

The background of the top half of the cover is a high-magnification, top-down view of a silicon microchip. It shows a dense, repeating grid of square dies, each containing intricate circuit patterns. The colors are primarily shades of blue, purple, and brown.A small, white, rectangular label is affixed to the top left corner of the cover, partially obscuring the microchip pattern.

# TOP-DOWN DIGITAL VLSI DESIGN

From Architectures  
to Gate-Level Circuits and FPGAs

**MK**  
MORGAN KAUFMANN

**Hubert Kaeslin**

# Top-Down Digital VLSI Design

From Architectures  
to Gate-Level Circuits and FPGAs

**Hubert Kaeslin**

Microelectronics Design Center  
Department of Information Technology and Electrical Engineering  
ETH Zürich  
Switzerland



AMSTERDAM • BOSTON • HEIDELBERG • LONDON  
NEW YORK • OXFORD • PARIS • SAN DIEGO  
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier



Acquiring Editor: Stephen Merken  
Editorial Project Manager: Nate McFadden  
Project Manager: Poulouse Joseph  
Cover Designer: Maria Inês Cruz

Morgan Kaufmann is an imprint of Elsevier  
225 Wyman Street, Waltham, MA 02451, USA

Copyright © 2015 Published by Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: [www.elsevier.com/permissions](http://www.elsevier.com/permissions).

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

### Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

### Library of Congress Cataloging-in-Publication Data

Kaeslin, Hubert, author.

Top-down digital VLSI design : from architectures to gate-level circuits and FPGAS / Hubert Kaeslin,  
Microelectronics Design Center, Dept. of Information Technology and Electrical Engineering,  
ETH Zurich, Switzerland.

pages cm

ISBN 978-0-12-800730-3

1. Digital integrated circuits--Design and construction. 2. Integrated circuits--Very large scale  
integration--Design and construction. I. Title.

TK7874.65.K336 2015

621.39'5--dc23

2014035133

### British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN: 978-0-12-800730-3

For information on all MK publications  
visit our website at [www.mkp.com](http://www.mkp.com)



Working together  
to grow libraries in  
developing countries

[www.elsevier.com](http://www.elsevier.com) • [www.bookaid.org](http://www.bookaid.org)

# Top-Down Digital VLSI Design





# Preface

---

## WHY THIS BOOK?

Designing integrated electronics has become a multidisciplinary enterprise that involves solving problems from fields as disparate as

- Hardware architecture
- Software engineering
- Marketing and investment
- Semiconductor physics
- Systems engineering and verification
- Circuit design
- Discrete mathematics
- Layout design
- Electronic design automation
- Hardware test equipment and measurement techniques

Covering all these subjects is clearly beyond the scope of this text and also beyond the author's proficiency. Yet, I have made an attempt to collect material from the above fields that I have found to be relevant for making major design decisions and for carrying out the actual engineering work when developing Very Large Scale Integration (VLSI) circuits.

The present volume covers front-end design, that is all steps required to turn a software model into a gate-level netlist or, alternatively, into a bit stream for configuring field-programmable logic devices. A second volume on back-end design may follow at a later date.

The text has been written with two audiences in mind. As a textbook, it wants to introduce engineering students to the beauty and the challenges of digital VLSI design while preventing them from repeating mistakes that others have made before. Practising electronics engineers should find it appealing as a reference book because of the many tables, checklists, diagrams, and case studies intended to help them not to overlook important action items and alternative options when planning to develop their own circuits.

What sets this book apart from others in the field is its top-down approach. Beginning with hardware architectures, rather than with solid-state physics, naturally follows the normal VLSI design flow and makes the material more accessible to readers with a background in systems engineering, information technology, digital signal processing, or management.

---

## HIGHLIGHTS

- Top-down approach.
- Systematic overview on architecture optimization techniques.
- A chapter on field-programmable logic devices, their technologies and architectures.
- Key concepts behind both VHDL and SystemVerilog without too many syntactical details.
- A proven naming convention for signals and variables.
- Introduction to assertion-based verification.
- Concepts for re-usable simulation testbenches.
- Emphasis on synchronous design and HDL code portability.
- Comprehensive discussion of clocking disciplines.
- Largely self-contained (required previous knowledge summarized in two appendices).
- Emphasis on knowledge likely to remain useful in the years to come.
- Plenty of detailed illustrations.
- Checklists, hints and warnings for various situations.
- A concept proven in classroom teaching and actual design projects.

