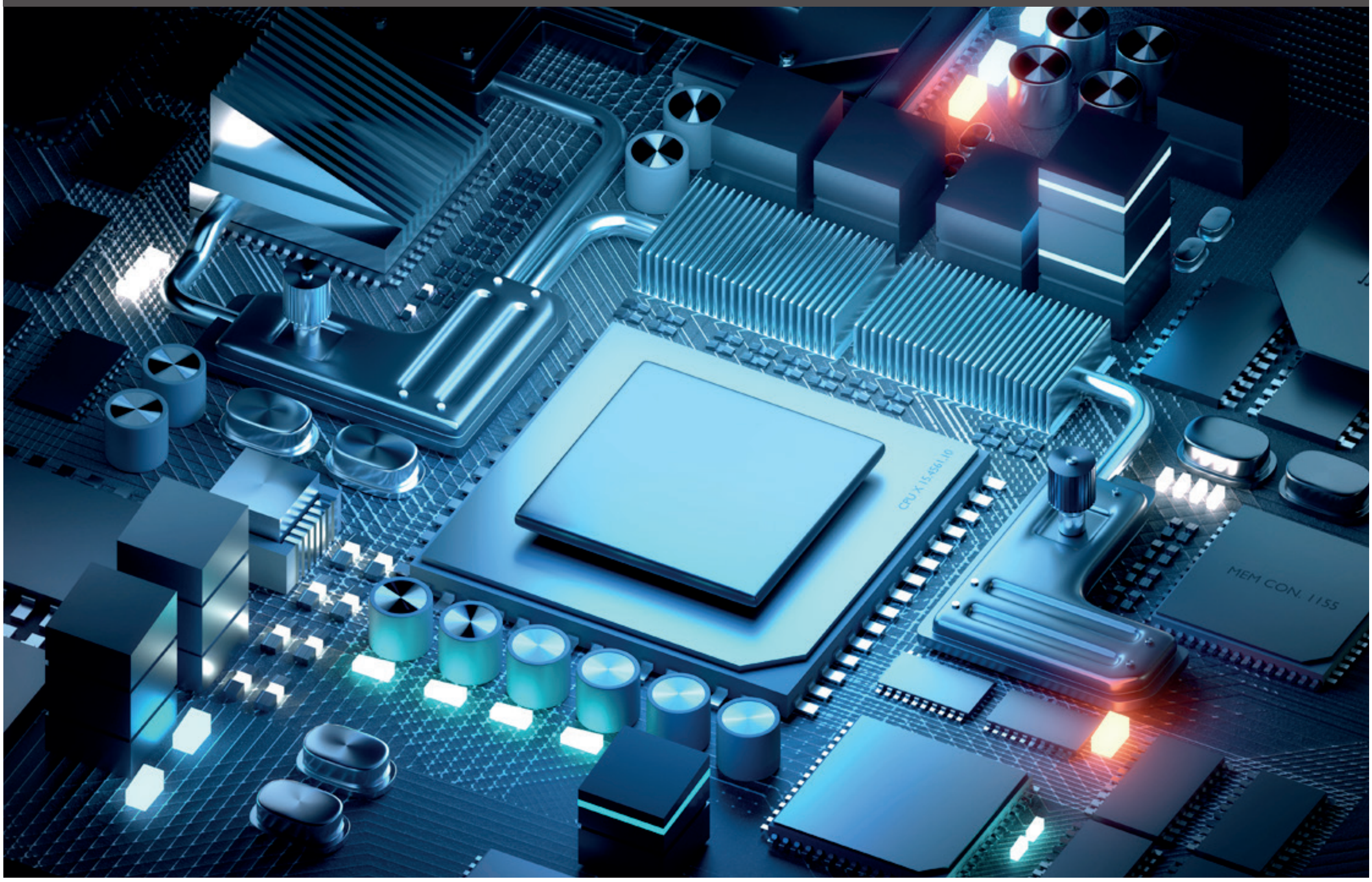
arm Education Media

Modern System-on-Chip Design on Arm

TEXTBOOK

David J. Greaves

SoC Design



Modern System-on-Chip Design on Arm

DAVID J. GREAVES

arm Education Media

Arm Education Media is an imprint of Arm Limited, 110 Fulbourn Road, Cambridge, CBI 9NJ, UK

Arm is committed to making the language we use inclusive, meaningful, and respectful. Our goal is to remove and replace non-inclusive language from our vocabulary to reflect our values and represent our global ecosystem.

Arm is working actively with our partners, standards bodies, and the wider ecosystem to adopt a consistent approach to the use of inclusive language and to eradicate and replace offensive terms. We recognise that this will take time. This book contains references to non-inclusive language; it will be updated with newer terms as those terms are agreed and ratified with the wider community.

Contact us at education@arm.com with questions or comments about this course. You can also report non-inclusive and offensive terminology usage in Arm content at terms@arm.com.

