TRANSACTION LEVEL MODELING WITH SYSTEMC
Transaction Level Modeling with SystemC
TLM Concepts and Applications for Embedded Systems

Edited by

FRANK GHENASSIA
STMicroelectronics, France
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td>vii</td>
</tr>
<tr>
<td>Preface</td>
<td>xi</td>
</tr>
<tr>
<td>Special Contribution</td>
<td>xv</td>
</tr>
<tr>
<td>TLM: An Overview and Brief History</td>
<td>1</td>
</tr>
<tr>
<td>FRANK GHENASSIA AND ALAIN CLOUARD</td>
<td></td>
</tr>
<tr>
<td>Transaction Level Modeling</td>
<td>23</td>
</tr>
<tr>
<td>LAURENT MAILLET-CONTOZ AND FRANK GHENASSIA</td>
<td></td>
</tr>
<tr>
<td>TLM Modeling Techniques</td>
<td>57</td>
</tr>
<tr>
<td>LAURENT MAILLET-CONTOZ AND JEAN-PHILIPPE STRASSEN</td>
<td></td>
</tr>
<tr>
<td>Embedded Software Development</td>
<td>95</td>
</tr>
<tr>
<td>ERIC PAIRE</td>
<td></td>
</tr>
<tr>
<td>Functional Verification</td>
<td>153</td>
</tr>
<tr>
<td>THIBAUT BULTIAUX, STEPHANE GUENOT, SERGE HUSTIN,</td>
<td></td>
</tr>
<tr>
<td>ALEXANDRE BLAMPEY, JOSEPH BULONE, MATTHIEU MOY</td>
<td></td>
</tr>
<tr>
<td>Architecture Analysis and System Debugging</td>
<td>207</td>
</tr>
<tr>
<td>ANTOINE PERRIN AND GREGORY POIVRE</td>
<td></td>
</tr>
<tr>
<td>Design Automation</td>
<td>241</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Design Automation</td>
<td>241</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>267</td>
</tr>
<tr>
<td>Index</td>
<td>269</td>
</tr>
</tbody>
</table>
Foreword

System-on-Chip and TLM

A System-on-Chip (SoC) is a blend of software and silicon hardware components intended to perform a pre-defined set of functions in order to serve a given market. Examples are SoCs for cell phones, DVD players, ADSL line cards or WLAN transceivers. These functions have to be delivered to the target users as a SoC product during the right market window at satisfactory levels of performance and cost.

Over the past 20 years, the productivity of SoC designers has not been able to keep pace with Moore’s Law, which states that the silicon process technology allows doubling the number of transistors per chip every 18 or 24 months. Since the advent of RTL, designers and design automation engineers have searched for the next design methodology allowing a step function in design productivity.

Simply put, we believe that we have found and delivered to the industry the next SoC design methodology breakthrough: System-C TLM. This book is a vibrant testimony by the people who made it happen, giving both some details on the search for this Holy Grail, and the many facets of the applications of TLM.

The Search for SystemC TLM

Raising the level of CMOS digital design abstraction from gate-level and schematic capture to Register-Transfer-Level (RTL) has enabled a fundamental breakthrough in digital circuit design in the 1980s and 1990s. RTL’s clean separation between Boolean operations on signals, and clocks registering the results of these operations, was first embodied in the Verilog language initially designed by Phil Moorby in 1985; then in VHDL with the initial IEEE standard approved in 1987. RTL was first thought of as a more
efficient way to model digital designs. Soon, its wonderful formal characteristics allowed separating combinatorial logic optimization as demonstrated by MIS\textsuperscript{1}, from sequential elements such as registers or latches. In turn, complete synthesis tools emerged, as exemplified by Design Compiler from Synopsys.

Since RTL, many attempts have been made at identifying and defining the ‘next’ practical level of design abstraction. Of course, algorithm developers start out at a very abstract level, which is not tied to any architecture decision or implementation. What missing was an intermediate level, which would be abstract enough to allow complete system architecture definition while being accurate enough to allow performance analysis.