---

## NOTES TO INSTRUCTORS

Over the past decade, the capabilities of Field-Programmable Gate Arrays (FPGA) have grown to a point where they compete with custom-fabricated ICs in many electronic designs, especially for products marketed by small and medium enterprises.

Beginning with the higher levels of abstraction enables instructors to focus on those topics that are equally relevant irrespective of whether a design eventually gets implemented as “mask-programmed” chip or from components that are configured electrically. That material is collected in chapters 1 to 6 of the book and best taught as part of a Bachelor program for maximum dissemination. No prior introduction to semiconductor physics or devices is required. For audiences with little exposure to digital logic and finite state machines, the material can always be complemented with appendices A and B.

Chapter 7 is the only one to have a clear orientation towards mask-programmed circuits as clocking and clock distribution are largely pre-defined in field-programmable logic devices. As opposed to this, the material on synchronization in chapter 8 is equally important to FPGA users and to persons specializing in full- or semi-custom design.

Probably the best way of preparing for an engineering career in the electronics and microelectronics industry is to complete a design project where circuits are not just being modeled and simulated on a computer but actually fabricated and tested. At ETH Zürich, students are given this opportunity as part of a three-semester course given by the author and his colleagues, see figure below. The 6th term covers front-end design. Building a circuit of modest size with an FPGA is practiced in a series of exercises. Provided they come up with a meaningful proposal, students then get accepted for a much more substantial project that runs in parallel with their regular lectures and exercises during the 7th term.

Typically working in teams of two, students are expected to devote at least half of their working time to that project. Following tapeout at the end of the term, chip fabrication via an external multi-project wafer service roughly takes three months. Circuit samples then get systematically tested by their very developers in the 8th and final term. Needless to say that students accepting this offer feel very motivated and that industry highly values the practical experience of graduates formed in this way.

Most chapters in this book come with student problems. Some of them expose ideas left aside in the main text for the sake of conciseness. Problems are graded as a function of the effort required to solve them.

- \* A few thoughts lead to a brief answer.
- \*\* Details need to be worked out, count between 20 min and 90 min.
- \*\*\* A small engineering project, multiple solutions may exist. Access to EDA and other computer tools may help.

Solutions and presentation slides are available to instructors who register with the publisher from the book's companion **website** <http://booksite.elsevier.com/9780128007303>.

