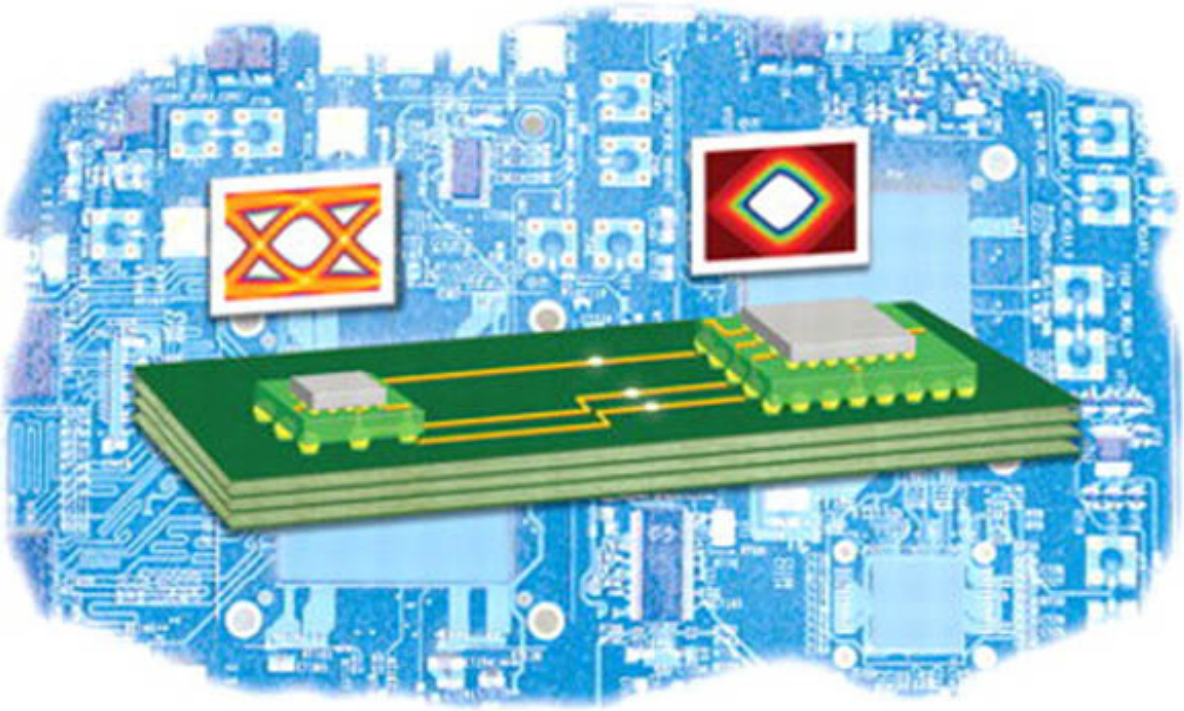
PRENTICE HALL

High-Speed Signaling:
Jitter Modeling, Analysis, and Budgeting



Kyung Suk (Dan) Oh and Xingchao (Chuck) Yuan

Prentice Hall Modern Semiconductor Design Series

Prentice Hall Signal Integrity Library

HIGH-SPEED SIGNALING

Jitter Modeling, Analysis, and Budgeting

KYUNG SUK (DAN) OH
XINGCHAO (CHUCK) YUAN

Prentice Hall
Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Library of Congress Cataloging-in-Publication Data

High speed signaling : jitter modeling, analysis, and budgeting / edited by Kyung Suk (Dan) Oh, Xingchao (Chuck) Yuan.

p. cm.

Includes bibliographical references.

ISBN-13: 978-0-13-282691-4 (hardback : alk. paper)

ISBN-10: 0-13-282691-7 (hardback : alk. paper) 1. Signal processing. 2. Telecommunication—Traffic—Mathematical models. 3. Signal integrity (Electronics) I. Oh, Kyung Suk, 1965- II. Yuan, Xingchao. III. Title.

TK5102.9.H564 2012

621.382’2—dc23

2011031034

ISBN-13: 978-0-13-282691-4
ISBN-10: 0-13-282691-7

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts. First printing October 2011

Publisher
Paul Boger

Acquisitions Editor
Bernard Goodwin

Managing Editor
Kristy Hart

Project Editor
Betsy Harris

Copy Editor
Paula Lowell

Indexer
Erika Millen

Proofreader
Debbie Williams

Publishing Coordinator
Michelle Housley

Cover Designer
Anne Jones

Compositor
Nonie Ratcliff

To my wife MyungSook, and our children, Terry and Christopher.
—Dan Oh

To my wife Jackie, my daughter Caterina, and my son Michael.
—Chuck Yuan

Contents

Preface

The majority of the books on signal integrity focus on techniques to validate and design physical passive channels, such as the package, printed circuit board, and power delivery network. Unfortunately, we cannot mitigate all signal integrity issues with package or board-level design improvements. Relying only on physical design improvements can lead to suboptimal or unrealistic system-level solutions. We must consider signal integrity issues at the early design stage of a modern high-speed I/O interface; engineers from various fields (such as architecture, circuit, system engineering, and signal integrity) must work together to find the best system-level solution for a target application.

This book serves as a bridge for engineers coming from different fields, because it is written from the perspective of I/O (input/output) link design. It starts with the basics of signaling components, which serves as a good starting point for circuit and architectural engineers who want to delve further into signal integrity issues. It also introduces the concept of I/O design, for the benefit of signal integrity and systems engineers.

Traditionally, I/O interface designs had clear boundaries for the roles for circuit, signal integrity, and systems engineers. Circuit designers designed a transceiver to meet the target performance requirements, using either a simplified channel model, or more complex channel models provided by the signal integrity engineers. Signal integrity or systems engineers designed the board and package to minimize signal-integrity issues, using either a simple behavior driver model, or a more accurate full-circuit model provided by the circuit designers. Although this approach is still widely used, it is no longer sufficient for today's high-performance systems. For instance, in high-speed I/O systems, noise and jitter (due to devices and boards) are no longer independent and separable: Engineers must co-optimize them in the circuit and board designs, sometimes even at the architectural level. To model this complex interaction of noise and jitter, modern high-speed interface designs need a new simulation methodology: one that predicts the accurate link-level performance (including the interaction of noise or jitter between the transmitter, receiver, and passive channel [such as packages and boards]). Traditional SPICE-based simulation approaches can no longer predict the performance of such a complex interaction. Some novel simulation methods have recently appeared in technical journals, and at conferences, but no comprehensive book has been written in this area. This book is perhaps the first book to systematically cover this new simulation methodology.

One of the lessons learned from the advances in power integrity engineering is that an ideal, stable power-delivery network design is no longer possible for modern power-hungry multi-core processors. Circuit designers have learned to design with significant power noise. Any

residual jitter or performance degradation due to power noise is budgeted for in the system margin. For example, in a high-speed I/O interface, the jitter induced by power supply noise is one of the dominant device-timing error terms. Consequently, power supply noise can no longer be budgeted for under the transistor voltage margin headroom. Additionally, this supply noise-induced jitter has broad frequency content, causing it to interact with the other parts of the channel, further complicating modeling issues. This book covers the basics of power supply–induced jitter, and describes characterization and simulation techniques.

Passive channel analysis and modeling has been, and still is, the main task for signal integrity engineers. Decades spent in the study of transmission-line modeling and macro modeling have given us fast channel simulations, but research is still in progress. Numerical inaccuracies or instability issues in transmission-line simulation or macro modeling methods are still some of the hottest topics at signal-integrity conferences. Until now, there has been no single numerical algorithm that provides a stable and accurate broadband model for general interconnect structures. Signal integrity engineers must understand the limitations of existing modeling methods, and apply them carefully. This book reviews the crucial limitations of some popular numerical models, and presents practical tips that you can use to avoid them.

This book is written for practicing engineers and managers working on high-speed system designs, as well as for professionals and graduate students doing research in this field. We have attempted to address all the latest issues and technologies with sufficient background and illustrations. Most of the information contained here has been verified, and has been widely used in real applications. Despite the fact that we have devoted significant effort toward making this book readable for entry-level engineers and graduate-level students, some of the advanced topics require some basic background on the part of the reader. As we cover different subjects from different engineering fields, the background requirements for individual chapters vary slightly. The minimum requirement is a basic knowledge of circuit theory. In addition, depending on the chapter, a basic knowledge of electromagnetics and/or statistics is required.

Acknowledgments

As illustrated by the long list of contributing coauthors, this book is the product of more than a decade of high-speed signal-integrity design experience on the part of many engineers at Rambus Inc. All the former, and current, Rambus SI engineers have contributed directly (or indirectly) to this book through their publications at Rambus. In addition to those listed as coauthors, these contributors include Dr. Wendem Beyene, Mr. Newton Cheng, Mr. Ben Chia, Ms. June Feng, Dr. Ching-Chao Huang, Dr. Cathy Huang, Dr. Woopoung Kim, Mr. Qi Lin, Mr. Frank Lambrecht, Dr. H.J. Liaw, Dr. Chris Madden, Dr. Xioning Qi, Mr. Ali Sarfaraz, and Ms. Ling Yang. As you can see from the contents of this book, the techniques discussed here go beyond the work of SI engineers. Specifically, the signal-integrity engineers at Rambus work closely with cross-functional teams that include architecture, circuit design, and system engineering. Many of these Rambus engineers have indirectly contributed to this book, in various ways, by working with us. We want to thank them all for their contributions. In particular, we want to distinguish a few individuals who have made direct contributions. The data coding (to reduce simultaneous switching noise) is largely the result of work done by Mr. Fred Ware, Dr. John Wilson, and Dr. Aliazam Abbasfar.

Special thanks also go to our former and current managers at Rambus. Without their support, this book would not have been possible. Particularly, we want to thank Mr. David Nguyen and Dr. Ely Tsern, whose encouragement made our work enjoyable. We also want to thank Mr. Kevin Donnelly, who provided some of the earliest encouragement when this book was just an idea. We also want to express our deep gratitude to Ms. Sharon Holt and to Mr. John Kent, both for their generous support, and for providing us with editorial resources.

Much of the transmission-line theory, and the recursive convolution method we describe, are based on the pioneering work done by Dr. Dmitri Kuzetzov and Prof. Jose Schutt-Aine. We want to thank them for their outstanding work and friendly discussions. The discussion on causality was inspired by discussions with Dr. Subramanian Lalgudi (Ansys). The editors also want to express our respect for the pioneering work of Prof. Mark Horowitz and his students. The on-chip measurement techniques and statistical simulation methodology described in this book were co-developed by Stanford University and Rambus, under Prof. Horowitz's guidance.

The editors also want to express our sincere gratitude to our reviewers: Dr. Dale Becker (IBM), Prof. Paul Franzon (NC), and Prof. Jose Schutt-Aine (UIUC) for their time and encouragement. These individuals are among the pioneers in the signal-integrity field, and their work laid the foundation for many of the topics covered in this book.

Our appreciation also goes to the Prentice Hall editors and staff, including Bernard Goodwin, Betsy Harris, Paula Lowell, Debbie Williams, and Michelle Housley, for their editorial

support and encouragement. Mr. Greg Morris is also gratefully acknowledged, for his grammatical editing and document formatting. The editors also want to thank Ms. Yi Jiang for providing the cover picture.