In 1999, a small motivated team of researchers from various fields at ST set out to design and verify a third-generation H263 video CODEC\textsuperscript{2}, architected with several dedicated heterogeneous processors as well as several hardware accelerators. As other SoC architects, they had to identify performance bottlenecks of the CODEC, while simultaneously defining and refining the micro-architecture of the hardware accelerators, the instruction set of the dedicated processors, and the embedded software performing control tasks and handshaking with the external world. On a previous incarnation of the CODEC, the designers had used extensive RTL-based verification methods, including hardware emulators, in order to verify the embedded software running on the selected micro-architecture with hundreds of reference image streams.

Every time a functional or performance issue requiring an architecture or micro-architecture change was encountered, a long re-design and re-verification cycle, spanning many weeks and sometimes months, would be necessary.

On the other hand, for the embedded software developer working with the processor architect, a modification requiring a change of the instruction set was almost immediate: a new Instruction Set Simulator (ISS) was generated and the embedded software could run very rapidly on the new ISS. The reason was that the processor was modeled in C as a functional model, and some wrapper code that represented the interface and communication to the processor peripherals.

During a project review the idea emerged that, using the same abstraction level as the ISS for other SOC hardware blocks would allow a breakthrough

\begin{footnotesize}
\begin{itemize}
\end{itemize}
\end{footnotesize}
in verification time. In itself, the idea of dissociating cleanly function and communication was not new, but the real breakthrough came from developing a framework for this modeling abstraction using an open and still evolving design modeling language: SystemC.

Using SystemC as a vehicle to provide the Transaction Level Modeling (TLM) abstraction proved to be the key to the fairly fast deployment of this methodology. There was no issue of proprietary language support by only one CAD vendor or university. There was also no issue of making a purchase decision by the design manager for yet another costly design tool.

Eventually, with the collaboration of ARM and Cadence Design Systems, a full-blown proposal was made to the Open SystemC Initiative (OSCI), under the name PV (Programmer View) and PVT (Programmer View Timed). Indeed ‘Programmer View’ clearly reflects the intent of this new abstraction level, which is to bridge the gap between the embedded software developer and the hardware architect.

**Paradigm Shift**

Not all the possible implications of sharing a single executable functional reference across the various teams have been explored yet.

Certainly, allowing the Algorithm, Hardware, Software and Functional Verification teams to rely on the same functional model is saving valuable time by avoiding misunderstandings due to informal or even formal paper-based communication.

However, we are also witnessing a real paradigm shift in the way software and hardware engineers work with each other. When an SD video movie can run at the rate of 1 image/second, equivalent to 12MHz, on an early model of the architecture, this allows SW development to start while the architecture is not yet frozen. Of course, earlier interactions between the hardware and software teams lead to better overall SoCs. Since more and more, delivering a prototype to the SOC customer is on the critical path of the application software development by that customer, TLM-based SoC platforms actually allow early application software development by the end customer before the actual hardware architecture is even frozen.

Next, a full ecosystem of system-level IP developers, both in-house and from third-party vendors, needs to develop. We are taking steps in raising the awareness level of the IP providers, so they start to include these TLM views as a standard part of their deliverables together with RTL models. Beyond this, we are making fast progress within the SPIRIT consortium, which will allow the SoC architect to mix and match IP blocks modeled in TLM, as system-level IP functional descriptions.

Philippe Magarshack
Crolles, April 18th 2005
Preface

Throughout the evolution of microelectronics industry, SoC designers have always been struggling to improve their productivity in order to fully exploit the growing number of transistors on a chip achievable by the silicon process capacity.

The answer to this challenge has always been increasing the level of abstraction used for the SoC implementation. From transistors to gates, and from gates to RTL, the design productivity has been maintained high enough to keep pace with and take advantage of the silicon technologies. Unfortunately, RTL as the design entry point cannot handle the complexity of 500 million-transistor SoCs designed with the CMOS90 process technology.