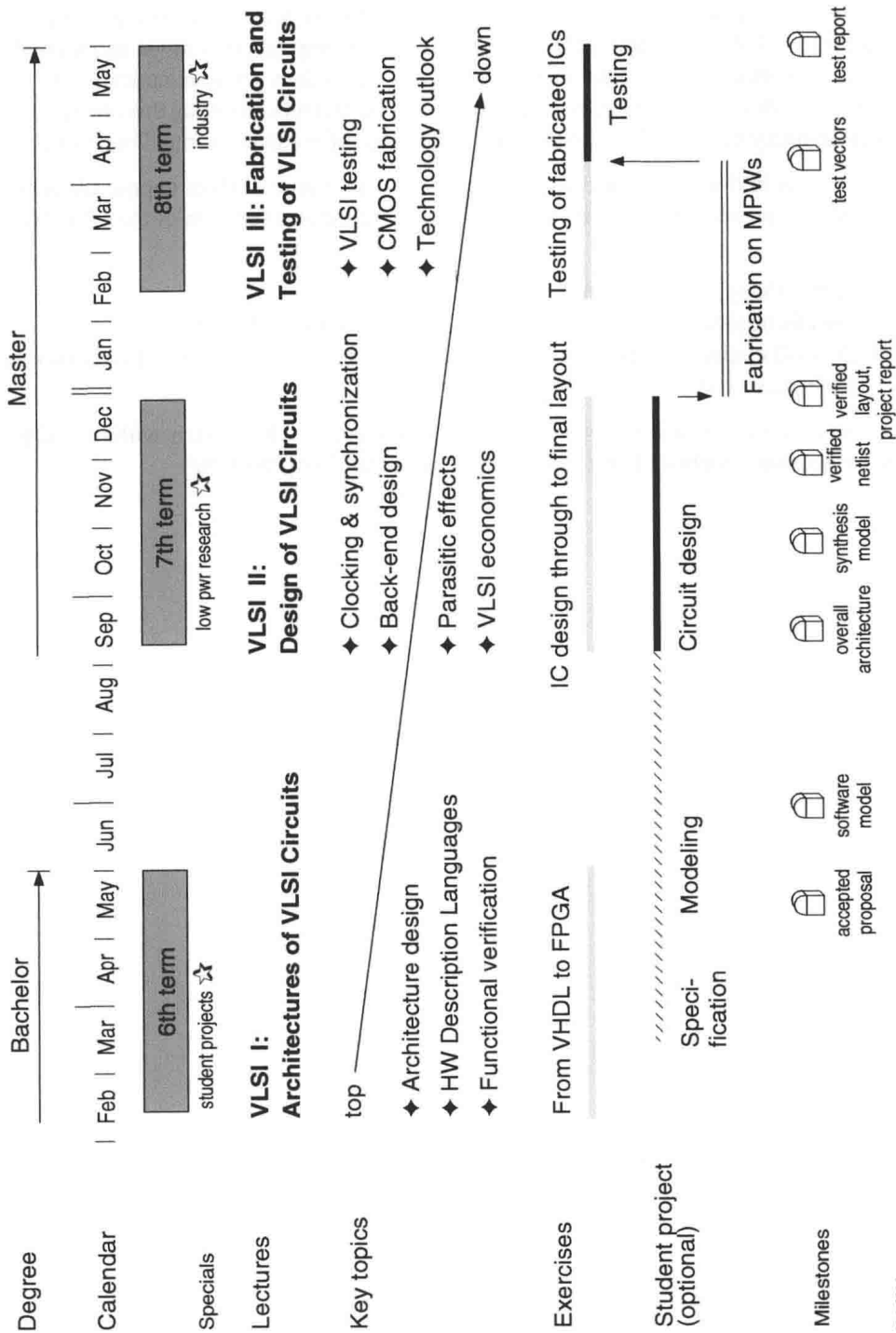


## Syllabus of ETH Zurich in

**Digital VLSI Design and Test**

Microelectronics Design Center  
Prof. Hubert Kaeslin

Integrated Systems Laboratory  
Dr. Norbert Felber



# Acknowledgments

This text collects the insight and the experience that many persons have accumulated over more than twenty years. While I was fortunate enough to author the book, I am indebted to all those who have been willing to share their expertise with me.

My thanks thus go to many past and present colleagues of mine including Christoph Balmer, David Bellasi, Prof. Andreas Burg, Dr. Felix Bürgin, Dr. Norbert Felber, Prof. em. Wolfgang Fichtner, Michael Gautschi, Dr. Pierre Greisen, Dr. Frank Gürkaynak, Christoph Keller, Prof. Mathieu Luisier, Dr. Patrick Mächler, Beat Muheim, Michael Mühlberghuber, Michael Schaffner, Prof. Christoph Studer, Prof. Jürgen Wassner, Dr. Markus Wenk, Prof. Paul Zbinden, and Dr. Reto Zimmermann. As long-time VHDL users, our staff and me are grateful to Dr. Christoph Sühnel who made us become fluent in SystemVerilog with as few detours and misunderstandings as possible. Most of these experts have contributed examples, have reviewed parts of my manuscript, or have otherwise helped improve its quality. Still, the only person to blame for all errors and other shortcomings that have remained in the text is me.

Next, I would like to extend my sincere thanks to all students who have followed our courses on Digital VLSI Design and Test. Not only their comments and questions, but also results and data from many of their projects have found their way into this text. Sebastian Axmann deserves special credit for helping with the solutions on a voluntary basis.

