Static Timing Analysis for Nanometer Designs

A Practical Approach

J. Bhasker • Rakesh Chadha

Static Timing Analysis for Nanometer Designs
A Practical Approach

Springer

J. Bhasker
eSilicon Corporation
1605 N. Cedar Crest Blvd.
Suite 615
Allentown, PA 18103, USA
jbhasker@esilicon.com

Rakesh Chadha
eSilicon Corporation
890 Mountain Ave
New Providence, NJ 07974, USA
rchadha@esilicon.com

ISBN 978-0-387-93819-6 e-ISBN 978-0-387-93820-2
DOI: 10.1007/978-0-387-93820-2

Library of Congress Control Number: 2009921502

© Springer Science+Business Media, LLC 2009
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden. The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.
While the advice and information in this book are believed to be true and accurate at the date of going to press, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Some material reprinted from "IEEE Std. 1497-2001, IEEE Standard for Standard Delay Format (SDF) for the Electronic Design Process; IEEE Std. 1364-2001, IEEE Standard Verilog Hardware Description Language; IEEE Std.1481-1999, IEEE Standard for Integrated Circuit (IC) Delay and Power Calculation System", with permission from IEEE. The IEEE disclaims any responsibility or liability resulting from the placement and use in the described manner.
Liberty format specification and SDC format specification described in this text are copyright Synopsys Inc. and are reprinted as per the Synopsys open-source license agreement.
Timing reports are reported using PrimeTime which are copyright © <2007> Synopsys, Inc. Used with permission. Synopsys & PrimeTime are registered trademarks of Synopsys, Inc.
Appendices on SDF and SPEF have been reprinted from "The Exchange Format Handbook" with permission from Star Galaxy Publishing.

Printed on acid-free paper.

springer.com

Preface

Timing, timing, timing! That is the main concern of a digital designer charged with designing a semiconductor chip. What is it, how is it described, and how does one verify it? The design team of a large digital design may spend months architecting and iterating the design to achieve the required timing target. Besides functional verification, the timing closure is the major milestone which dictates when a chip can be released to the semiconductor foundry for fabrication. This book addresses the timing verification using static timing analysis for nanometer designs.

The book has originated from many years of our working in the area of timing verification for complex nanometer designs. We have come across many design engineers trying to learn the background and various aspects of static timing analysis. Unfortunately, there is no book currently available that can be used by a working engineer to get acquainted with the details of static timing analysis. The chip designers lack a central reference for information on timing, that covers the basics to the advanced timing verification procedures and techniques.

The purpose of this book is to provide a reference for both beginners as well as professionals working in the area of static timing analysis. The book

is intended to provide a blend of the underlying theoretical background as well as in-depth coverage of timing verification using static timing analysis. The book covers topics such as cell timing, interconnect, timing calculation, and crosstalk, which can impact the timing of a nanometer design. It describes how the timing information is stored in cell libraries which are used by synthesis tools and static timing analysis tools to compute and verify timing.

This book covers CMOS logic gates, cell library, timing arcs, waveform slew, cell capacitance, timing modeling, interconnect parasitics and coupling, pre-layout and post-layout interconnect modeling, delay calculation, specification of timing constraints for analysis of internal paths as well as IO interfaces. Advanced modeling concepts such as composite current source (CCS) timing and noise models, power modeling including active and leakage power, and crosstalk effects on timing and noise are described.

The static timing analysis topics covered start with verification of simple blocks particularly useful for a beginner to this area. The topics then extend to complex nanometer designs with concepts such as modeling of on-chip variations, clock gating, half-cycle and multicycle paths, false paths, as well as timing of source synchronous IO interfaces such as for DDR memory interfaces. Timing analyses at various process, environment and interconnect corners are explained in detail. Usage of hierarchical design methodology involving timing verification of full chip and hierarchical building blocks is covered in detail. The book provides detailed descriptions for setting up the timing analysis environment and for performing the timing analysis for various cases. It describes in detail how the timing checks are performed and provides several commonly used example scenarios that help illustrate the concepts. Multi-mode multi-corner analysis, power management, as well as statistical timing analyses are also described.