Finally, we are deeply indebted for the support that we received from our families: Dan's wife Myung (and sons Christopher and Terry), and Chuck's wife Jackie (and daughter Caterina and son Michael); who supported us through their endless love. This book is a result of more than three years of effort. In particular, we want to thank our kids, for their encouragement, when they kept asking us "Dad, have you finished your book yet?" Without our family's love and support, which allowed us to work countless weekends and nights, this book would not be possible.

DAN OH AND CHUCK YUAN

About the Authors

Kyung Suk (Dan) Oh received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana-Champaign, in 1990, 1992, and 1995, respectively. His doctoral research was in the area of computational electromagnetics applied to transmission line modeling and simulation. He is a Senior Principal Engineer at Rambus Inc. He leads signal integrity analysis for various products including serial, parallel, and memory interfaces. He is also responsible for developing advanced signal and power integrity analysis tools. His current interests include advance signal and power integrity modeling and simulation techniques, optimization of channel designs for various standard or proprietary I/O links, and application of signaling techniques to high-speed digital links.

Dr. Oh has published more than 80 papers and holds 7 issued U.S. patents and 10 pending patent applications in areas of high-speed link design. He received two Best Paper Awards in DesignCon and 2008 Best Paper Award in the IEEE Advanced Packaging journal. Dr. Oh serves on the technical program committee of IEEE EPEPS, and is a former member of the IEC DesignCon Technical Program Committee.

Xingchao (Chuck) Yuan received his B.S. degree in Electronic Engineering from Nanjing Institute of Technology (now Southeast University), Nanjing, China, in 1982. He received both his M.S. and Ph.D. degrees in Electrical Engineering from Syracuse University, Syracuse, New York, in 1983 and 1987, respectively. After receiving his Ph.D., Dr. Yuan was at the Thayer School of Engineering at Dartmouth College; first as a Postdoctoral fellow, and later as a Research Professor from 1987 to 1990.

From 1990 to 1995, Dr. Yuan was employed by Ansoft Corp., where he led the development of Ansoft's flagship product *HFSS*™ (High Frequency Structure Simulator). His work led to three different product releases, which included features such as modeling conductor and dielectric loss, radiation and periodic boundary conditions for modeling antennas, and electromagnetic scattering/interference problems. He pioneered a fast frequency sweep method that combined a finite element method and an asymptotic waveform evaluation method. This led to a dramatic speed improvement in the speed of 3D full-wave modeling. From 1995 to 1998, Dr. Yuan was with Cadence Corp. where he led the research and development of the signal integrity and EMI tools. His work focused on modeling SSO noise and induced electromagnetic interference, which led to some of the earliest research in power plane modeling.

Since 1998, Dr. Yuan has been with Rambus Inc, Sunnyvale, California, as a director of signal integrity engineering. Dr Yuan is responsible for designing, modeling, and implementing Rambus multi-gigahertz signaling technologies using conventional interconnect technologies.

His technical and managerial leadership at Rambus has led to an industry-recognized signal and power integrity team of experts. Rambus' SI/PI papers are closely followed by the rest of the industry, and represent the latest developments in high-performance signal and power integrity modeling and design. Dr. Yuan's team was among the first to apply BER and statistical methodology to memory interface designs, and to explore the relationship between the supply noise spectrum and the jitter spectrum. His team's work led to the successful development of Rambus' XDR memory architecture, which was adopted by PlayStation® 3, DLP projectors, and DTVs. Since 2009, Dr. Yuan has served as an engineering director in charge of a silicon team with dozens of engineers (in both the U.S. and India) who are responsible for designing next-generation Rambus graphics and main memory interfaces. In 2010, the team taped out a multi-modal PHY that explores the limits of single-ended signaling beyond 12.8Gbps, a power efficient differential interface at 20Gbps, and backward compatibility with existing memory interfaces (including GDDR5 and DDR3).

Dr. Yuan has authored more than 100 papers in technical journals and conferences and holds 8 issued U.S. patents. He is a senior member of IEEE, and served on the technical program committee of IEEE EPEPS from 2008 to 2009.

Jihong Ren received her Ph.D in Computer Science from the University of British Columbia in 2006. Since January 2006, she has been with Rambus working on high-speed link analysis, adaptive equalization algorithms, advanced signaling schemes, and circuit analysis. She was the recipient of the Best Paper Award from the IEEE Transactions on Advanced Packaging Journal for 2008.

Hai Lan is a Principal Engineer at Rambus Inc., where he has been focusing on on-chip power integrity and jitter analysis in high-speed and/or low-power I/O interfaces, specializing in advanced modeling, simulation, and on-chip characterization. He received his Ph.D. degree in Electrical Engineering from Stanford University in 2006. His doctoral research focused on substrate coupling noise in mixed-signal ICs and SoC. He received his M.S. in Electrical and Computer Engineering from Oregon State University in 2001, where he developed closed-form expressions for frequency-dependent on-chip interconnect parameters. He received his B.S. in Electronic Engineering from Tsinghua University in 1999. His professional interests include power integrity and signal integrity, mixed-signal integrated circuits design, high-speed interconnects modeling, and advanced silicon effects such as substrate noise coupling.

Ralf Schmitt received his Ph.D. in Electrical Engineering from the Technical University of Berlin, Germany. Since 2002, he has been with Rambus as an engineering manager responsible for power integrity at the chip, package, and system level. His professional interests include on-chip signal integrity, power integrity, timing analysis, clock distribution, and high-speed digital circuit design.

Sam Chang is currently an engineer in the Signal Integrity Division at Rambus. His current work involves the modeling and development of solutions for high-speed digital links. Dr. Chang holds

a Ph.D. in Electrical Engineering (2005) from University of California, Riverside. He has published more than 20 papers in areas of high-speed link design and digital communications. He received the 2008 Best Paper Award in the IEEE Advanced Packaging journal.

Joong-Ho Kim received his Ph.D. degrees in electrical and computer engineering from Georgia Institute of Technology, Atlanta, Georgia, in 2002. During his graduate study, he worked at the Packaging Research Center (PRC), where he researched efficient electrical modeling techniques for analyzing packages/systems. As a result of his research, Dr. Kim developed transmission matrix and macro-modeling methods for analyzing large and complex power delivery networks.

In 2005, Dr. Kim joined Rambus. He is currently a Principal Engineer in the Technology Development Division. He is responsible for product/technology design and analysis in signal/power integrity (SI/PI) area for high-performance memory interface products (such as XDR, DDR2/3, GDDR3/4/5, Mobile XDR, and LPDDR1/2), and next generation mobile, computing, and graphic memory systems. Previously, he worked on SI/PI analysis for CMOS microprocessors and in-house tool developments at Intel Corporation. His current research interests are a systematic approach that considers both signal and power integrity simultaneously, characterization of high-speed interconnects, S-parameter–based simulations using VNA measurements or full-wave solvers, and macro modeling for circuit simulations.

Dr. Kim has published more than 40 papers in journals and conferences, one book (as a chapter author in the IEC conference publications), and has nine issued patents. He received the Best Paper Award at EPEP 2000, 1st Place Poster Award at NSF-PRC 2000, Best Paper Award at Intel DTTC 2004, and two Best Paper Awards at IEC DesignCon 2008.

Ravi Kollipara is a Senior Principal Engineer responsible for the signal integrity of high-speed serial links and parallel bus channels. His duties include design and characterization of channel passive components (such as packages and PCBs), measurement-based modeling of connectors, and development of system voltage and timing budgets. He has been with Rambus since 1998. Prior to joining Rambus, Dr. Kollipara worked at LSI Logic and taught as a visiting assistant professor. He received his B.E, M.Tech, and Ph.D. degrees from Andhra University, IIT Delhi, and Oregon State University, respectively.

Jared Zerbe graduated from Stanford University, Stanford, California, in 1987. From 1987 to 1992, he was with VLSI Technology and MIPS Computer Systems. In 1992, he joined Rambus, where he has since specialized in the design of high-speed I/O, PLL/DLL clock-recovery, and equalization and data-synchronization circuits. He has authored multiple papers and has been awarded patents in the areas of high-speed clocking and data transmission. Mr. Zerbe has taught courses at Berkeley and Stanford in link design. He is currently a Technical Director, and focuses on the development of future signaling technologies.

Hao Shi received his B.S. degree in microelectronics from Beijing University in 1984, M.S.-Physics degree in Physics from University of Missouri-Kansas City in 1993, M.S.-E.E. and

Ph.D.-E.E. degrees from Missouri University of Science and Technology (also known as University of Missouri-Rolla) in 1995 and 1997, respectively. Between 1984 and 1993, he was involved in material science research of high-Tc superconductors, specializing in measurements. From 1998 to 2002, he was with Hewlett-Packard and Agilent Technologies, holding engineer positions in the fields of EDA, analog, and microwave designs. From 2002 to 2008, he was with Rambus Inc. as a signal integrity engineer. He is currently with Apple, Inc. as a principal SI Engineer. His professional interests include modeling and simulation of interconnects in PCB and packaging, power delivery system decoupling, noise containment of switched-mode power supplies, connector modeling, and electromagnetic compatibility.

On various subjects concerning signal integrity, Dr. Shi has published 8 peer-reviewed journal papers (as the first author in 4), and 17 conference papers (as the first author in 8). He was one of the lead authors of the issued U.S. Patent No. 7476813 as well as the patent application No. 20100096725, both of which are related to packaging designs. He was the winner of the President's Memorial Award from the IEEE-EMC society in 1995, and he is currently a senior member of the IEEE.

Vladimir Stojanovic is the Emmanuel E. Landsman Associate Professor of Electrical Engineering and Computer Science at MIT. His research interests include design, modeling and optimization of integrated systems, from CMOS-based VLSI blocks and interfaces to system design with alternative devices (such as NEM relays and silicon-photonics). He is also interested in the design and implementation of energy-efficient electrical and optical networks, and in digital communication techniques for high-speed interfaces and high-speed mixed-signal IC design. He is a recipient of the 2009 NSF CAREER award.

Dr. Stojanovic received his Ph.D. in Electrical Engineering from Stanford University in 2005. He received his M.S. degree in Electrical Engineering from Stanford University in 2000, and the Dipl. Ing. degree from the University of Belgrade, Serbia, in 1998. He was with Rambus from 2001 through 2004. He was also a visiting scholar with the Advanced Computer Systems Engineering Laboratory, Department of Electrical and Computer Engineering, University of California, Davis, during 1997–1998.

Elad Alon received B.S., M.S., and Ph.D. degrees in Electrical Engineering from Stanford University in 2001, 2002, and 2006, respectively. In Jan. 2007, he joined the University of California at Berkeley as an Assistant Professor of Electrical Engineering and Computer Sciences, where he is now a co-director of the Berkeley Wireless Research Center (BWRC). He has held positions at Sun Labs, Intel, AMD, Rambus, Hewlett-Packard, and IBM Research, where he worked on digital, analog, and mixed-signal integrated circuits for computing, test and measurement, and high-speed communications. Dr. Alon received the IBM Faculty Award in 2008, the 2009 Hellman Family Faculty Fund Award, and the 2010 UC Berkeley Electrical Engineering Outstanding Teaching Award. His research focuses on energy-efficient integrated systems, including the circuit, device, communications, and optimization techniques used to design them.