Giving students the opportunity to design microchips, to have them fabricated, and to test physical samples is a rather onerous undertaking that would clearly have been impossible without the continuous funding by ETH Zürich. Let me express our gratitude for that on behalf of all our graduates.

In cooperation with Christoph Wicki and his IT support team, the staff of the local Microelectronics Design Center does a superb job in setting up and maintaining the EDA infrastructure and the services indispensable for VLSI design in spite of the frequent landslides caused by rapid technological evolution and by unforeseeable business changes. I am particularly grateful to them for occasionally filling all sorts of gaps in my technical knowledge without making me feel too badly about it.

I am further indebted to Todd Green, Nate McFadden, Poulouse Joseph, and many more members of the staff at Morgan Kaufmann Publishers for their support with turning my LaTeX manuscript into a printed book. Finally, I would like to thank all persons and organizations who have taken their time to answer my reprint requests and who have granted me the right to reproduce illustrations of theirs.



# Contents

Preface .....	xv
Acknowledgments .....	xix
<b>CHAPTER 1 Introduction to Microelectronics.....</b>	<b>1</b>
1.1 Economic Impact .....	1
1.2 Microelectronics Viewed From Different Perspectives .....	4
1.2.1 The Guinness Book of Records Point of View .....	4
1.2.2 The Marketing Point of View .....	5
1.2.3 The Fabrication Point of View .....	7
1.2.4 The Design Engineer's Point of View .....	11
1.2.5 The Business Point of View .....	18
1.3 The VLSI Design Flow .....	20
1.3.1 The Y-chart, a Map of Digital Electronic Systems .....	20
1.3.2 Major Stages in Digital VLSI Design .....	22
1.3.3 Cell Libraries .....	31
1.3.4 Electronic Design Automation Software .....	32
1.4 Problems .....	34
1.5 Appendix I: A Brief Glossary of Logic Families .....	35
1.6 Appendix II: An Illustrated Glossary of Circuit-Related Terms.....	37
<b>CHAPTER 2 Field-Programmable Logic .....</b>	<b>41</b>
2.1 General Idea .....	41
2.2 Configuration Technologies .....	43
2.2.1 Static Memory .....	43
2.2.2 Flash Memory .....	44
2.2.3 Antifuses .....	44
2.3 Organization of Hardware Resources.....	46
2.3.1 Simple Programmable Logic Devices (SPLD) .....	46
2.3.2 Complex Programmable Logic Devices (CPLD) .....	47
2.3.3 Field-Programmable Gate Arrays (FPGA) .....	47
2.4 Commercial Aspects.....	53
2.4.1 An Overview on FPL Device Families .....	53
2.4.2 The Price and the Benefits of Electrical Configurability.....	53
2.5 Extensions of the Basic Idea .....	55
2.6 The FPL Design Flow .....	59
2.7 Conclusions .....	61

<b>CHAPTER 3 From Algorithms to Architectures</b>	<b>63</b>
<b>3.1 The Goals of Architecture Design</b>	63
3.1.1 Agenda	63
<b>3.2 The Architectural Solution Space</b>	65
3.2.1 The Antipodes	65
3.2.2 What Makes an Algorithm Suitable for a Dedicated VLSI Architecture?	71
3.2.3 There is Plenty of Land Between the Antipodes	74
3.2.4 Assemblies of General-Purpose and Dedicated Processing Units	75
3.2.5 Host Computer with Helper Engines	76
3.2.6 Application-Specific Instruction set Processors	77
3.2.7 Reconfigurable Computing	79
3.2.8 Extendable Instruction set Processors	81
3.2.9 Platform ICs (DSPP)	82
3.2.10 Digest	83
<b>3.3 Dedicated VLSI Architectures and How to Design Them</b>	87
3.3.1 There is Room for Remodeling in the Algorithmic Domain ...	87
3.3.2 ... and there is Room in the Architectural Domain	89
3.3.3 Systems Engineers and VLSI Designers Must Collaborate	90
3.3.4 A Graph-Based Formalism for Describing Processing Algorithms	92
3.3.5 The Isomorphic Architecture	94
3.3.6 Relative Merits of Architectural Alternatives	95
3.3.7 Computation Cycle Versus Clock Period	97
<b>3.4 Equivalence Transforms for Combinational Computations</b>	98
3.4.1 Common Assumptions	99
3.4.2 Iterative Decomposition	100
3.4.3 Pipelining	103
3.4.4 Replication	108
3.4.5 Time Sharing	110
3.4.6 Associativity Transform	116
3.4.7 Other Algebraic Transforms	117
3.4.8 Digest	118
<b>3.5 Options for Temporary Storage of Data</b>	120
3.5.1 Data Access Patterns	120
3.5.2 Available Memory Configurations and Area Occupation	120
3.5.3 Storage Capacities	121
3.5.4 Wiring and the Costs of Going Off-Chip	122
3.5.5 Latency and Timing	122
3.5.6 Digest	123

