONAL SYSTEM LEVEL DESIGN AND MODELING from Toczek, Houzet, and Mancini proposes a Haskell-based approach relying on functional programming principles to face the challenges when designing very big and complex systems. In an application example they validate the applicability of their modelling technique.

Reconfigurable technologies are highly flexible but require new design methodologies as standard techniques cannot cope adequately with adaptivity. In GENERIC MODEL FOR APPLICATION-SPECIFIC PROCESSORS ON RECONFIGURABLE FABRIC, A. Yurdakul et al. present a framework containing a new model, a design language (LRH+) and a set of tools for the efficient automated design of runtime-reconfigurable processors. Based on an application model in LRH+, the presented framework allows for generating a specific instruction set to be implemented on a runtime-reconfigurable fabric, exploiting the full flexibility of this technology.

Most current C-based synthesis tools restrict the designer either to the OSCI SystemC Synthesis subset or to a sequential ANSI C/C++ model. In A SYSTEMC SUPERSET FOR HIGH-LEVEL SYNTHESIS, M. Smirnov and A. Takach propose to combine these modelling styles into a new concept. Their approach not only addresses high-level synthesis but also stepwise refinement and enables simulation at different levels of abstraction.