This page intentionally left blank

CHAPTER 1

Introduction

Dan Oh and Chuck Yuan

Computing devices, such as computer servers, workstations, personal computers, game consoles, and smart phones, have become increasingly more powerful with each new generation of semiconductor process. Thanks to Moore's Law, which states that the number of transistors on a chip doubles every two years [1], there is not only more functionality available for a given device, but also an increase in performance. To keep up with the increase in performance, the data communication speed between the components of the computing device has also been increasing, rising from a few hundred Mb/s in the early 1990s to several Gb/s in 2008. It is projected that data communication rates will soon increase to tens of Gb/s. For instance, the next generation PCIe specification is considering 8Gb/s as a target data rate and is expected to be in production by 2012.

As data communication reaches multi-gigabit/sec rates, the task of ensuring good signal integrity, both on-chip and off-chip, becomes increasingly important. Understanding the high-frequency physical effects introduced by the wire or interconnect is as important as the silicon design itself. Moreover, device jitter (generated by on-chip circuitry) now becomes a signal-integrity (SI) problem, because system-level behavior (such as jitter amplification and cancelation) must be modeled. The time when signal integrity was considered, only after the silicon was built, has passed. The I/O interface designer, or system designer, must perform a thorough signal-integrity analysis to avoid producing non-reliable or overly constrained systems, or incurring costly recalls of products from the marketplace.

Signal-integrity design considerations must be considered upfront to ensure the robust operation of modern high-speed digital systems. New design methodologies must be introduced and employed to account for the physical effects that could be ignored at lower data rates. To minimize timing errors with the new target data rates and channel designs, clocking or timing circuitry designs must be optimized. Before building any hardware or system, worst-case design

parameters and interconnect electrical behavior must be evaluated and analyzed. A detailed and accurate understanding of the electrical behavior of interconnect, advanced signaling, and circuit techniques (such as equalization) can be used to overcome the non-ideal effects introduced by interconnects.

Accurate prediction of system behavior at multi-gigabit data rates is a challenging task that requires a signal-integrity engineer who possesses knowledge of, and experience in, several diverse engineering disciplines. Specifically, such an engineer must have knowledge of digital system engineering, high-speed I/O circuit design, electronic package and printed circuit board design, communications theory, microwave engineering, and computational electromagnetics. Because of these multi-disciplinary requirements, signal-integrity engineers come from many different technical backgrounds, such as circuit and printed circuit board design, RF/microwave engineering, and electromagnetic modeling. The signal-integrity engineer then gains the necessary knowledge and experience on the job. Few universities offer courses and training programs that specifically teach signal integrity, which contributes to the growing shortage of signal integrity engineers.

Because signal integrity is a relatively new, fast-evolving, and multi-disciplinary field, few good reference books exist on the subject. H.B. Bakoglu's book, published in 1990, is a good introduction to signal integrity [2]. Bakoglu's primary audience is the silicon circuit designer who wants to understand the impact of interconnects on high-speed data transmission. H. W. Johnson's book, published in 1993 [3], is a practical handbook for signal integrity engineers. W. Dally's work, published by Cambridge University Press in 1998 [4], offers comprehensive information on high-speed digital system design. It offers excellent information about designing high-speed signaling systems by considering the impact of circuit design, packaging and interconnect design, and power distribution network design. Recently, more books on signal integrity design and engineering have become available [5–19]. Those books cover a wide range of topics, including printed circuit board design, system timing analysis, substrate noise coupling, and power supply noise modeling.

Although the aforementioned books have been very useful to signal integrity engineers, most of them focus on one specific topic. Few take a systematic approach and discuss how to design a high-speed system from the architecture design phase to production, or how to ensure robust system operation under worst-case operating conditions. Finally, few offer information about how to achieve maximum system yield for high-volume manufacturing. Some of the material is now outdated, because the data rates have increased from a few megabits to several gigabits. As a result, signal-integrity engineers, who must confront the new challenges of multi-gigabit designs, lack adequate reference material. They must study topics that are common in communication theory, circuit theory, microwave engineering, and computational electromagnetic theory to understand and design a multi-gigahertz system.

This book offers a comprehensive discussion of high-speed signal integrity engineering. It is intended as an intermediate to advanced text to aid signal-integrity engineers in acquiring the necessary skills and knowledge needed to design and model multi-gigabit digital systems. It

assumes the reader has some basic understanding of various electrical engineering subjects, such as VLSI design, transmission-line theory, and microwave engineering. This book draws on 10 years of high-speed signal integrity design experience, from more than two dozen engineers at Rambus Inc. Rambus®-designed I/O interfaces have had a wide range of data rates, ranging from 800Mb/s in the early 1990s, to 16Gb/s in 2009. Most Rambus I/O interfaces have been proprietary, and SI engineers have worked closely with other circuit and architecture engineers to ensure reliable channel performance. SI engineers were involved in defining signaling definitions and circuit requirements, characterizing and simulating a prototype virtual channel, and accounting for mass production environments. This book shares more than a decade of collective experience in analyzing various I/O interfaces, such as on-board parallel busses, backplanes, consumer memory, and PC main memory.

What is unique about this book?

- This book takes a systematic approach and considers signal integrity from the architecture phase to high volume production.
- This book covers a broad range of topics, including the design, implementation, and verification of high-speed I/O interfaces.
- Passive-channel modeling, power-supply noise and jitter modeling, as well as system margin prediction, are considered in extraordinary depth.
- Both signal integrity (SI) and power integrity (PI) are considered in a holistic approach, designed to capture actual system behavior. The impact of power noise-to-signal quality (including both on-chip and off-chip noise) is also considered.
- Methodologies for balancing system voltage and timing budget are explained in detail to help ensure system robustness in high-volume manufacturing.
- Practical, yet stable, formulae to convert various network parameters are described for the first time, as network and transmission line theories are an important part of channel analysis. Broadband modeling of interconnects is quite challenging. Some fundamental issues with existing models and tools are described, along with potential improvements and tips to avoid inaccurate models.
- This book takes a systematic approach, and considers signal integrity from the architecture phase to high volume production.
- This book presents the most recent advances in SI and PI engineering. Specifically, equalization techniques to improve channel performance are explained at a high level. High-volume manufacturing modeling and link jitter/statistical simulation methodologies are covered for the first time. The relationship between jitter and clocking topology is explored in detail. On-chip measurement techniques for in-situ link performance testing are also presented.

1.1 Signal Integrity Analysis Trends

Signal-integrity engineering is a relatively new engineering discipline; its development is driven by the need to design high-speed digital systems. In the early 1990s, when digital systems were relatively slow in terms of operating data rates, signal integrity was often an afterthought. Engineers did not have to worry much about the parasitic effects of passive interconnect, which includes package and printed circuit board traces, via transitions, and connectors. The physical designs of the package and PCB were often simply "connecting dots" in layout tools. However, as the data rates of the high-speed systems increased, people encountered numerous system failures due to parasitic effects, such as crosstalk, reflections, and power-supply noise. As a result, signal integrity engineering has grown from relative obscurity into one of the most important engineering disciplines. This section reviews the history of signal-integrity engineering, discusses its evolution over the past decade, and explores its future directions.

1.1.1 Pre-1990: Era of "Black Magic"

During the early days of the computer, transistor device speed limited the I/O speed. As a result, the parasitic effect on the digital system was negligible. There was no need to be concerned about signal integrity, unless one was designing super computers. During this period, the noise problems related to crosstalk and supply noise were addressed on a case-by-case basis. The necessity of trying to debug system failures drew engineers, with various technical backgrounds and experience, to SI engineering. Typical engineering backgrounds included analog design, I/O circuit design, printed circuit board (PCB) and package design, microwave engineering, and electromagnetic modeling. In fact, SI tasks were considered a "side job," rather than as a primary job function.

During these early days, SI engineering was in its infancy. Several problems typified the period: First, the physics of noise in digital systems was poorly understood. Though parasitic effects at high frequency were well studied in related microwave engineering, little knowledge was transferred to digital design. Second, digital designers ignored the impact of parasitic effects during the design phase. The problem was addressed only after the appearance of system instability, or a failure. Little effort was spent trying to understand the failure mechanism. As a result, signal integrity was jokingly referred to as "black magic," rather than engineering. Third, a very limited number of tools and methodologies were available with which to model the parasitic effects in digital systems accurately. Finally, the roles and responsibilities of SI engineers were not well defined. As stated before, most engineers had diverse technical backgrounds, and most had a primary job other than SI engineering.

Fortunately, researchers working for high-end system manufacturers (such as IBM®, DEC®, HP®, and Bell Labs®) and engineering schools devoted a vast amount of time to modeling and analyzing interconnect systems. Although their work was published in technical journals and conferences (beginning in the early 1970s), there were no textbooks on these topics, as their applications were limited to very high-end computing systems, such as supercomputers and mainframes.

1.1.2 1990–2000: Era of "Passive Channel"

By the early 1990s, the data rates within a computer system had reached several hundred megabits. For example, a high-end PC system had a memory system running at 500–800Mb/s in the early 1990s, while Intel's microprocessor was operating in the gigahertz range. Noise considerations for such systems became much more important. An early signal integrity–related conference called Electrical Performance of Electronic Packaging was established in 1992, and a few other electrical engineering conferences included signal integrity as part of their conference sessions. During this period, SI engineering was quickly developing and rapidly changing in both technical breadth and depth. More practical issues and solutions soon complemented the early research work done by the high-end system manufacturers and university researchers. Figure 1.1 illustrates an SI engineer's various tasks in a typical design process. Many pieces of the "puzzle" had to fit together in order to design a robust high-speed digital system. In contrast to SI engineering in the pre-1990 period, SI engineering was now no longer an afterthought, but an integrated part of high-speed digital system designs. Tools and methodologies that were once only available to a few high-end system manufacturers became readily available through various EDA vendors.