<b>3.6</b>	Equivalence Transforms for Non-Recursive Computations .....	126
3.6.1	Retiming .....	126
3.6.2	Pipelining Revisited .....	127
3.6.3	Systolic Conversion .....	130
3.6.4	Iterative Decomposition and Time Sharing Revisited .....	130
3.6.5	Replication Revisited .....	131
3.6.6	Digest .....	132
<b>3.7</b>	Equivalence Transforms for Recursive Computations .....	133
3.7.1	The Feedback Bottleneck .....	133
3.7.2	Unfolding of First-Order Loops.....	134
3.7.3	Higher-Order Loops .....	137
3.7.4	Time-Variant Loops .....	139
3.7.5	Nonlinear or General Loops.....	140
3.7.6	Pipeline Interleaving, not Quite an Equivalence Transform .....	144
3.7.7	Digest .....	146
<b>3.8</b>	Generalizations of the Transform Approach .....	148
3.8.1	Generalization to other Levels of Detail .....	148
3.8.2	Bit-Serial Architectures .....	150
3.8.3	Distributed Arithmetic.....	152
3.8.4	Generalization to other Algebraic Structures.....	155
3.8.5	Digest .....	160
<b>3.9</b>	Conclusions .....	160
3.9.1	Summary .....	160
3.9.2	The Grand Architectural Alternatives from an Energy Point of View ....	163
3.9.3	A Guide to Evaluating Architectural Alternatives .....	165
<b>3.10</b>	Problems .....	168
<b>3.11</b>	Appendix I: A Brief Glossary of Algebraic Structures .....	170
<b>3.12</b>	Appendix II: Area and Delay Figures of VLSI Subfunctions .....	174
<b>CHAPTER 4</b>	<b>Circuit Modeling with Hardware Description Languages .....</b>	<b>179</b>
<b>4.1</b>	Motivation and Background.....	179
4.1.1	Why Hardware Synthesis? .....	179
4.1.2	Agenda .....	179
4.1.3	Alternatives for Modeling Digital Hardware .....	180
4.1.4	The Genesis of VHDL and SystemVerilog .....	180
4.1.5	Why Bother Learning Hardware Description Languages? .....	182
4.1.6	A First Look at VHDL and SystemVerilog.....	184
<b>4.2</b>	Key Concepts and Constructs of VHDL.....	186
4.2.1	Circuit Hierarchy and Connectivity.....	186
4.2.2	Interacting Concurrent Processes .....	190



