

Preface

We can trace the origin of this book back to the year 2003, when we first started working more closely together. On the one hand, Bart had been working on system on chip (SOC) debugging since joining Philips Research in 1997. He had architected with other colleagues a low-cost, core-based scan architecture to provide bit-level access to the state of a digital SOC via an IEEE 1149.1 test access port.

Two problems that still needed to be addressed however were (1) how to set appropriate execution breakpoints in a generic SOC, and (2) how to extract and interpret a consistent state from a globally-asynchronous locally-synchronous (GALS) SOC. Earlier analyses had shown that state data might become inconsistent on clock domain boundaries, when a GALS SOC is stopped at the clock cycle level.

Kees, on the other hand, had been working on real-time networks on chip (NoCs) for use in SOC's for Philips's advanced set-top boxes. Apart from designing the NoC itself, he worked on debugging NoC-based systems, together with other colleagues, by monitoring packets inside the NoC and reconstructing transactions.

We realized at this point that these two debugging problems may be solved by utilizing the communication protocol agreements between pairs of SOC building blocks. Through the use of protocols, we may raise the level of abstraction of both the on-chip breakpoints and of the interpretation of the SOC state. In addition, we envisioned that manipulating the transactions at the NoC boundary for debugging would be more intuitive, simpler, and precise.

For the next six years, we worked together with students to further explore and validate these concepts. A sizable number of publications were written to document our progress. In 2009, Bart started to write his PhD thesis on this subject matter, to bring together these two fields in a consistent manner and to document all our research results so far. The book in front of you is based on this thesis, which Bart successfully defended at the Eindhoven University of Technology in December 2013.

In this book, we first identify the key requirements for debugging a modern, silicon SOC implementation, and the technical factors that complicate this debugging task. We then propose our novel communication-centric, scan-based, abstraction based, run/stop-based (CSAR) debug approach with associated on-chip debug architecture, electronic design automation (EDA) tools, and off-chip debugger software that addresses these requirements and complicating factors. This debug infrastructure allows a debug engineer to localize faults in silicon SOC implementations, by

controlling its execution and observing the internal state of an SOC at different levels of abstraction. We subsequently evaluate the efficiency and effectiveness of aspects of our approach and its supporting infrastructure using six industrial SOC models and an illustrative, example SOC model. We also quantify the hardware cost and design effort to support our approach.

This book is generally intended for readers interested in SOC models. We have written it for senior architects and engineers, as well as academics (both teachers and students), who look to better understand the problems involved in debugging digital integrated circuits (ICs). New concepts are gradually introduced and illustrated. This book is especially well-suited for readers that intend to implement debug support in their own digital ICs. In particular:

Novice/student & teachers Learns/teaches about SOC design, verification, and validation trends and challenges, the need to address the debug problem for silicon SOC implementations and the fundamental problems in meeting the debug requirements in practice, existing options for supporting software and hardware debug by design, and illustrating and evaluating an approach, an on-chip hardware architecture, EDA tools, and an off-chip debugger software architecture to address these problems.

Practitioner Learns about concrete concepts, architectures, and implementations for supporting the debugging of SOC software and hardware by design. This includes detailed descriptions of the on-chip debug architecture, the required EDA tools, and the off-chip debugger software.

Expert Learns about a comprehensive debug approach for modern, silicon SOC implementations with its associated debug infrastructure, which is directly usable with different SOC implementations, and provides comprehensive details on hardware and software architecture implementation details and choices therein that usually are not published.

We could not have performed the work described in this book without the support of the management of Philips Research Laboratories and NXP Semiconductors. We have collaborated with many colleagues at Philips Research Laboratories and NXP Semiconductors, and thank them all for the support they have given over the years. We also gratefully acknowledge the standardization efforts by colleagues in the Philips Debug Action Group and Core Test Action Group, and the efforts on the proprietary InCiDE software. Many thanks also go to the design teams of the SOC models described in Chap. 8. They allowed the validation of our debug theories, described in this book, in practice on their SOC models. We also had the pleasure to work together with university employees and students on debug research as well. We thank them for their debug contributions.

Last but certainly not least, we express our deepest thanks to our families and friends for their many years of support and for making the journey that led to this book both possible and so much more enjoyable.

Eindhoven,
March 2014

Bart Vermeulen
Kees Goossens

Debugging Systems-on-Chip
Communication-centric and Abstraction-based
Techniques

Vermeulen, B.; Goossens, K.

2014, XV, 311 p. 127 illus., 7 illus. in color., Hardcover

ISBN: 978-3-319-06241-9