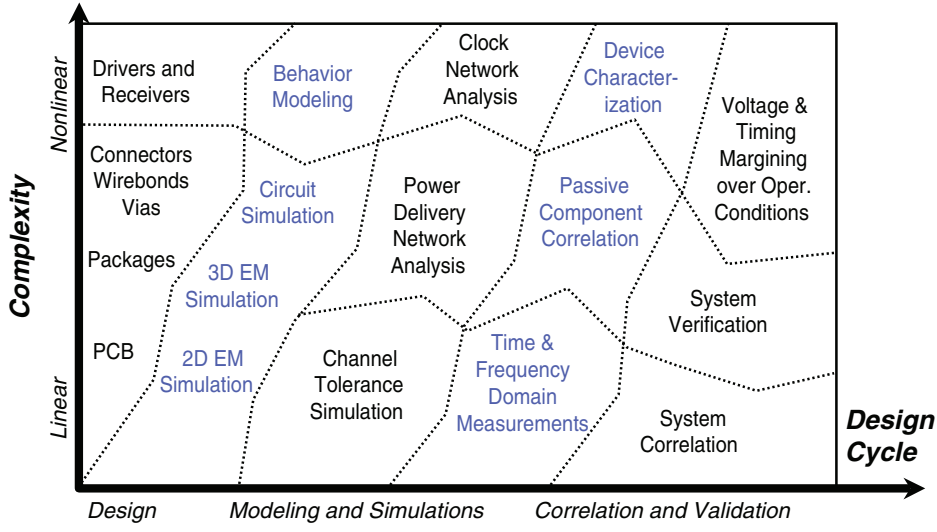


Figure 1.1 SI Engineering Tasks for High-Speed Digital System Design

During this period, much of the SI analysis focused on modeling transmission lines. With HSPICE stable and accurate transmission-line model implementation, engineers were finally able to evaluate the impact of crosstalk, loss, and reflections. Frequency dependent loss, due to dielectric and conductor skin loss, was conveniently evaluated in transient analysis. Electromagnetic (EM) 2D and 3D solvers became available with which to extract either RLGC

(Resistance, inductance, conductance, and capacitance) matrices or scattering parameters. SI engineers created SPICE circuit models based on physical designs using EM modeling. Correlation was performed to validate the passive model in the time domain (using time domain reflectometry [TDR]/scope), or in the frequency domain (using vector network analyzer [VNA]). Finally, the system time and voltage margins under worst-case operating conditions were verified.

Much of the SI work of this period focused on passive-channel modeling and its correlation with hardware measurements. This period can be characterized as "modeling from transmitter die pad to receiver die pad." Everything in the passive channel was modeled. However, what was implemented in silicon was treated as a black box. Behavior models (such as IBIS) were often adopted for transmitter and receivers to minimize SPICE transient simulation time. The interaction between the passive channel and active (Tx/Rx) circuits was ignored, or poorly modeled. Even when a "violation" of the passive channel specification was observed, the overall system failure could not necessarily be concluded. Furthermore, many companies did not understand the importance of signal-integrity engineering; some continued to treat SI as a back-end process, and ignored it until problems appeared later in the design cycle. In addition, there was still some debate regarding the roles and responsibilities of SI engineers, and the future of SI engineering [20]. In summary, SI engineering played an important, but limited, role in high-speed digital system design during the 1990s.

1.1.3 2000–Present: Era of "Entire Link"

At present, data rates for computing systems have reached several gigahertz levels. For example, Sony's PlayStation® 3 uses a differential XDR™ memory system that supports data rates ranging from 3.2Gb/s to 6.4Gb/s. Intel's microprocessor currently operates at more than 3GHz. Data rates for parallel on-board interfaces and high-end graphics memory interfaces have reached several Gb/s levels. Data communications for modern routers and switches have driven the need for very-high-speed serial links. For example, the Optical Internetworking Forum (OIF) standards call for 6 to 12Gb/s for backplane systems. For multi-gigahertz applications, the channel often defines the speed limit. As a result, much of the design attention focuses on mitigating the non-ideal physical effects caused by channel, and in particular, by inter-symbol interference (ISI).

During this period, SI has become one of the important architecture drivers. SI engineers now interact with system architects, circuit designers, and system engineers throughout the design cycle: from conception to mass production to cost reduction. SI engineering has gone beyond conventional passive interconnect modeling, and now attempts to model the entire link. This includes the transmitter, receiver, clock, and channel. SI engineering excels in signaling architecture analysis and performance trade-offs. SI modeling analysis of the entire link influences design issues, such as equalization architecture, clock architecture, timing calibration architecture, coding, and/or error correction architectures. A significant portion of this book is dedicated to this new era of signal integrity analysis, which is henceforth referred to as signaling analysis.

1.1.4 Future: Era of "Power Optimized Link"

This section describes new areas where SI analysis will be required in the near future, based on the authors' current experience and technology trends. Briefly look back at what has happened in the past from a device point of view. The scaling of CMOS feature size and voltage has helped maintain constant power per unit area [21] allowing more transistors to be packed in the same area. This directly increased the performance of the chip, which in turn, required a high-speed I/O interface. However, the voltage scaling has significantly slowed, as the threshold voltage (Vth) could not scale accordingly due to leakage power. As a result, power consumption per unit area is no longer constant and continues to increase. Figure 1.2 shows this scaling trend and the power consumption for microprocessors.

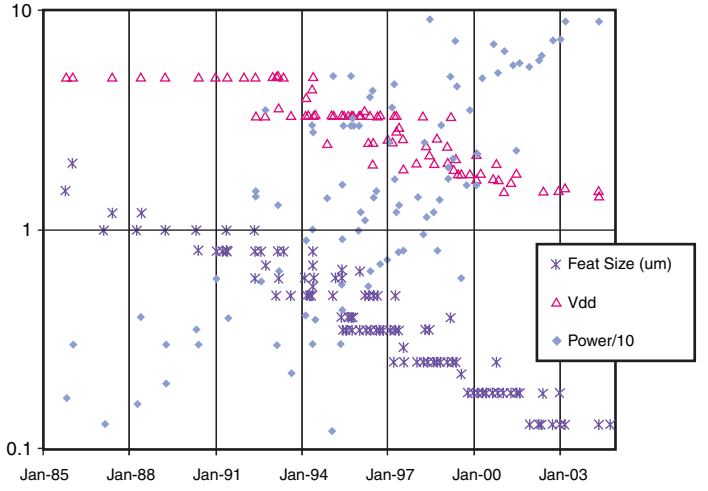


Figure 1.2 Microprocessor Vdd, Power/10, and Feature Size vs. Year [21] (© 2005 IEEE).

Given the slowdown in voltage scaling, the current generation of I/O interface designs needs to consider the optimized data rate for a target process. Power per bit became a common metric for evaluating the link performance, rather than pure performance. Power consumption for a given process is normalized, in terms of FO4 (fan out of 4) delay time, in order to predict the optimum data rate, independent of process technology in [22]. Basic trade-off analysis, in terms of data rate and power consumption with different signal conditioning schemes, would be useful in future signaling analysis.

Traditional I/O interface designs focus on one target data rate, which represents the highest performance in terms of data rate, power consumption, and system cost. This is no longer sufficient for power-critical applications, such as fast-growing mobile applications. Application

processors for such systems now use multiple I/O data rates to optimize the power consumption for various applications. Furthermore, extensive power-managing schemes, such as shutting down the I/O interface (or portions of it), are now commonly employed [23]. Therefore, signaling design must consider a wide range of data rates, and signal integrity analysis must consider non-ideal conditions (due to transitions to different power modes or data rates), to achieve uninterrupted or minimum degradation of I/O performance.

3D integration is another new area to apply signaling analysis. 3D integration shortens the I/O channel, but is subject to more on-chip noise, because the small form-factor makes providing a stable supply quite challenging. In this application, I/O performance is limited more by clock distribution, because the clock tree can span a greater distance than the I/O interconnect itself. Modeling and minimizing the jitter of the clock distribution is crucial in this application. So far, the impact of core noise on I/O has largely been neglected, as I/O typically has a separate power rail (but this may no longer be true for 3D integration). For high-speed I/O with 3D integration, an on-chip power regulator is desirable, and the design trade-off between the on-chip regulator and I/O interface is critical.

On-chip regulators will be more common with off-chip interfaces, because low-swing signaling is desirable for low-power applications [23]. Such interfaces have a minimum output supply noise, even for single-ended signaling designs, and the major supply noise–induced jitter would be from the pre-driver or clock tree. Power supply noise–induced jitter for these circuits will play a more important role, and the signaling analysis must include the impact of these effects. In summary, future signal integrity analysis will be more challenging, and will require a broader knowledge of interface architecture.

1.2 Challenges of High-Speed Signal Integrity Design

This section provides detailed descriptions of a number of the challenges facing signal-integrity (SI) engineers during high-speed SI design.

One challenge is that system-design methodology must change so that SI concerns are accounted for during the architecture phase, rather than later in the process. This issue is more pronounced for designs moving into high data rates. In the past, engineers have relied on their own experience until something goes wrong. This can be very costly in terms of product delay and returns.

SI engineers also need to identify the critical timing and voltage parameters and relationships of the design. Having a good understanding of the signaling methods and clocking architecture is critically important. Not all SI engineers will have the opportunity to work on a new signaling method; most engineers typically work with a standard defined by industry consortium. Even in this case, the SI engineer needs to understand how signaling functions and its key requirements. Identifying worst-case scenarios is also crucial.

SI engineers must be able to build accurate models for passive interconnect as well, including the package, PCB, and connectors. These models must capture frequency dependent loss,

crosstalk, and reflections. They must also capture 3D, as well as full-wave effects. These models can be used in either time-domain or frequency-domain simulations.

SI engineers must build confidence in the accuracy of the passive model by performing detailed correlation with the hardware, in both the time and frequency domains, using VNA and TDR. The impact of manufacturing tolerances must be considered for high volume productions.

SI engineers also need to build an accurate model of the power distribution network, to account for the effects of supply noise on system performance. The power distribution network must not only be appropriate for on-chip power delivery analysis (such as IR, EM, and AC supply noise); it must also be able to capture system behavior, such as coupling between signal and supply rails. The supply voltage tolerance at transistors and package pins must be defined. Bypassing requirements on-chip, on-package, and on PCB must be defined for suppressing high, medium, or low frequency supply noise.

SI engineers need to account for the effects of non-ideal circuit behavior as well, such as transmitter jitter and receiver offset and/or sensitivity. Deterministic noise sources (such as DCD or ISI), and random noise sources (due to thermal or shot noise) must be modeled. The ability of SI engineers to work in a multi-disciplinary environment is very important when accessing the risks or benefits of various design options, and when helping to define an optimal signaling architecture, in terms of both speed and power. Moreover, SI engineers must understand the relationship between supply noise and jitter for a given clocking architecture. Certain clocking architectures may be more susceptible to noise than others. Finding the noise-to-jitter transfer function is essential.

Finally, SI engineers must be able to work in the lab, using various instruments, ranging from VNA, TDR, DCA (digital sampling scope), spectrum analyzers, and BERT (bit error rate tester). One must be able to capture waveforms in the lab, correlate them with simulation, and explain the observed system behavior. Using the correlated model, one must be able to find the root cause of the failure or instability, as well as to recommend design changes for future improvements.

1.3 Organization of This Book

Chapter 2, “High-Speed Signaling Basics,” is an overview of signaling basics. It describes the fundamental blocks of I/O signaling channels and introduces basic I/O interface design. Without delving into details, it depicts an overall description of I/O interface design, including various clocking and topology options that are often not considered in a traditional signal integrity subject. It also covers major noise components in high-speed I/O links. The basic physics of these noise components are discussed, along with modeling issues.

The remaining chapters are organized into four parts. Part I consists of three chapters on passive-channel modeling. The first chapter, Chapter 3, “Channel Modeling and Design Methodology,” presents an overall channel modeling and design methodology. It focuses on a general flow of passive channel modeling. Channel modeling often requires conversion of various network models, and Chapter 4, “Network Parameters,” provides conversion formulae for different

network parameters. It also presents a few issues in S-parameter modeling, which has recently gained more popularity. This chapter also describes the passivity condition of a network parameter. Finally, Chapter 5, "Transmission Lines," discusses the transmission line model, as well as a popular recursive convolution method and its limitations. Generating transmission line models from measurement data is described in detail. The characteristics of three different interconnect types, a PCB trace, package trace, and on-chip interconnect, are discussed.