4.2.3	A Discrete Replacement for Electrical Signals.....	198
4.2.4	An Event-Driven Scheme of Execution .....	206
4.2.5	Facilities for Model Parametrization.....	218
4.2.6	Concepts Borrowed from Programming Languages .....	226
<b>4.3</b>	<b>Key Concepts and Constructs of SystemVerilog .....</b>	<b>234</b>
4.3.1	Circuit Hierarchy and Connectivity.....	234
4.3.2	Interacting Concurrent Processes .....	237
4.3.3	A Discrete Replacement for Electrical Signals.....	243
4.3.4	An Event-Driven Scheme of Execution .....	249
4.3.5	Facilities for Model Parametrization.....	254
4.3.6	Concepts Borrowed from Programming Languages .....	257
<b>4.4</b>	<b>Automatic Circuit Synthesis From HDL Models .....</b>	<b>262</b>
4.4.1	Synthesis Overview.....	262
4.4.2	Data Types .....	263
4.4.3	Finite State Machines and Sequential Subcircuits in General .....	263
4.4.4	RAM and ROM Macrocells .....	272
4.4.5	Timing Constraints .....	275
4.4.6	Limitations and Caveats .....	281
4.4.7	How to Establish a Register Transfer Level Model Step by Step .....	282
<b>4.5</b>	<b>Conclusions .....</b>	<b>284</b>
<b>4.6</b>	<b>Problems .....</b>	<b>286</b>
<b>4.7</b>	<b>Appendix I: VHDL and SystemVerilog Side by Side.....</b>	<b>289</b>
<b>4.8</b>	<b>Appendix II: VHDL Extensions and Standards .....</b>	<b>293</b>
4.8.1	Protected Shared Variables IEEE 1076a.....	293
4.8.2	The Analog and Mixed-Signal Extension IEEE 1076.1 .....	294
4.8.3	Mathematical Packages for Real and Complex Numbers IEEE 1076.2 ..	296
4.8.4	The Arithmetic Packages IEEE 1076.3 .....	296
4.8.5	The Standard Delay Format (SDF) IEEE 1497 .....	297
4.8.6	A Handy Compilation of Type Conversion Functions .....	298
4.8.7	Coding Guidelines .....	300
<b>CHAPTER 5</b>	<b>Functional Verification .....</b>	<b>301</b>
<b>5.1</b>	<b>Goals of Design Verification .....</b>	<b>301</b>
5.1.1	Agenda .....	302
<b>5.2</b>	<b>How to Establish Valid Functional Specifications .....</b>	<b>303</b>
5.2.1	Formal Specification.....	304
5.2.2	Rapid Prototyping .....	304
5.2.3	Hardware-Assisted Verification .....	306
<b>5.3</b>	<b>Preparing Effective Simulation and Test Vectors .....</b>	<b>307</b>
5.3.1	A First Glimpse at VLSI Testing .....	307
5.3.2	Fully Automated Response Checking is a Must .....	308

5.3.3	Assertion-Based Verification Checks from Within.....	310
5.3.4	Exhaustive Verification Remains an Elusive Goal .....	315
5.3.5	Directed Verification is Indispensable but has its Limitations.....	316
5.3.6	Directed Random Verification Guards Against Human Omissions .....	322
5.3.7	Statistical Coverage Analysis Produces Meaningful Metrics.....	327
5.3.8	Collecting Test Cases from Multiple Sources Helps .....	328
5.3.9	Separating Test Development from Circuit Design Helps .....	329
<b>5.4</b>	<b>Consistency and Efficiency Considerations .....</b>	<b>331</b>
5.4.1	A Coherent Schedule for Simulation and Test .....	332
5.4.2	Protocol Adapters Help Reconcile Different Views on Data and Latency .....	335
5.4.3	Calculating High-Level Figures of Merit.....	338
5.4.4	Patterning Simulation Set-Ups after the Target System.....	339
5.4.5	Initialization.....	340
5.4.6	Trimming Run Times by Skipping Redundant Simulation Sequences....	340
<b>5.5</b>	<b>Testbench Coding and HDL Simulation.....</b>	<b>341</b>
5.5.1	Modularity and Reuse are the Keys to Testbench Design .....	341
5.5.2	Anatomy of a File-Based Testbench .....	342
<b>5.6</b>	<b>Conclusions .....</b>	<b>346</b>
<b>5.7</b>	<b>Problems .....</b>	<b>347</b>
<b>5.8</b>	<b>Appendix I: Formal Approaches to Functional Verification .....</b>	<b>350</b>
<b>5.9</b>	<b>Appendix II: Deriving a Coherent Schedule for Simulation and Test.....</b>	<b>352</b>
<b>CHAPTER 6</b>	<b>The Case for Synchronous Design .....</b>	<b>357</b>
<b>6.1</b>	<b>Introduction .....</b>	<b>357</b>
<b>6.2</b>	<b>The Grand Alternatives for Regulating State Changes .....</b>	<b>359</b>
6.2.1	Synchronous Clocking .....	360
6.2.2	Asynchronous Clocking .....	360
6.2.3	Self-Timed Clocking .....	361
<b>6.3</b>	<b>Why a Rigorous Approach to Clocking is Essential in VLSI .....</b>	<b>364</b>
6.3.1	The Perils of Hazards.....	364
6.3.2	The Pros and Cons of Synchronous Clocking.....	365
6.3.3	Clock-as-Clock-Can is not an Option in VLSI.....	367
6.3.4	Fully Self-Timed Clocking is not Normally an Option Either.....	368
6.3.5	Hybrid Approaches to System Clocking .....	368
<b>6.4</b>	<b>The Dos and Dons of Synchronous Circuit Design .....</b>	<b>370</b>
6.4.1	First Guiding Principle: Dissociate Signal Classes! .....	370
6.4.2	Second Guiding Principle: Allow for Circuits to Settle Before Clocking! .....	371
6.4.3	Synchronous Design Rules at a More Detailed Level .....	372
<b>6.5</b>	<b>Conclusions .....</b>	<b>380</b>