Several chapters on background reference materials are included in the appendices. These appendices provide complete coverage of SDC, SDF and SPEF formats. The book describes how these formats are used to provide information for static timing analysis. The SDF provides cell and interconnect delays for a design under analysis. The SPEF provides parasitic information, which are the resistance and capacitance networks of nets in a

design. Both SDF and SPEF are industry standards and are described in detail. The SDC format is used to provide the timing specifications or constraints for the design under analysis. This includes specification of the environment under which the analysis must take place. The SDC format is a defacto industry standard used for describing timing specifications.

The book is targeted for professionals working in the area of chip design, timing verification of ASICs and also for graduate students specializing in logic and chip design. Professionals who are beginning to use static timing analysis or are already well-versed in static timing analysis can use this book since the topics covered in the book span a wide range. This book aims to provide access to topics that relate to timing analysis, with easy-to-read explanations and figures along with detailed timing reports.

The book can be used as a reference for a graduate course in chip design and as a text for a course in timing verification targeted to working engineers. The book assumes that the reader has a background knowledge of digital logic design. It can be used as a secondary text for a digital logic design course where students learn the fundamentals of static timing analysis and apply it for any logic design covered in the course.

Our book emphasizes practicality and thorough explanation of all basic concepts which we believe is the foundation of learning more complex topics. It provides a blend of theoretical background and hands-on guide to static timing analysis illustrated with actual design examples relevant for nanometer applications. Thus, this book is intended to fill a void in this area for working engineers and graduate students.

The book describes timing for CMOS digital designs, primarily synchronous; however, the principles are applicable to other related design styles as well, such as for FPGAs and for asynchronous designs.

Book Organization

The book is organized such that the basic underlying concepts are described first before delving into more advanced topics. The book starts

with the basic timing concepts, followed by commonly used library modeling, delay calculation approaches, and the handling of noise and crosstalk for a nanometer design. After the detailed background, the key topics of timing verification using static timing analysis are described. The last two chapters focus on advanced topics including verification of special IO interfaces, clock gating, time borrowing, power management and multi-corner and statistical timing analysis.

Chapter 1 provides an explanation of what static timing analysis is and how it is used for timing verification. Power and reliability considerations are also described. Chapter 2 describes the basics of CMOS logic and the timing terminology related to static timing analysis.

Chapter 3 describes timing related information present in the commonly used library cell descriptions. Even though a library cell contains several attributes, this chapter focuses only on those that relate to timing, crosstalk, and power analysis. Interconnect is the dominant effect on timing in nanometer technologies and Chapter 4 provides an overview of various techniques for modeling and representing interconnect parasitics.

Chapter 5 explains how cell delays and paths delays are computed for both pre-layout and post-layout timing verification. It extends the concepts described in the preceding chapters to obtain timing of an entire design.

In nanometer technologies, the effect of crosstalk plays an important role in the signal integrity of the design. Relevant noise and crosstalk analyses, namely glitch analysis and crosstalk analysis, are described in Chapter 6. These techniques are used to make the ASIC behave robustly from a timing perspective.

Chapter 7 is a prerequisite for succeeding chapters. It describes how the environment for timing analysis is configured. Methods for specifying clocks, IO characteristics, false paths and multicycle paths are described in Chapter 7. Chapter 8 describes the timing checks that are performed as part of various timing analyses. These include amongst others - setup, hold and asynchronous recovery and removal checks. These timing checks are intended to exhaustively verify the timing of the design under analysis.

Chapter 9 focuses on the timing verification of special interfaces such as source synchronous and memory interfaces including DDR (Double Data Rate) interfaces. Other advanced and critical topics such as on-chip variation, time borrowing, hierarchical methodology, power management and statistical timing analysis are described in Chapter 10.

The SDC format is described in Appendix A. This format is used to specify the timing constraints of a design. Appendix B describes the SDF format in detail with many examples of how delays are back-annotated. This format is used to capture the delays of a design in an ASCII format that can be used by various tools. Appendix C describes the SPEF format which is used to provide the parasitic resistance and capacitance values of a design.

All timing reports are generated using PrimeTime, a static timing analysis tool from Synopsys, Inc. Highlighted text in reports indicates specific items of interest pertaining to the explanation in the accompanying text.

New definitions are highlighted in **bold**. Certain words are highlighted in *italics* just to keep the understanding that the word is special as it relates to this book and is different from the normal English usage.

Acknowledgments

We would like to express our deep gratitude to eSilicon Corporation for providing us the opportunity to write this book.