Part II considers the simulation and analysis aspects of the channel. Five chapters are devoted to this topic. The first chapter, Chapter 6, "Channel Voltage and Timing Budget," discusses challenges in link performance analysis and reviews conventional voltage and timing budget analysis. The remaining four chapters address these challenges, and cover new simulation methodologies. Chapter 7, "Manufacturing Variation Modeling," introduces Design of Experiment (DoE) in channel analysis. DoE guarantees reliable channel performance for mass-production systems with manufacturing variations. Chapter 8, "Link BER Modeling and Simulation," presents a statistical link simulation framework, which can model both device-timing jitter and voltage noise, in addition to the traditional channel effects. Although the statistical link simulator is a powerful tool, used to predict the link's performance, it has a few serious limitations (such as difficulties in modeling non-linear drivers and accounting for data coding). Chapter 9, "Fast Time-Domain Channel Simulation Techniques," explores a fast-time domain simulator, which can be used in conjunction with the statistical framework to mitigate the issues from a pure statistical approach. A significant portion of jitter or noise can be mitigated by using a proper clocking architecture. Chapter 10, "Clock Models in Link BER Analysis," reviews some of the common clocking architectures and their simulation models for statistical link simulators.

Part III explores the impact of power noise to link performance. Chapter 11, "Overview of Power Integrity Engineering," as its name implies, provides an overview of power integrity engineering. Simultaneous Switching Noise (SSN) analysis is a hot issue for modern high-speed memory interface designs. Chapter 12, "SSN Modeling and Simulation," discusses an efficient and accurate simulation methodology for SSN analysis, using a DDR2 memory system to demonstrate the effectiveness of the presented simulation methodology. Noise mechanisms of SSN for common single-ended signaling technologies are also explained. The reduction of SSN is quite challenging, due to the physical limitation of package designs, and Chapter 13, "SSN Reduction Codes and Signaling," presents bus-coding techniques to mitigate SSN. By using differential signaling or data coding, SSN (due to output power supply noise) is no longer the dominant factor for timing jitter. Power supply noise, on the pre-driver and clock path, induces a significant amount of jitter. Chapter 14, "Supply Noise and Jitter Characterization," discusses the basics of power supply noise–induced jitter (PSIJ). Chapter 14 also covers useful on-chip measurement circuits for measuring power noise and power distribution network (PDN) impedance. The proposed measurement technique is further extended to substrate noise measurement in Chapter 15, "Substrate Noise Induced Jitter."

Part IV is devoted to advanced SI/PI topics. Chapter 16, "On-Chip Link Measurement Techniques," describes on-chip measurement techniques for signal performance and noise measurement. Such features are becoming more important, due to the popularity of 3D packages, such as PoP, SiP, and 3D integration. Modern high-speed links utilize signal-conditioning techniques,

used to overcome physical channel limitations, and Chapter 17, "Signal Conditioning," presents a general overview of these equalization techniques. Chapter 18, "Applications," provides three signaling examples to demonstrate the list of common features used in different applications. The first example is an XDR memory system for the high-end PC, game, and graphics applications. Several key architecture-level features, such as FlexPhase for timing adjustment and Dynamic Point-to-Point (DPP) for mitigating multi-drop issues, are also reviewed. The second example is Mobile XDR™ for low-power applications. Additional features, used in Mobile XDR to reduce the interface power, are reviewed in detail. The third example applies these advanced signaling features to the current generation of DDR main memory systems, in order to provide a future roadmap for increasing the data rate. A few highlights of future high-speed interfaces are also presented.

References

1. G. E. Moore (1965). "Cramming more components onto integrated circuits," *Electronics Magazine*.
2. H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.
3. H. W. Johnson and M. Graham, *High Speed Digital Design*, Prentice Hall, 1993.
4. W. Dally and J. Poulton, *Digital System Engineering*, Cambridge University Press, 1998.
5. Balsha R. Stanisic, Rob A. Rutenbar, and L. Richard Carley, *Synthesis of Power Distribution to Manage Signal Integrity in Mixed-Signal ICs*, Springer, 1996.
6. B. Young, *Digital Signal Integrity: Modeling and Simulation with Interconnects and Packages*, Prentice Hall, 2000.
7. S. H. Hall, G. W. Hall, and J. A. McCall, *High Speed Digital System Design: A Handbook of Interconnect Theory and Design Practices*, IEEE Press, 2000.
8. R. Singh, *Signal Integrity Effects in Custom IC and ASIC Designs*, Wiley-IEEE Press, 2001.
9. S. Donnay and G. Gielen, *Substrate Noise Coupling in Mixed-Signal ASICs*, Springer, 2003.
10. E. Bogatin, *Signal Integrity - Simplified*, Prentice Hall, 2003.
11. D. Brooks, *Signal Integrity Issues and Printed Circuit Board Design*, Prentice Hall, 2003.
12. T. Granberg, *Handbook of Digital Techniques for High Speed Design*, Prentice Hall, 2004.
13. D. Miller, *Designing High-Speed Interconnect Circuits: An Introduction for Signal Integrity Engineers*, Intel Press, 2004.

14. S. C. Thierauf, *High-Speed Circuit Board Signal Integrity*, Artech House, 2004.
15. Roy G. Leventhal, L. Green, and D. J. Carpenter, *Semiconductor Modeling: For Simulating Signal, Power, and Electromagnetic Integrity*, Springer, 2006.
16. G. Edlund, *Timing Analysis and Simulation for Signal Integrity Engineers*, Prentice Hall, 2007.
17. M. P. Li, *Jitter, Noise, and Signal Integrity at High-Speed*, Prentice Hall, 2007.
18. M. Swaminathan and A. E. Engin, *Power Integrity Modeling and Design for Semiconductors and Systems*, Prentice Hall, 2007.
19. S. H. Hall and H. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*, IEEE Press, 2008.
20. A. Fraser and S. Argyrakis, "Does signal integrity engineering have a future?," presented at the IEC DesignCon, Santa Clara, CA, 2003.
21. M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, power, and the future of CMOS," in *International Electron Devices Meeting Technical Digest*, Dec. 2005, pp.7–13.
22. H. Hatamkhani and C.K. Yang, "Power analysis for high-speed I/O transmitters," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 2004, pp. 142–145.
23. B. Leibowitz, R. Palmer, J. Poulton, Y. Frans, S. Li, J. Wilson, M. Bucher, A.M. Fuller, J. Eyles, M. Aleksic, T. Greer, and N. Nguyen, "A 4.3GB/s mobile memory interface with power-efficient bandwidth scaling," *IEEE Journal of Solid-State Circuits*, vol 45, no. 4, pp. 889–898, 2010.

CHAPTER 2

High-Speed Signaling Basics

Chuck Yuan and Dan Oh

Signal-integrity analysis has evolved greatly over the last two decades, reaching beyond traditional passive-channel analysis. Previously, driver and receiver performance dominated link performance. However, as data rates continue to increase, link performance is limited more by noise and jitter than intrinsic transistor performance. As the interface speed increases, the noise and jitter (due to the device, channel, and power distribution network) start to interact with each other. This interaction makes analyzing link performance a challenging task, and a common topic for both circuit designers and SI engineers. Often, SI engineers focus on high-frequency modeling and simulation of passive channels: They lack a system-level or circuit-level design perspective. This chapter presents a basic introduction to system-level issues and various noise sources in high-speed link designs.

2.1 I/O Signaling Basics and Components

The particular signaling method used in a digital system has the most direct impact on signal integrity. It fundamentally determines how fast, efficient, and robust the digital system will be. In the past, relatively little attention has been paid to an optimized signaling method for high-speed data transmission, because overall system performance was not severely limited by I/O performance. With system data rates approaching multi-gigahertz, one must carefully choose the signaling method to achieve the desired system performance and robustness. This section introduces the basics of signaling from the perspective of high-speed I/O design and explains how each aspect of a signaling method affects signal integrity.

Signaling is a method used to translate digital symbols (1s and 0s) into physical quantities (voltage or current). A signaling system can be characterized by six basic elements: topology,

transmitter, receiver, interconnect, termination, and clock. Many different signaling methods are available, based on different combinations of these six basic elements.

Here is a brief description of how a typical signaling system works:

1. The transmitter translates logic 1s or 0s into either voltage or current levels, or analog signals.
2. The analog signals are delivered to the receiver by interconnects between the transmitter and receiver.
3. The receiver compares the analog signal to a reference (typically voltage, Vref) to translate the signal back to logic 1s or 0s.
4. After the analog signals arrive at the receiver, terminators remove them from the system.
5. The clock tells the transmitter when to send new signals, and the receivers when to sample them.

Ideally, the signals arrive at the receiver with no distortion. However, this is not the case in the real world, especially when data rates reach many hundreds of megahertz. As the data rate increases, so does signal distortion. In addition, different signaling methods result in dramatically different amounts of signal distortion. Consequently, choosing the right signaling method is essential for high-speed I/O systems. Things to consider when designing, or determining, a signaling method include the data rate, silicon area, power, and system cost. A good signaling method achieves the right balance of these factors for a target application.

2.1.1 Signaling Topologies

Several common signaling topologies exist, as shown in Figure 2.1. Point-to-point topology is the simplest and best, in terms of signal integrity. With point-to-point, one transmitter communicates with one receiver. Multidrop signaling topology (popular for memory systems) is more challenging in terms of signal integrity. With multidrop, several transmitters and receivers share the same communication channel or interconnect. These two topologies are most commonly used. However, many other interesting topologies exist, such as near-end clustering, far-end clustering, and Y-junction.

2.1.2 Transmitters

The transmitter translates the logic 1s or 0s into analog signals, or waveforms, in terms of voltage or current. These, in turn, propagate down the channel to the receivers. The two major classes of transmitters are single-ended and differential. Transmitters can be further classified as either voltage-mode or current-mode transmitters. The voltage-mode transmitter directly forces the output voltage, whereas the current-mode transmitter injects a constant current to drive the output voltage. Transmitter output impedance plays an important role in determining whether the driver is voltage-mode or current-mode (as described in the following subsection).