ISBN: 978-1-911531-37-1
978-1-911531-38-8 (ePub)
978-1-911531-36-4 (print)

Version: ePDF

For information on all Arm Education Media publications, visit our website at
https://www.arm.com/resources/education/books

To report errors or send feedback, please email edumedia@arm.com

Contents

Preface

Silicon technology has seen relentless advances in the past 50 years, driven by constant innovation and miniaturisation. As a result, more and more functionality has been placed into a single chip. Today, entire systems, including processors, memory, sensors and analogue circuitry, are integrated into one single chip (hence, a system-on-chip or SoC), delivering increased performance despite tight area, power and energy budgets. The aim of this textbook is to expose aspiring and practising SoC designers to the fundamentals and latest developments in SoC design and technologies. The processors within a SoC run a huge body of software. Much of this code is portable over many platforms, but low-level components, such as device drivers, are hardware-dependent and may be CPU-intensive. Power use can be reduced using custom accelerator hardware. Although this book emphasises the hardware design elements, it also addresses *co-design*, in which the hardware and software are designed hand in hand. It is assumed that the reader already understands the basics of processor architecture, computer technology, and software and hardware design.

Is This Book Suitable For You?

We assume that you have some experience with hardware design using an RTL such as Verilog or VHDL, and that you understand assembly language programming and basic principles of operating systems. In other words, you have completed the first two years of a degree in Computer Science or Electronic Engineering.

Many of the principles taught in this book are relevant for all forms of system architect, including those who are designing cloud-scale applications, custom accelerators or IoT devices in general, or those making FPGA designs. But the details of design verification in Chapter 8 are likely to be just of interest to those designing semi-custom silicon using standard cells.

A Git repository of **online additional material** is available at http://bitbucket.org/djg11/modern-soc-design-djg

This contains data used for generating tables and graphs in the book, as well as further source code, lab materials, examples and answers to selected exercises.

The repo contains a SystemC model of the Zynq super FPGA device family, coded in blocking TLM style. It is sufficient to run an Arm A9 Linux kernel using an identical boot image as the real silicon.

Book Structure

This book contains nine chapters, each devoted to a different aspect of SoC design.

Chapter 1 reviews basic computer architecture, defining terms that are used in later chapters. Readers are expected to be largely familiar with most of this material, although the transactional-level modelling (TLM) view of the hardware is likely to be new. A SoC is an assembly of intellectual property (IP) blocks.

Chapter 2 describes many of the standard IP blocks that make up a typical SoC, including processors, memories, input/output devices and interrupts.

Chapter 3 considers the interconnect between the IP blocks, covering the evolution of processor busses and networks-on-chip (NoCs).

Chapter 4 teaches basic principles of system architecture, including dimensioning of busses and queuing theory and arbitration policies. It also discusses debug support.

Chapter 5 presents Electronic System Level (ESL) modelling, where a simulation model for a whole SoC, also known as a virtual platform, is put together. The ESL model is sufficient to test and develop software, as well as to perform architectural exploration, where the throughput, energy use and silicon area of a proposed system implementation can be examined at a high level.

Chapter 6 presents further architectural exploration considerations, including the design of custom accelerators for a specific application. The languages Bluespec and Chisel are described as alternatives to RTL for design entry and the basic principles of high-level synthesis (HLS) are covered.

Chapter 7 is a primer for formal verification of SoCs, comparing the usefulness of formal compared with simulation for bug hunting and right-first-time solutions. A number of useful formal tricks are covered.

Chapter 8 presents semi-custom fabrication flows for making the physical silicon and covers advanced verification and variability mitigation techniques for today's deep sub-micron devices using FinFETs.

Chapter 9 covers what to do when the first SoC samples arrive back from the wafer processing plant, including booting an operating system and checking environmental compatibility (operating temperature and unwanted radio emissions).

Acknowledgements

I am very grateful to Professor Sir Andy Hopper, who was my PhD supervisor, who has been a constant source of inspiration and direction, and who has often been my boss both in industry and at the Computer Laboratory. He introduced me to the field of chip design. I am also very grateful to the late M. G. Scroggie, the principal author of 'Foundations of Wireless', which was a book I read and re-read all through my childhood. I can only hope some people find this current book as valuable as I found his. Certainly I have tried to mix breadth and depth in the same accessible way that he managed. I would like to thank those working in the Computer Laboratory who helped with this book, including David Chisnall, Robert Mullins, Omer Sella and Milos Puzovic. I would also like to thank my wife, Aldyth, for putting up with me for this last year. I've often read such comments in the acknowledgement sections of other books, but now I understand what causes it.

Most importantly, I'd like to thank the many Arm staff who have helped with this book, either by contributing text to large chunks of it, or with additional information and suggestions:

Khaled Benkrid, who made this book possible.

Liz Warman, who kept me on track and assisted me with the process.

Shuojin Hang and Francisca Tan who helped create the scope and reviewed early drafts.