Two major directions are contributing to bridge the gap between design productivity and process capacity:

- Raise the level of abstraction to specify and model a SoC design.
- Adopt a different design paradigm, going from hardwired blocks to partially or fully programmable solutions, as pioneered by Paulin et al.

The proposed solution resolves critical system level issues encountered in designing a SoC and its associated embedded software. The brief history of our reaching TLM at STMicroelectronics is traced in Chapter 1.

---

TLM, an acronym for Transaction Level Modeling, has become an overloaded buzzword hiding too many different abstractions and modeling techniques. Applications of our TLM definition as described in Chapters 2 and 3, have proved to successfully tackle the following topics:

- **Productivity** through a veritable hardware/software co-development based on virtual prototypes, as described in Chapter 4.
- **First-time Silicon Success (FTSS)** achieved by using TLM as golden reference in the functional verification flow, which also enables a system-oriented verification, as described in Chapter 5. Ensuring the compliance of the SoC design with real-time constraints of the targeted application also contributes to FTSS, as discussed in Chapter 6.
- **Efficient workflow** between the numerous teams contributing to the development of the SoC and associated software. This is attainable by sharing a unique set of specification documents and models, as well as by keeping consistency between the various teams through platform automation tools, as described in Chapter 7.

This book is intended for engineers and managers who face challenges of designing SoCs in advanced CMOS technologies, and seek for solutions to enhance their current SoC and system level methodologies. It also serves engineers looking for SystemC modeling guidelines. More generally, we hope that this book will trigger new ideas in the research community to enhance design techniques based on Transaction Level Modeling.

**Acknowledgements**

This book is the result of five years of research and development work at STMicroelectronics. The authors of the chapters are the main architects of the resulting environment. I wish to thank all of them here for their key contributions. Our solution could not be industrialized without the great help of many engineers, especially knowing that “the devil is in the details” as the old saying goes. I am grateful to all of them for their great contributions: Aditya Raghunath, Adene Ben-Halima, Alain Kaufman, Amandeep Khurmi, Amit Mangla, Andrei Danes, Ankur Sareen, Anmol Sethy, Arnaud Richard, Ashish Trambadia, Bruno Galilée, Christophe Leclerc, Claude Helmstetter, Dinesh Kumar Garg, Dheeraj Kaushik, Dorsaf Fayech, Emmanuel Viaud, Etienne Lantreibeccq, Hervé Broquin, Jérôme Cornet, Jérôme Peillat, Julien Thevenon, Kamlesh Pathak, Kshitiz Jain, Laurent Bernard, Maher Hechkel, Mamta Bansal, Maxime Fiandino, Michel Bruant, Mukesh Chopra, Naveen Khajuria, Olivier Raout, Ramesh Mishra, Rohit Jindal, Sandeep Khurana, Stephane Maulet, Tran Nguyen, Vincent Motel, and Walid Joulak.

Many from the industry and the academia have guided us in the right direction. Special thanks go to Ahmed Jerraya, Chuck Pilkington, Florence
Maraninchi, Jean-Michel Fernandez, John Pierce, Joseph Borel, Laurent Ducousso, Marcello Coppola, Mark Burton, Michel Favre, Pierre Paulin, and Thorsten Grötker.

I also appreciate our employer for giving us ample freedom to propose and experiment new ideas, especially Philippe Magarshack, for his continuous support and encouragement.

Last but not least, I wish to acknowledge the special contribution of Jim-Moi Wong. Writing a book while employed in the industry is a great challenge because other urgent tasks consume most of the time. Despite such situation, her professionalism and dedication have smoothly implemented this book project; and finally, we are able to publish this book.

Frank Ghenassia
Crolles, May 2005
All of the material and technical information in this book are collected, organized, elaborated, structured, and drafted by Jim-Moi Wong through the very tight cooperation with the editor-in-chief, the co-authors, and the team of System Platform Group (SPG) at STMicroelectronics.