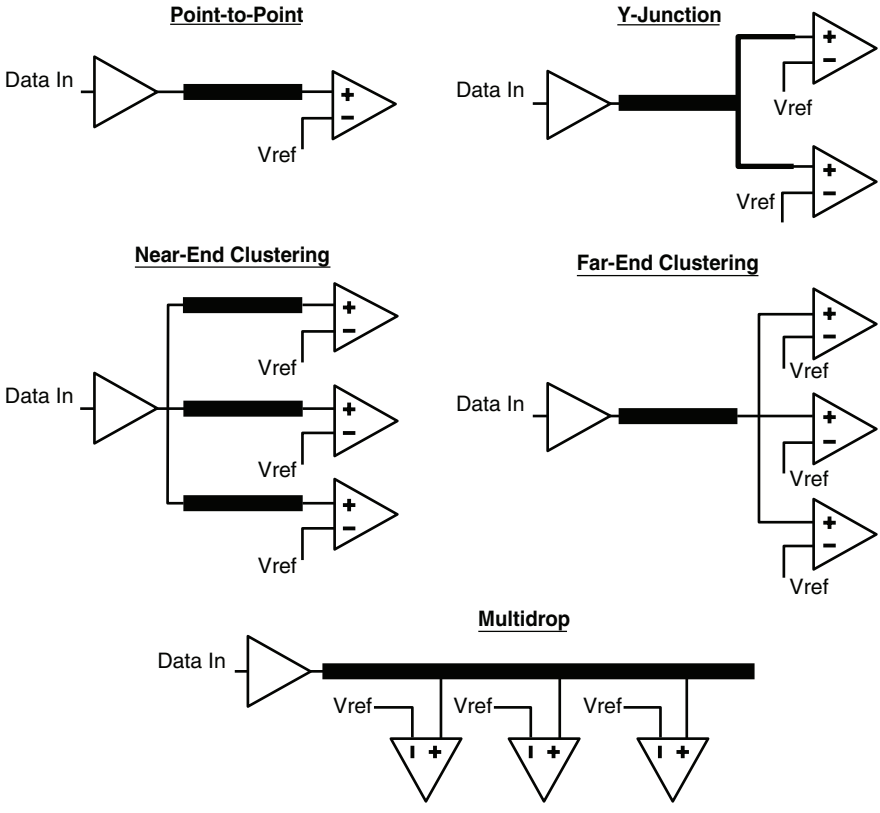


Figure 2.1 Various Signaling Topologies

2.1.2.1 Transmitter Output Impedance

Output impedance is the ratio of delta voltage and delta current at the operating point: $R_{out} = (\Delta V / \Delta I)$. Figure 2.2 shows the output IV characteristic of a transmitter when transmitting a logic 1. Each point on the curve has dynamic output impedance, which is a function of the operating point. When a device is operating in the linear region of the IV curve, R_{out} is constant and relatively small, making the device a voltage-mode driver. On the other hand, R_{out} is also constant, but large, when the device is in the saturation region, making the device a current-mode driver. If ΔV is so large that it encompasses both the linear and saturation regions, the output impedance is non-linear, and varies with the voltage.

2.1.2.2 Single-Ended versus Differential Transmitters

Single-ended signaling uses one wire to send signals from a transmitter to a receiver. In contrast, differential signaling requires two wires to send signals. From a pin-count perspective, the overhead of differential signaling is 2:1. However, this ratio is actually about 1.3 to 1.8, because the

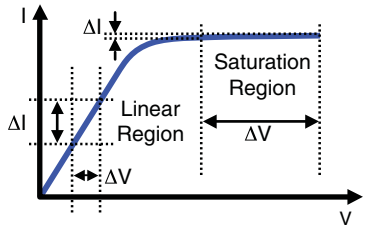


Figure 2.2 IV Characteristics of a Typical CMOS Transistor

signal-to-power/ground ratio, required for reliable operation at high speed, is smaller for differential signaling than for single-ended.

The main advantage of differential signaling is superior noise immunity. The receiver only sees the relative voltage, or voltage difference, between the two transmission lines in the differential pair, enabling it to reject common-mode influences, such as crosstalk, simultaneous switching noise, and power supply noise. The inherent noise immunity and high gain of differential circuits make them less sensitive to the attenuation of the signal in the transmission medium, allowing for reduced signal levels. Differential signaling also improves switching speed, reduces power dissipation, and mitigates the impact of noise in the digital system. Additionally, because the currents through the differential pair are complementary, there is little net magnetic flux. Consequently, the fields radiated by each signal are of opposite polarity, and cancel out dramatically, reducing the far-field electromagnetic radiation.

2.1.2.3 Voltage-Mode versus Current-Mode Transmitters

As previously stated, the major difference between voltage-mode and current-mode transmitters is their output impedance. Often, the voltage-mode driver requires less voltage, because it operates with low V_{ds} (see Figure 2.2), making it ideal for low-power interface designs. However, the current-mode transmitter offers better power-supply noise rejection, particularly for single-ended signaling, because the voltage-mode transmitter references a local ground. There is finite impedance (Z_{gnd}) between the local ground and the system ground, as shown in Figure 2.3. Z_{gnd} is a function of the power-distribution design on the die, package, and board. R_{out} is typically big for the current-mode case. The termination on the receiver side is not shown in the figure.

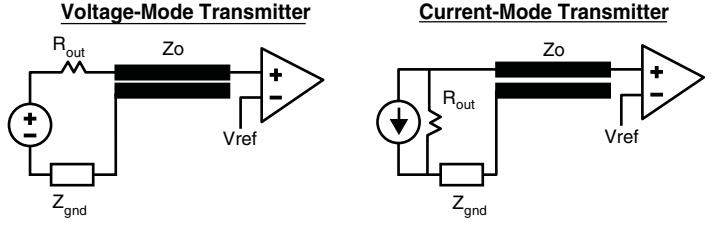


Figure 2.3 Equivalent Circuits for Voltage-Mode and Current-Mode Transmitters with Non-Ideal Grounds

With a voltage-mode transmitter, the voltage launched into the channel ($V(t)$) depends on Z_{gnd}, because $V(t)$ is a function of the voltage drop across Z_{gnd}, which is shared by transmitters on the same die. Coupling occurs through Z_{gnd} when a neighboring transmitter switches. With a current-mode transmitter, the current ($I(t)$) is independent of Z_{gnd}, because it is biased in the saturation region. We can observe this in Figure 2.2, where a small change in ΔV does not incur a large change in current in the saturation region, and has little impact on the signals launched on the channel. The current-mode transmitter has very good isolation from power supply noise.

Figure 2.4(a) is a commonly used single-ended current-mode driver, also called a high-common mode (HCM) driver [16]. Figure 2.4(b) is a single-ended voltage-mode driver, also called a low-common mode (LCM) driver, because it often uses a low-power supply. The voltage-mode driver has an additional pull-up device to maintain constant impedance all times. Because the current-mode driver is high-impedance, it does not require the additional pull-up. Figure 2.4(c) and Figure 2.4(d) illustrate differential versions of the current-mode and voltage-mode drivers, respectively.

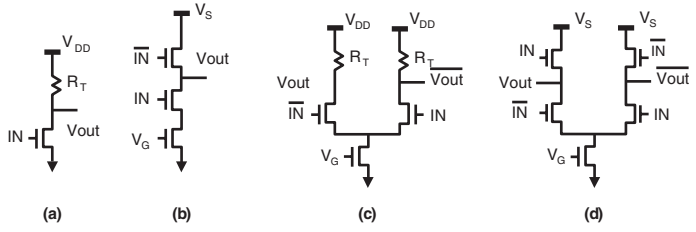


Figure 2.4 (a) Single-Ended Current-Mode, (b) Single-Ended Voltage-Mode, (c) Differential Current-Mode, and (d) Differential Voltage-Mode Transmitters

2.1.2.4 Transmitter Output Timing Jitter

Transmitter output timing jitter (t_Q) accounts for the uncertainty involved with transmitting data at a given time. In Figure 2.5, CLOCK and DATA are shifted by half of a CLOCK cycle (t_{CYCLE}). This is the ideal timing offset between CLOCK and DATA, and maximizes the sampling window at the receiver. t_Q represents the amount of time that CLOCK and DATA vary from this ideal timing relationship.

t_Q contains both static and dynamic components. The static component is caused by the skew between the clock and data signals, and the duty-cycle distortion (DCD) on either the clock or data signal. Employing correction circuitry and performing calibration procedures can reduce the static error. The dynamic component of t_Q is difficult to minimize. It is due to dithery clock, power-supply noise, ISI, and so on. Finally, t_Q is often specified at the package pin under an idealized test condition. Therefore, it includes any timing error due to the package, such as coupling and attenuation, in addition to device effects.

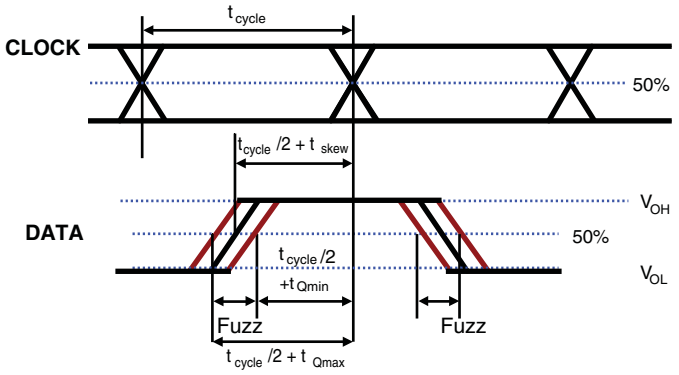


Figure 2.5 Definition of Transmitter Output Timing

2.1.2.5 Full-Circuit versus Behavior-Driver Models

A full transistor model always provides an accurate simulation response, but it requires a significant amount of time to simulate. Often, a simpler behavior model, such as IBIS, or voltage-controlled current source, is used for SI analysis [7–9]. Although these models provide reasonably good accuracy, one must consider potential limitations. In general, a current-mode driver fits better with behavior models, because it tends to be more linear than a voltage-mode driver is. Using a tabular model, behavior models (such as IBIS) can mimic the non-linear output waveform response. However, this approach often fails to model non-linear termination characteristics. Consequently, the simulation will lose accuracy when there is a significant reflection from the channel.

Additionally, with single-end drivers, power-supply noise can push an output driver into the non-linear region; a simple behavior model cannot model this phenomenon. Moreover, typical behavior driver models do not include pre-driver and crowbar currents [10]. The references at the end of this chapter include proposals for improved versions of behavior models [11] [12]. Chapter 12, "SSN Modeling and Simulation," discusses an alternative approach that uses a full-circuit driver in conjunction with current mirrors.

2.1.3 Receiver

The receiver's function is to translate the voltage waveforms received from the transmission medium or channel into logic 1s or 0s. A receiver is characterized by its input sensitivity (V_{IN}), which represents the voltage resolution, and by the set-up and hold times (t_S and t_H), which represent the timing resolution. V_{IN} consists of three components: an input voltage offset, which is static and can be corrected by offset cancellation techniques; receiver deadband, which is determined by receiver bandwidth, or ISI and power supply noise at the receiver; and random voltage noise, due to the thermal properties of the resistors and transistors. The second and third components are dynamic, or time varying, and generally hard to mitigate.

t_S and t_H define the receiver time resolution, or aperture time, required to sample the data reliably. For the system to operate reliably, the window defined by both V_{IN}, t_S, and t_H must reside completely within the data eye for all the worst cases. To understand this in more detail, examine the example shown in Figure 2.6. Here, the receiver window (lightly shaded) overlays the data's eye diagram. The height of the window is two times that of the input voltage requirement (V_{IN}). The width of the window defines the set-up and hold times (t_S and t_H). This window represents the worst case, as defined by process, supply voltage, and temperature variations (PVT). The receiving window (dark shaded area) must be smaller than, and reside completely within, the window defined by V_{IN}, t_S, and t_H.

