Digital Design

A Systems Approach

This introductory textbook provides students with a system-level perspective and the tools they need to understand, analyze, and design digital systems. It goes beyond the design of simple combinational and sequential modules to show how such modules are used to build complete systems.

- All the essential topics needed to understand modern design practice are covered, including:
  - Design and analysis of combinational and sequential modules
  - Composition of combinational and sequential modules
  - Data and control partitioning
  - Factoring and composition of finite-state machines
  - Interface specification
  - System timing
  - Synchronization
- Teaches how to write Verilog HDL in a productive and maintainable style that enables CAD tools to do much of the tedious work.
- Covers the fundamentals of logic design, describing an efficient method to design combinational logic and state machines both manually and using modern CAD tools.

A complete introduction to digital design is given through accurate, clear explanations, extensive examples, and online Verilog files. The teaching package is completed with lecture slides, labs, and a solutions manual for instructors. Assuming no previous digital knowledge, this textbook is ideal for undergraduate digital design courses that will prepare students for modern digital practice.

William J. Dally is the Willard R. and Inez Kerr Bell Professor of Engineering at Stanford University and Chief Scientist at NVIDIA Corporation. He and his group have developed system architecture, network architecture, signaling, routing, and synchronization technology that can be found in most large parallel computers today. He has many years of experience working in industry and academia, previously holding positions at Bell Labs, Caltech and MIT, and consulting for Digital Equipment, Cray Research and Intel. He is a Member of the National Academy of Engineering, a Fellow of the IEEE, a Fellow of the ACM, and a Fellow of the American Academy of Arts and Sciences. He has received numerous honors, including the ACM Eckert–Mauchly Award, the IEEE Seymour Cray Award, and the ACM Maurice Wilkes Award. He has published over 200 papers in these areas, holds over 75 issued patents, and is an author of the textbooks *Digital Systems Engineering* and *Principles and Practices of Interconnection Networks*.

R. Curtis Harting is a Ph.D. candidate at Stanford University. He graduated with honors in 2007 from Duke University with a B.S.E., majoring in Electrical & Computer Engineering and Computer Science. He received his M.S. in 2009 from Stanford University. His primary research interest is in computer architecture, focusing on parallel, high-performance, and energy-efficient design.

‘Dally and Harting blend circuit and architecture design in a clear and constructive manner on the basis of their exceptional experience in digital design.

‘Students will discover a modern and effective way to understand the fundamental underpinning of digital design, by being exposed to the different abstraction levels and views of computing systems.’

Giovanni De Micheli, *EPFL Switzerland*

‘Bill and Curt have combined decades of academic and industry experience to produce a textbook that teaches digital system design from a very practical perspective without sacrificing the theoretical understanding needed to train tomorrow’s engineers. Their approach pushes students to understand not just what they are designing but also what they are building. By presenting key advanced topics, such as synthesis, delay and logical effort, and synchronization, at the introductory level, this book is in the rare position of providing both practical advice and deep understanding. In doing so, this book will prepare students well even as technology, tools, and techniques change in the future.’

David Black-Schaffer, *Uppsala University*

‘Everything you would expect from a book on digital design from Prof. Dally. Decades of practical experience are distilled to provide the tools necessary to design and compose complete digital systems. A clear and well written text that covers the basics and system-level issues equally well. An ideal starting point for the microprocessor and SoC designers of the future!’

Robert Mullins, *University of Cambridge and the Raspberry Pi Foundation*

‘This textbook sets a new standard for how digital system design is taught to undergraduates. The practical approach and concrete examples provides a solid foundation for anyone who wants to understand or design modern complex digital systems.’