This book would not have been possible without the collaboration of the following Arm engineers who have co-written, reviewed and commented on the book:

- Chapter 2: Processors, Memory and IP Blocks
  Rahul Mathur, Staff Engineer

- Chapter 3: SoC Interconnects
  Anup Gangwar, Distinguished Engineer
  Antony Harris, Senior Principal AMBA Architect

- Chapter 6: Architectural Design Exploration
  Edwin Dankert, Director, Technology Management

- Chapter 7: Formal Methods and Assertion-Based Design
  Daryl Stewart, Distinguished Engineer

- Chapter 8: Fabrication and Production
  Jim Dodrill, Fellow
  Christophe Lopez, Senior Principal Engineer
  Aurelien Merour, Principal Engineer
  Jean-Luc Pelloie, Fellow

Author Biography

Dr. David J. Greaves, PhD CEng. is Senior Lecturer in Computing Science at the University of Cambridge, UK and a Fellow of Corpus Christi College. Born in London, he grew up in a house full of engineering textbooks, circuit diagrams and pieces of telecommunications equipment. His grandfather had built his own television set as soon as television broadcasts started. His father worked at EMI and IBM, developing modems and computer interfaces. With the shift of head office of IBM UK to Portsmouth, the family moved to Romsey in Hampshire.

Plessey Roke Manor was also situated in Romsey, along with IBM's UK research laboratories at Hursley Park. These were, and remain, world-leading research centres in the field of radio communications and computing. The young David J. Greaves was a regular visitor and intern at both sites, and by the age of 17 had designed and built his first computer. The chips had been mostly removed from old circuit boards using a blow lamp. The software, including the disk operating system and a Pascal compiler, had all been written from scratch.

During his A-level studies, Greaves designed a local area network for Commodore PET computers. The design was published in Wireless World magazine and commercially replicated.

As an undergraduate at St John's College Cambridge, he offered consultancy services to various small electronics companies in the field of professional audio, developing MIDI interfaces and low-noise pre-amplifiers. His final-year degree project was a fully digital keyboard instrument that was serialised in Wireless World and copied by many enthusiasts worldwide. A main interest became the design and implementation of compilers, as encouraged by Dr. Martin Richards of St Johns, who had developed the BCPL language, the direct precursor of C.

Greaves designed his first silicon chips during his PhD studies, which were in the field of metropolitan area networks. He designed fibre optic transceivers that sent the first mono-mode signals over the newly installed fibres that criss-crossed Cambridge. In collaboration with Acorn Computer, in 1995 Greaves was the chief network architect for the Cambridge ITV trial, which put ATM switches in the kerbside cabinets belonging to Cambridge Cable Ltd and delivered video on demand to 50 or so homes. It was 20 years later that the last movie rental shop in Cambridge closed.

Also in 1995, he implemented CSYN, one of the first Verilog compilers for synthesising hardware specifically for field programmable gate arrays. This compiler was distributed widely among local companies on the Cambridge Science Park and also used for undergraduate teaching. It was licensed to a multinational to bundle with its own family of FPGAs.

Greaves had visited Arm when it first spun out of Acorn and consisted of six engineers in a barn. At the university, Greaves used a donation of Arm circuit boards for a new practical course in which the students wrote assembly language and Verilog to learn about low-level hardware and software interfacing. These courses still run today and the lecture notes have evolved into this textbook.

Greaves has been on the board or technical advisory board of at least ten start-up companies. He has supervised or examined more than 60 PhD students. He holds at least five international patents in the field of communications and electronics. His company Tenison EDA was, before acquisition, directly providing tools to all major chip makers. His current research interests remain in the field of compilation tools for design automation and scientific acceleration.

Modern System-on-Chip Design on Arm

SoC design has seen significant advances in the decade and Arm-based silicon has often been at the heart of this revolution. Today, entire systems including processors, memories, sensors and analogue circuitry are all integrated into one single chip (hence "System-on-Chip" or SoC). The aim of this textbook is to expose aspiring and practising SoC designers to the fundamentals and latest developments in SoC design and technologies using examples of Arm® Cortex®-A technology and related IP blocks and interfaces. The entire SoC design process is discussed in detail, from memory and interconnects through to validation, fabrication and production. A particular highlight of this textbook is the focus on energy efficient SoC design, and the extensive supplementary materials which include a SystemC model of a Zynq chip.

This textbook is aimed at final year undergraduate students, master students or engineers in the field looking to update their knowledge. It is assumed that readers will have a pre-existing understanding of RTL, Assembly Language and Operating Systems. For those readers looking for a entry-level introduction to SoC design, we recommend *our Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers* textbook.

Table of contents

Dr. David J. Greaves, PhD CEng. is a Senior Lecturer in Computing Science at the University of Cambridge, UK and a Fellow of Corpus Christi College.

In 1995, Dr Greaves implemented CSYN, one of the first Verilog compilers for synthesising hardware specifically for field programmable gate arrays. This compiler was distributed widely among local companies on the Cambridge Science Park and also used for undergraduate teaching. It was licensed to a multinational to bundle with its own family of FPGAs.

Dr Greaves holds at least five international patents in the field of communications and electronics. His current research interests remain in the field of compilation tools for design automation and scientific acceleration.

arm Education Media

Arm Education Media is a publishing operation within Arm Ltd, providing a range of educational materials for aspiring and practicing engineers. For more information, visit: arm.com/resources/education

ISBN 978-1-911531-37-1