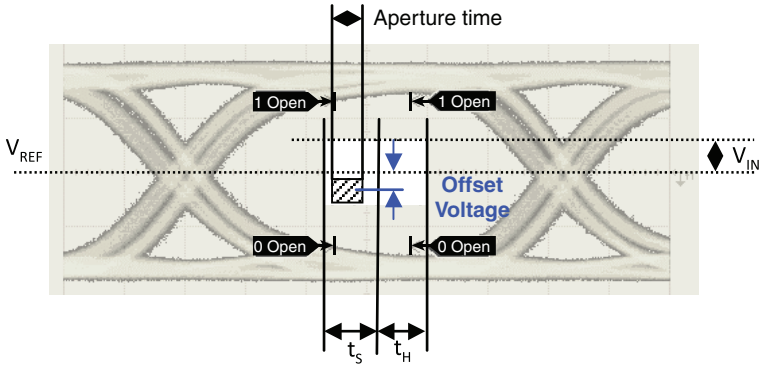


Figure 2.6 Eye Diagram Illustrating Receiver Input Voltage and Timing Requirements

It is important to note that the shape of the receiver window is a hexagon or diamond, rather than a rectangle, due to the finite edge rate of the input signals. This information may be useful when trying to determine whether a system would pass or fail for a given set of eye diagrams. A rectangular window is too pessimistic for multi-gigabit systems.

2.1.4 Terminators

Terminators absorb signals after they arrive at the receivers. If signals are not removed from the signaling system, multiple reflections and resonance will occur, severely distorting the signals under certain conditions. Often, when the data rate is below a few hundred megahertz, terminators are not used, which simplifies the design and minimizes power consumption.

Figure 2.7 illustrates five types of transmission-line-termination methods: series, parallel, RC network, Thevenin network, and Diode network.

Series termination places a resistor at the transmitter (Tx). If the sum of the terminator resistance (R_S) and the Tx output impedance (R_{out}) is equal to the transmission line impedance (Z_o), the channel is perfectly terminated at the source. The voltage launched into the channel is half of the Tx voltage, due to the resistor divider.

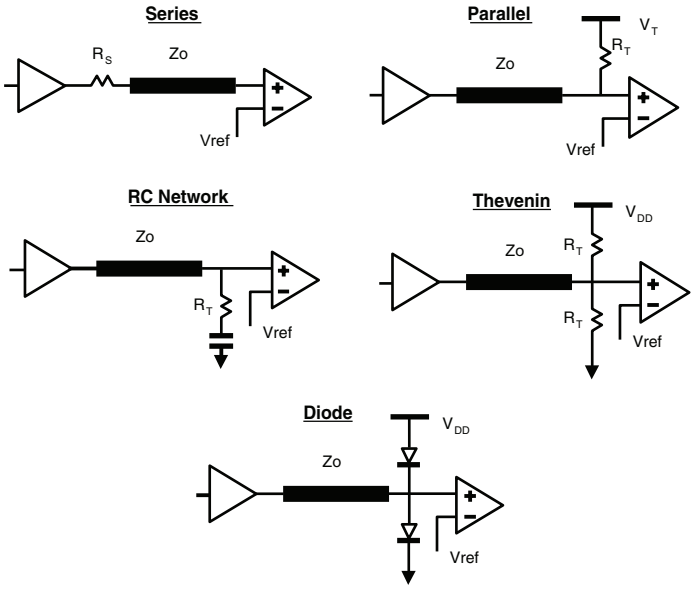


Figure 2.7 Common Termination Methods

Parallel termination places a resistor at the receiver (Rx). The resistor connects to a termination power supply (V_T), or to GND. When the resistor value (R_T) equals (Z_o), the channel is perfectly terminated at the load (Rx). R_T removes the signal from the channel, when it arrives at Rx. The main disadvantage of parallel termination is that it consumes DC power.

RC network termination places a resistor and a capacitor at the receiver (Rx). Similar to parallel termination, the termination resistor absorbs the signals during AC switching. The capacitor maintains the DC voltage level, thereby reducing DC power consumption. The resistor should be equal to Z_o, and the capacitor value should be chosen so that the RC constant is larger than twice the loaded propagation delay (200pF–600pF). Although RC network termination was popular in the past, it becomes less useful for multi-gigabit applications due to the large RC constant.

Thevenin network termination connects one resistor to power, and the other resistor to GND at the receiver (Rx). It is equivalent to parallel termination, but does not require a separate termination supply (and the resulting additional DC power consumption). During switching activities, the termination provides additional current, which reduces the burden on the driver, and makes it optimal for high-voltage swing applications, such as TTL logic. When used in CMOS systems, the resistor values must be carefully chosen, to obey the proper Rx switching voltage levels.

Diode network termination places two diodes at the receiver (Rx). Diodes do not absorb signal waveform energy as resistors do. Instead, they only limit the amount of overshoot and undershoot (commonly referred to as clamping). Signals are reflected back to the channel. For

high-frequency applications, a diodes network is not used alone, and must be combined with other termination methods.

2.1.4.1 On-Chip versus Off-Chip Termination

The termination network can be implemented either on-chip or off-chip (on-board). Electrically, a large difference exists between these two termination schemes, especially for very-high-speed applications. Three reasons exist as to why on-chip termination is better, in terms of signal integrity:

- With off-chip termination, the receiver package parasitic acts as a stub at high speed, so the signal is not entirely absorbed by the terminator.
- The terminator connects to the receiver package pin with a PCB trace. This connecting PCB trace also acts as a stub, and creates an impedance discontinuity.
- The off-chip (on-board) resistor has its own package parasitic. As a result, for multi-gigabit applications, on-chip termination is highly desirable.

An additional advantage of on-chip termination is that the termination value can be changed dynamically, which is useful for multi-rank systems, and it can be turned off during idle time to lower power consumption. On the other hand, on-chip termination varies significantly over process and temperature variations. Seeing at least +/–10% impedance variation for on-chip termination is common. For high-speed interfaces, calibrate on-chip termination periodically to account for temperature variations.

2.1.4.2 Single versus Double Termination

The termination network can be implemented either on one end (series or parallel termination), or on both ends (both series and parallel terminations). In many practical applications, a single termination is sufficient to remove signal reflections in the channel: The signal is reflected from one end and absorbed at the other. A single termination also significantly reduces power consumption, compared to a double termination. However, channel resonance can occur when large multiple reflections exist within the channel, and the electrical length of the channel is a multiple of a quarter of the wavelength. When using a double termination at both source and load, the signals are absorbed at both ends. The multiple reflections only occur within the channel. The receiver only receives the reflections of reflections. Hence, the signal integrity at the receiver is much improved. As mentioned earlier, for on-chip termination, the impedance variation is quite large. A double termination drastically reduces the reflection, without requiring a high precision termination. The trade-off, of course, is the cost of implementation with the performance goals.

2.1.5 Interconnect

Interconnects, or passive channels, play an important role in high-speed signaling when they are used to communicate between transmitters and receivers. In fact, interconnects become a major bottleneck as the frequency reaches multi-gigahertz. At a system level, a typical interconnect

consists of cables, connectors, PCB power/ground planes, PCB traces, PCB vias, package substrate traces, package vias, package wirebonds, package solder balls, and die solder bumps. As an example, Figure 2.8 shows a chip-to-chip interconnect system consisting of a wire-bond package on one side, and a chip-scale package (CSP) on the other side. PCB traces connect two chips together. At the chip level, the wiring connecting transistors is much smaller when compared to the system-level interconnect, and has very different electrical properties. To understand why interconnects are important to signal integrity, the following section provides an analysis of their electrical behavior based on simplified equivalent circuit models.

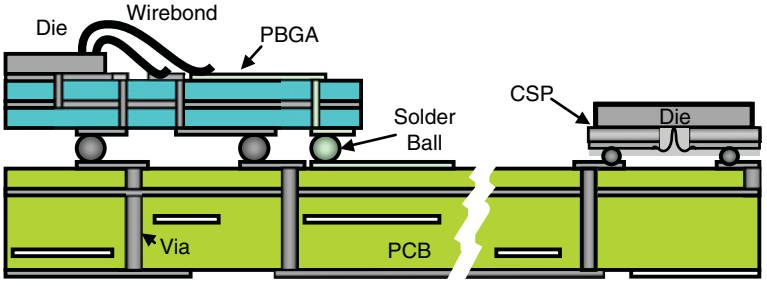


Figure 2.8 Chip-to-Chip Interconnect System

2.1.5.1 Distributed Model for Transmission Lines

For electrically long interconnects, such as cables, PCB traces, and package substrate traces, a distributed transmission line model is required to accurately represent their electrical behavior. (Chapter 5, "Transmission Lines," discusses transmission lines in detail.) This section discusses the basic characteristics of transmission lines.

Transmission lines have four important electrical behaviors to keep in mind. One is that a finite signal propagation delay exists for a given length of the line. For example, the delay is roughly 150 ps/in for PCB micro-stripline, and 170 ps/in for PCB stripline. Another is that signal reflection occurs if the termination impedances do not match the characteristic impedance of the line. The reflection coefficient is determined by

$$\Gamma = \frac{Z_o - Z_L}{Z_o + Z_L} \tag{2.1}$$

where Z_L is the load impedance. When the length of the line is a multiple of a quarter of the wavelength, a standing wave is built up on the line. The reflected waveform interferes with the incidence wave constructively at certain points, and destructively at other points. This condition is referred to as resonance. Another behavior is signal attenuation exists when the resistance (due to

conductor loss) or the conductance (due to dielectric loss) is not negligible. Finally, signal dispersion occurs when the propagation constant γ is a function of frequency. Specifically, the high-frequency component of the signal is attenuated more than the lower frequency components. The signal waveform changes shape as it propagates along the line. As a result, inter-symbol interference (ISI) occurs; in other words, the current signal waveform (or bit) is distorted due to signals (bits) transmitted previously.

2.1.5.2 Lumped Model for Interconnect Discontinuities

A π- or T-lumped equivalent circuit model (as shown in Figure 2.9) can be used to represent the behavior of electrically short interconnects, such as package wirebonds, package/PCB vias, package solder balls/bumps, and connector pins. However, the accuracy of such lumped models deceases as frequency increases. For one thing, they do not capture frequency-dependant loss, due to conductor and dielectric. Also, they do not capture the distributed behavior. To ensure the accuracy of the model shown in Figure 2.9, the electrical size should be much smaller than the wavelength of the highest frequency of interest. That is, $l/\lambda_{\min} \ll 1$, where l is the length of the line, and $\lambda_{\min} = 1/f_{\max}$. Typically, a factor of 10 is used.

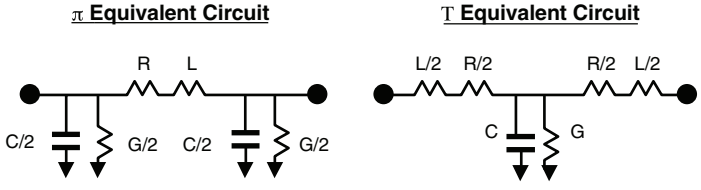


Figure 2.9 π- and T-Lumped Equivalent Circuit Model for Electrically Short Interconnects

How does a short interconnect affect signal integrity? The two basic properties to examine are delay and reflection. The propagation delay is $t_d = 1 \Big/ \sqrt{LC}$. Even though the individual delay may be small for many structures, such as solder balls and vias, they can accumulate along a long interconnect path. To minimize skew between different signal lines, the propagation delay, due to various interconnect structures, needs to be taken into account. Reflection is caused by a difference in impedance from the intended transmission line characteristic impedance. The effective impedance is $Z_{eff} = \sqrt{L/C}$. From (Equation (2.1), earlier), it is clear that the amount of reflection is proportional to the difference of Z_{eff} and Z_o. Much of the interconnect design effort is directed towards optimizing the physical design, so that its impedance (Z_{eff}) matches the transmission line characteristic impedance (Z_o).