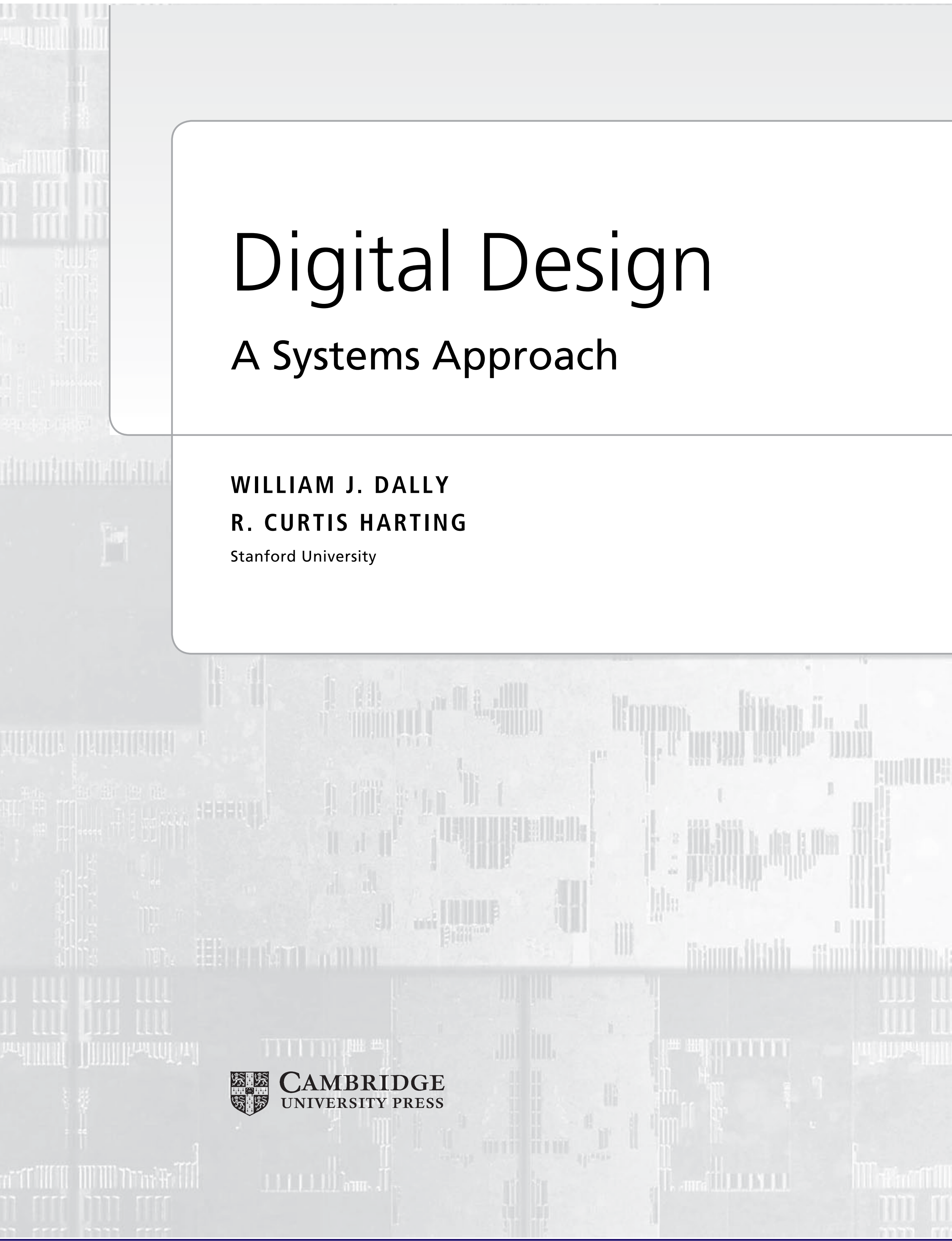
Steve Keckler, *The University of Texas at Austin*

‘This book not only teaches how to do digital design, but more importantly shows how to do *good* design. It stresses the importance of modularization with clean interfaces, and the importance of producing digital artifacts that not only meet their specifications, but which can also be easily understood by others. It uses an aptly-chosen set of examples and the Verilog code used to implement them.

‘It includes a section on the design of asynchronous logic, a topic that is likely to become increasingly important as energy consumption becomes a primary concern in digital systems.

‘The final appendix on Verilog coding style is particularly useful. This book will be valuable not only to students, but to practitioners in the area. I recommend it highly.’