6.6	Problems .....	381
6.7	Appendix: On Identifying Signals .....	382
6.7.1	Signal Class .....	382
6.7.2	Active Level .....	384
6.7.3	Signaling Waveforms .....	385
6.7.4	Three-State Capability .....	386
6.7.5	Inputs, Outputs and Bidirectional Ports .....	386
6.7.6	Present State vs. Next State .....	387
6.7.7	Signal Naming Convention Syntax .....	387
6.7.8	Usage of Upper and Lower Case Letters in HDL Source Code .....	388
6.7.9	A Note on the Portability of Names Across EDA Platforms .....	389
<b>CHAPTER 7</b>	<b>Clocking of Synchronous Circuits .....</b>	<b>391</b>
7.1	What is the Difficulty With Clock Distribution? .....	391
7.1.1	Agenda .....	392
7.1.2	Timing Quantities Related to Clock Distribution .....	393
7.2	How Much Skew and Jitter Does a Circuit Tolerate? .....	394
7.2.1	Basics .....	394
7.2.2	Single-Edge-Triggered One-Phase Clocking .....	396
7.2.3	Dual-Edge-Triggered One-Phase Clocking .....	402
7.2.4	Symmetric Level-Sensitive Two-Phase Clocking .....	404
7.2.5	Unsymmetric Level-Sensitive Two-Phase Clocking .....	407
7.2.6	Single-Wire Level-Sensitive Two-Phase Clocking .....	411
7.2.7	Level-Sensitive One-Phase Clocking and Wave Pipelining .....	412
7.3	How to Keep Clock Skew Within Tight Bounds .....	416
7.3.1	Clock Waveforms .....	416
7.3.2	Collective Clock Buffers .....	417
7.3.3	Distributed Clock Buffer Trees .....	419
7.3.4	Hybrid Clock Distribution Networks .....	421
7.3.5	Clock Skew Analysis .....	421
7.4	How to Achieve Friendly Input/Output Timing .....	423
7.4.1	Friendly as Opposed to Unfriendly I/O Timing .....	423
7.4.2	Impact of Clock Distribution Delay on I/O Timing .....	424
7.4.3	Impact of PTV Variations on I/O Timing .....	426
7.4.4	Registered Inputs and Outputs .....	427
7.4.5	Adding Artificial Contamination Delay to Data Inputs .....	427
7.4.6	Driving Input Registers From an Early Clock .....	428
7.4.7	Clock Tapped From Slowest Component in Clock Domain .....	428
7.4.8	“Zero-Delay” Clock Distribution by Way of a DLL or PLL .....	429
7.5	How to Implement Clock Gating Properly .....	432
7.5.1	Traditional Feedback-Type Registers with Enable .....	432
7.5.2	A Crude and Unsafe Approach to Clock Gating .....	433