We also would like to acknowledge the numerous and valuable insights provided by Kit-Lam Cheong, Ravi Kurlagunda, Johnson Limqueco, Pete Jarvis, Sanjana Nair, Gilbert Nguyen, Chris Papademetrious, Pierrick Pedron, Hai Phuong, Sachin Sapatnekar, Ravi Shankar, Chris Smirga, Bill Tuohy, Yeffi Vanatta, and Hormoz Yaghutiel, in reviewing earlier drafts of the book. Their feedback has been invaluable in improving the quality and usefulness of this book.

Last, but not least, we would like to thank our families for their patience during the development of this book.

Dr. Rakesh Chadha
Dr. J. Bhasker

January 2009

CHAPTER

1

Introduction

This chapter provides an overview of the static timing analysis procedures for nanometer designs. This chapter addresses questions such as, what is static timing analysis, what is the impact of noise and crosstalk, how these analyses are used and during which phase of the overall design process are these analyses applicable.

1.1 Nanometer Designs

In semiconductor devices, metal interconnect traces are typically used to make the connections between various portions of the circuitry to realize the design. As the process technology shrinks, these interconnect traces have been known to affect the performance of a design. For deep submi-

J. Bhasker and R. Chadha, *Static Timing Analysis for Nanometer Designs: A Practical Approach*,
DOI: 10.1007/978-0-387-93820-2_1, © Springer Science + Business Media, LLC 2009

cron or nanometer process technologies[1], the coupling in the interconnect induces noise and crosstalk - either of which can limit the operating speed of a design. While the noise and coupling effects are negligible at older generation technologies, these play an important role in nanometer technologies. Thus, the physical design should consider the effect of crosstalk and noise and the design verification should then include the effects of crosstalk and noise.

1.2 What is Static Timing Analysis?

Static Timing Analysis (also referred as STA) is one of the many techniques available to verify the timing of a digital design. An alternate approach used to verify the timing is the timing simulation which can verify the functionality as well as the timing of the design. The term *timing analysis* is used to refer to either of these two methods - static timing analysis, or the timing simulation. Thus, timing analysis simply refers to the analysis of the design for timing issues.

The STA is *static* since the analysis of the design is carried out statically and does not depend upon the data values being applied at the input pins. This is in contrast to simulation based timing analysis where a stimulus is applied on input signals, resulting behavior is observed and verified, then time is advanced with new input stimulus applied, and the new behavior is observed and verified and so on.

Given a design along with a set of input clock definitions and the definition of the external environment of the design, the purpose of static timing analysis is to validate if the design can operate at the rated speed. That is, the design can operate safely at the specified frequency of the clocks without any timing violations. Figure 1-1 shows the basic functionality of static

1. Deep submicron refers to process technologies with a feature size of 0.25µm or lower. The process technologies with feature size below 0.1µm are referred to as *nanometer technologies*. Examples of such process technologies are 90nm, 65nm, 45nm, and 32nm. The finer process technologies normally allow a greater number of metal layers for interconnect.

timing analysis. The **DUA** is the design under analysis. Some examples of timing checks are setup and hold checks. A setup check ensures that the data can arrive at a flip-flop within the given clock period. A hold check ensures that the data is held for at least a minimum time so that there is no unexpected pass-through of data through a flip-flop: that is, it ensures that a flip-flop captures the intended data correctly. These checks ensure that the proper data is ready and available for capture and latched in for the new state.

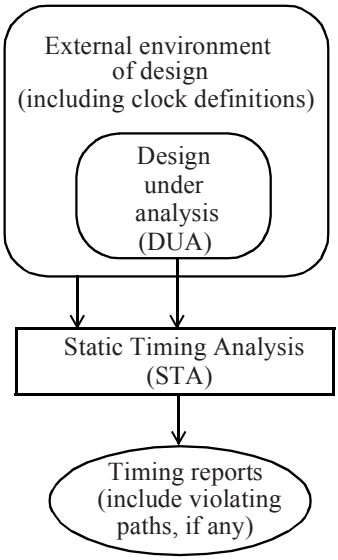

Figure 1-1 *Static timing analysis.*

The more important aspect of static timing analysis is that the entire design is analyzed once and the required timing checks are performed for all possible paths and scenarios of the design. Thus, STA is a complete and exhaustive method for verifying the timing of a design.