Chuck Thacker, *Microsoft*



Digital Design

A Systems Approach

WILLIAM J. DALLY
R. CURTIS HARTING
Stanford University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Mexico City

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521199506

First published 2012

Printed in the United States of America

A catalog record for this publication is available from the British Library

Library of Congress Cataloging in Publication data
Dally, William J.
Digital design : a systems approach / WILLIAM J. DALLY, R. CURTIS HARTING.
pages cm
Includes bibliographical references and index.
ISBN 978-0-521-19950-6 (hardback)
1. Digital electronics. 2. Electronic systems. I. Harting, R. Curtis. II. Title.
TK7868.D5D328 2012
621.39′1 – dc23 2012021831

ISBN 978-0-521-19950-6 Hardback

Additional resources for this publication at www.cambridge.org/9780521199506

CONTENTS

PREFACE

This book is intended to teach an undergraduate student to understand and design digital *systems*. It teaches the skills needed for current industrial digital system design using a hardware description language (Verilog) and modern CAD tools. Particular attention is paid to system-level issues including factoring and partitioning digital systems, interface design, and interface timing. Topics needed for a deep understanding of digital circuits, such as timing analysis, metastability, and synchronization, are also covered. Of course, we cover the manual design of combinational and sequential logic circuits. However, we do not dwell on these topics because there is far more to digital system design than designing such simple modules.

Upon completion of a course using this book, students should be prepared to practice digital design in industry. They will lack experience, but they will have all of the tools they need for contemporary practice of this noble art. The experience will come with time.

This book has grown out of more than 25 years of teaching digital design to undergraduates (CS181 at Caltech, 6.004 at MIT, and EE121 and EE108A at Stanford). It is also motivated by 35 years of experience designing digital systems in industry (Bell Labs, Digital Equipment, Cray, Avici, Velio Communications, Stream Processors, and NVIDIA). It combines these two experiences to teach what students need to know to function in industry in a manner that has been proven to work on generations of students.

We wrote this book because we were unable to find a book that covered the system-level aspects of digital design. The vast majority of textbooks on this topic teach the manual design of combinational and sequential logic circuits and stop. While most texts today use a hardware description language, the vast majority teach a TTL-esqe design style that, while appropriate in the era of 7400 quad NAND gate parts (the 1970s), does not prepare a student to work on the design of a three billion transistor GPU. Today's students need to understand how to factor a state machine, partition a design, and construct an interface with correct timing. We cover these topics in a simple way that conveys insight without getting bogged down in details.

Outline of the book

A flow chart showing the organization of the book and the dependences between chapters is shown in Figure 1. The book is divided into an introduction, five main sections, and chapters about style and verification.

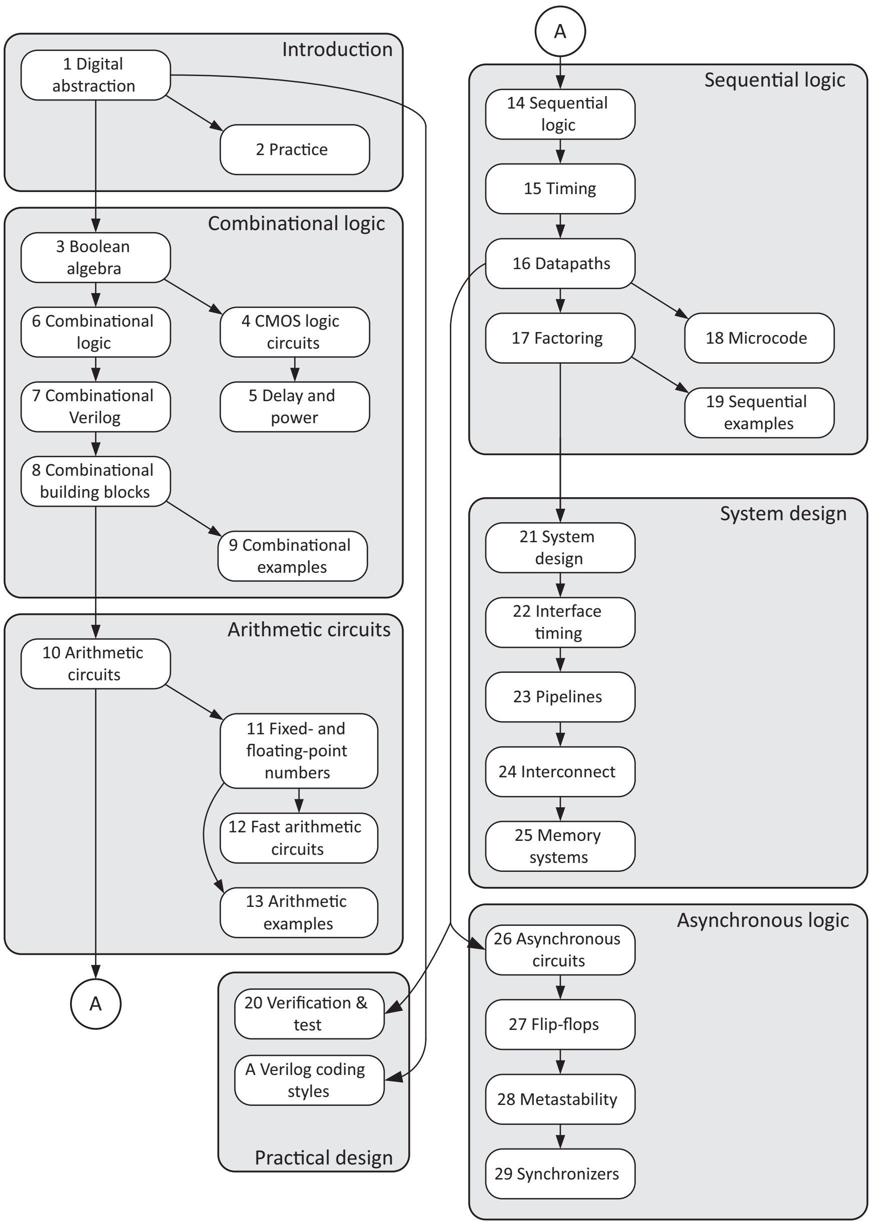


Figure 1. Organization of the book and dependence between chapters.

Part I Introduction

Chapter 1 introduces digital systems. It covers the representation of information as digital signals, noise margins, and the role of digital logic in the modern world. The practice of digital design in industry is described in Chapter 2. This includes the design process, modern implementation technologies, computer-aided design tools, and Moore's law.

Part II Combinational logic

Chapters 3–9 deal with combinational logic circuits – digital circuits whose outputs depend only on the current values of their inputs. Boolean algebra, the theoretical underpinning of logic design, is discussed in Chapter 3. Switching logic and CMOS gate circuits are introduced in Chapter 4. Chapter 5 introduces simple models for calculating the delay and power of CMOS circuits. Manual methods for designing combinational circuits from basic gates are described in Chapter 6. Chapter 7 discusses how to automate the design process by coding behavioral descriptions of combinational logic in the Verilog hardware description language. Building blocks for combinational logic, decoders, multiplexers, etc. are described in Chapter 8, and several examples of combinational design are given in Chapter 9.

Part III Arithmetic circuits

Chapters 10 –13 describe number systems and arithmetic circuits. Chapter 10 describes the basics of number representation and arithmetic circuits that perform the *four functions* $+$, $-$, $\times$, and $\div$ on integers. Fixed-point and floating-point number representations and their accuracy are presented in Chapter 11. This chapter includes a discussion of floating-point unit design. Techniques for building fast arithmetic circuits including carry look-ahead, Wallace trees, and Booth recoding are described in Chapter 12. Finally, examples of arithmetic circuits and systems are presented in Chapter 13.

Part IV Synchronous sequential logic

Chapters 14–19 describe synchronous sequential logic circuits – sequential circuits whose state changes only on clock edges – and the process of designing finite-state machines. After describing the basics in Chapter 14, timing constraints are covered in Chapter 15. The design of *datapath* sequential circuits – whose behavior is described by an equation rather than a state table – is the topic of Chapter 16. Chapter 17 describes how to factor complex state machines into several smaller, simpler state machines. The concept of stored program control, and how to build finite-state machines using microcoded engines, is described in Chapter 18. This section closes with a number of examples in Chapter 19.

Part V Practical design

Chapter 20 and the Appendix discuss two important aspects of working on digital design projects. The process of verifying the correctness of logic and testing that it works after manufacturing are the topics of Chapter 20. The Appendix teaches the student proper Verilog coding style. It is a style that is readable, maintainable, and enables CAD tools to produce optimized hardware. Students should read this chapter before, during, and after writing their own Verilog.

Part VI System design

Chapters 21–25 discuss system design and introduce a systematic method for the design and analysis of digital systems. A six-step process for system design is introduced in

Chapter 21. System-level timing and conventions for the timing of interfaces are discussed in Chapter 22. Chapter 23 describes pipelining of modules and systems, and includes several example pipelines. System interconnects including buses, crossbar switches, and networks are described in Chapter 24. A discussion of memory systems is given in Chapter 25.

Part VII Asynchronous logic

Chapters 26–29 discuss asynchronous sequential circuits – circuits whose state changes in response to any input change, without waiting for a clock edge. The basics of asynchronous design including flow-table analysis and synthesis and the problem of races are introduced in Chapter 26. Chapter 27 gives an example of these techniques, analyzing flip-flops and latches as asynchronous circuits. The problem of *metastability* and synchronization failure is described in Chapter 28. This section, and the book, closes with a discussion of synchronizer design – how to design circuits that safely move signals across asynchronous boundaries – in Chapter 29.

Teaching using this book

This book is suitable for use in a one-quarter (10-week) or one-semester (13-week) introductory course on digital systems design. It can also be used as the primary text of a second, advanced, course on digital systems.

There need not be any formal prerequisites for a course using this book. A good understanding of high-school level mathematics is the only required preparation. Except for Chapters 5 and 28 , the only place derivatives are used, the material does not require a knowledge of calculus. At Stanford, E40 (Introduction to Electrical Engineering) is a prerequisite for EE108A (Digital Systems I), but students often take EE108A without the prerequisite with no problems.

A one-quarter introductory course on digital systems design covers the material in Chapters 1, 3, 6, 7, 8, 10, (11), 14, 15, 16, (17), 21, 22, (23), 26, 28, and 29. For the one-quarter course we omit the details of CMOS circuits (Chapters 4 and 5), microcode (Chapter 18), and the more advanced systems topics (Chapters 24 and 25). The three chapters in parentheses are optional and can be skipped to give a slightly slower-paced course. In offerings of this course at Stanford, we typically administer two midterms: one after covering Chapter 11, and the second after covering Chapter 22.

A one-semester introductory course on digital systems can use the three additional weeks to include the material on CMOS circuits and a few of the more advanced systems topics. A typical semester-long course covers the material in Chapters 1, 2, 3, 4, (5), 6, 7, 8, 9, 10, (11), 13, 14, 15, 16, (17), (18), (19), 21, 22, (23), (24), (25), 26, (27), 28, 29.

This book can be used for an advanced course on digital systems design. Such a course covers the material from the introductory courses in more depth and includes advanced topics that were omitted from the introductory courses. Such a course usually includes a significant student project.

Materials

To support teaching with this book, the course website includes teaching materials including: lecture slides, a series of laboratories, and solutions to selected exercises. The laboratories are intended to reinforce the material in the course and can be performed via simulation or a combination of simulation and implementation on FPGAs.

ACKNOWLEDGMENTS

We are deeply indebted to the many people who have made contributions to the creation of this book. This book has evolved over many years of teaching digital design at MIT (6.004) and Stanford (EE108A). We thank the many generations of students who took early versions of this class and provided feedback that led to constant refinement of our approach. Professors Subhasish Mitra, Phil Levis, and My Le have taught at Stanford using early versions of this material, and have provided valuable comments that led to many improvements. The course and book benefited from contributions by many great teaching assistants over the years. Paul Hartke, David Black-Shaffer, Frank Nothaft, and David Schneider deserve special thanks. Frank also deserves thanks for contributing to the exercise solutions. Teaching 6.004 at MIT with Gill Pratt, Greg Papadopolous, Steve Ward, Bert Halstead, and Anant Agarwal helped develop the approach to teaching digital design that is captured in this book.

Julie Lancashire and Kerry Cahill at Cambridge University Press have helped us throughout the project. We thank Irene Pizzie for careful copy editing and Abigail Jones for shepherding the book through the sometimes difficult passage from manuscript to finished project.

Finally, our families: Sharon, Jenny, Katie, and Liza Dally, and Jacki Armiak, Eric Harting, and Susanna Temkin have offered tremendous support and made significant sacrifices so we could have time to devote to writing.