2.1.6 Clocking

For the system to operate reliably, a method must exist for coordinating when the transmitter sends data, and when the receiver receives data. This method is referred to as clocking or timing conventions. The four classes of clocking methods are as follow:

- **Asynchronous:** No clock is used. This method uses handshake mechanisms to ensure the correct sequencing of events.
- **Synchronous:** Every component gets the same clock frequency and known phase.
- **Mesochronous:** Every component gets the same clock frequency, but an unknown clock phase.
- **Plesiochronous:** Every component gets almost the same clock frequency, and a slowly drifting clock phase.

Commonly used clocking architectures are common (synchronous) clock, source synchronous clock, forwarded clock, and embedded clock. The common clock timing method is simple and widely used, but its use is limited to frequencies below 300MHz. The source synchronous timing method is used in high-performance systems and has no theoretical frequency limit. The forwarded clock tracks timing variations due to supply noise and temperature. The embedded clock not only tracks out transmitter jitter, but also removes the need to route clock wires in the system. However, the tracking bandwidth is typically more limited than that of the forwarded clock. Chapter 10, "Clock Models in Link BER Analysis," discusses the advantages and disadvantages of the different clocking schemes and provides simulation models.

2.2 Noise Sources

The robustness of a high-speed digital system is largely dependent on the noise impact from various sources. The signal at the receiver is the sum of the intended signal, plus all the unintended noise. One can increase the signal-to-noise ratio, reject the noise, or both, to transmit information reliably. Understanding the physics of noise sources and mitigating their impact is the key to robust high-speed signaling.

Two categories of noise sources exist in digital systems. The first category of noise source affects the silicon directly (transmitter, receiver, clock, and terminator). These sources include power supply noise, transistor device noise, alpha particles, thermal (Johnson) noise, shot noise, flicker noise (or $1/f$ noise), process variations, electromagnetic interference (EMI), and so on. The second category of noise source affects the passive channel directly. These sources include attenuation (or loss), crosstalk, reflections and/or resonances, inter-symbol interference (or ISI), and so on.

Self-induced noise is noise whose magnitude is proportional to the signal magnitude (that is, power-supply noise, crosstalk, reflections, and EMI). Self-induced noise cannot be mitigated by simply increasing signal magnitude. On the other hand, noise sources that are due to thermal noise, or process variations, are independent of the system's switching activities.

Noise sources can also be classified as random or deterministic noise. A device's thermal noise, or process variation, are examples of random noise. Random noise can be described by an unbounded Gaussian distribution. The impact of random noise on system performance is negligible when the data rate is less than 3Gb/s. However, it becomes important for data rates higher than 3Gb/s, and especially, for data rates higher than 5Gb/s. Noise sources that affect the channel (that is, ISI or crosstalk) are deterministic. In contrast to random noise, deterministic noise is bounded.

This section primarily focuses on noise sources that signal-integrity engineers have direct influence over, namely, power-supply noise and noise on the channel. The section covers the underlying physics of the noise sources and illustrates how they can result in signal degradation, ultimately reducing the channel voltage and timing margins. Although the fundamental noise mechanism remains the same, the relative importance of various noise sources varies, depending on the signaling methods used in a particular system. For example, crosstalk could be the dominant noise source in a single-ended signaling system, whereas ISI could be the dominant noise source in a differential signaling system. The challenge to signal-integrity engineers is to perform accurate and detailed modeling and analysis of the specific system, and determine the relative importance of various noise sources. Please refer to Chapter 6 of *Digital System Engineering* [1] for a more detailed analysis of device noise, process variation, and EMI.

2.2.1 Attenuation or Loss

The signal attenuation, or loss, in the channel, is due to three major components: device parasitic capacitance (C_i), transmission line dielectric loss, and transmission line conductor loss.

Figure 2.10 shows an equivalent circuit for an example channel. In the figure, R_i represents the substrate loss (which is typically less than 10Ω). C_i represents the parasitic capacitance from various sources (such as ESD structure, termination network, driver and receiver transistors, and routing metals). C_i is a low-pass filter for high-frequency signals. The RC constant, formed by the termination, determines the rise time of the signal that can be launched into the channel. To understand how C_i attenuates high-frequency signals, one must analyze the scattering parameters for the R_iC_i network, which is determined by

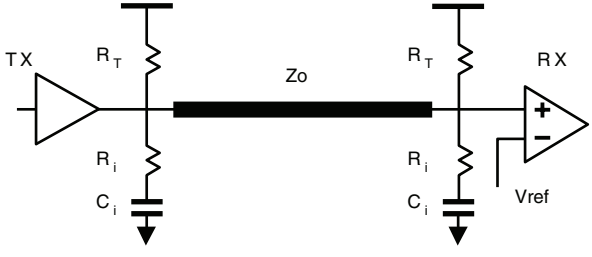


Figure 2.10 Transmitter and Receiver Parasitic Capacitance Model

$$S_{11} = S_{22} = \frac{-Z_0 Y}{2 + Z_0 Y} \tag{2.2a}$$

$$S_{12} = S_{21} = \frac{2}{2 + Z_0 Y} \tag{2.2b}$$

where Z_o is the characteristic impedance, and

$$Y = \frac{j\omega C_i}{1 + j\omega C_i R_i}. \tag{2.3}$$

If Z_o is equal to 50Ω, and R_i is equal to 10Ω, S_{12} approaches 0.28. This means that 70% of the signal is lost, due to just the parasitic capacitance C_i. The value of C_i determines the amount of signal lost at a given frequency, before approaching the asymptote value determined by the R_i value. This example illustrates the critical role of C_i in channel loss at high frequencies. Reducing C_i is a must for multi-gigabit data rates. The value of C_i is a function of the process and the silicon design. The ESD parasitic capacitance is typically 30%–40% of C_i. C_i decreases as device geometry shrinks. For example, the value of C_i is 3pF–5pF for a 130-nm process. C_i can be reduced to 1.0pF–2pF for a 65-nm process.

The physical medium that connects the transmitter and receiver consists of many different parasitic elements and transmission lines, as illustrated in Figure 2.8. The primary loss mechanism on the physical medium is due to loss in the transmission line; the propagation constant (γ), and the characteristic impedance (Z_o) characterize this. The propagation constant is a complex number determined by

$$\gamma = \alpha + j\beta. \tag{2.4}$$

The signal wave on the transmission line is in the form of $e^{-\gamma z}$. The real part (α) is referred to as the attenuation constant representing the loss of signal. When α is larger than zero, the signal wave is decaying exponentially.

The four types of transmission-line losses are dielectric loss, conductor loss, radiation loss, and leakage loss. The attenuation constant is determined by

$$\alpha = \alpha_c + \alpha_d + \alpha_l + \alpha_r \tag{2.5}$$

where α_c is the attenuation constant for conductor loss, α_d is the attenuation constant for dielectric loss, α_l is the attenuation constant for leakage loss, and α_r is the attenuation constant for radiation loss. For data rates below 20 Gb/s, the losses due to radiation and leakage are negligible.

The dielectric loss, due to material properties, is the second major source of signal loss. The dielectric loss for a material is characterized by the loss tangent. This is determined by

$$\tan\delta = \frac{\varepsilon''}{\varepsilon'} \tag{2.6}$$

where $\varepsilon = \varepsilon' - j\varepsilon'' = \varepsilon_r(1 - j\tan\delta)\varepsilon_0$, $\varepsilon_0 = 8.854\times10^{-12}$. The physics behind dielectric loss is due to electrical polarization [2]. An ideal dielectric material is one that has no free charges. However, all material consists of molecules, which in turn, are made of charged atoms and electrons. When a time-varying electric field is impressed onto a material, molecules inside the material align in the direction opposite that of the applied electric field. Hence, the material is polarized, with the negative and positive parts of the molecule displaced from their equilibrium positions. The time-varying electric field causes the molecules to vibrate. The vibration of the molecules generates heat and dissipates electric energy. The attenuation dielectric constant for inhomogeneous material is determined by

$$\alpha_d = \frac{f\tan\delta\pi\sqrt{\varepsilon_r}}{c} \tag{2.7}$$

where c is the speed of light in a vacuum. For a transmission line with inhomogeneous material, the attenuation constant has a similar form, except that the dielectric constant is replaced by an effective dielectric constant ε_r [2], which is the ratio of capacitance with and without dielectric present.

Regarding (2.7), it is important to note that the dielectric loss linearly increases with signal frequency and loss tangent. The most common PCB material is FR4, which has a loss tangent of 0.02. To reduce signal loss, you can use a lower loss tangent material, such as Rogers 4350, but the penalty is higher cost. This may not be acceptable for consumer products, but it may be feasible for the backplanes of computer servers or Internet routers. Furthermore, the dielectric constant and loss tangent may be functions of frequency. They may also be functions of temperature and humidity. For example, the FR4 loss tangent may increase to 0.03 at high temperature, while the dielectric constant can increase by 20% in higher humidity.

The third major source of signal loss is due to transmission line conductor loss. The DC resistance of a conductor is determined by

$$R_{dc} = \frac{\rho l}{A} \tag{2.8}$$

where ρ is the resistivity, l is the conductor length, and A is the cross-sectional area. The internal impedance of the conductor is a function of frequency

$$Z_s = R_s(1+j) \tag{2.9}$$

where the surface resistivity R_s is determined by

$$R_s = \frac{1.0}{\sigma\delta} = \sqrt{\frac{\pi f\mu}{\sigma}}. \tag{2.10}$$

And, the skin depth δ is determined by

$$\delta = \frac{1.0}{\sqrt{\pi f\mu\sigma}} = \sqrt{\frac{2.0}{\omega\mu\sigma}}. \tag{2.11}$$

It is important to note that AC conductor loss is proportional to the square root of the frequency. For coaxial cable, the attenuation constant is obtained by

$$\alpha_c = \frac{0.014272\sqrt{f}}{Z_0}\left(\frac{1}{d} + \frac{1}{D}\right) \quad (dB/m) \tag{2.12}$$

where Z_o is the characteristic impedance, d and D are the diameters of the inner and outer conductor, respectively. For a microstrip line, the closed form equation for attenuation constant is more complicated (refer to page 96, [2]).

The surface roughness of the conductor induces additional loss. The attenuation constant for a microstrip line is modified [3] as follows:

$$\alpha_c = \alpha_{c0}\left(1.0 + \frac{2.0}{\pi}\tan^{-1}\left[1.40\left(\frac{\Delta}{\delta}\right)\right]\right) \tag{2.13}$$

where α_{c0} is the attenuation constant for a perfectly smooth conductor, Δ is the rms surface roughness, and δ is the skin depth. When the ratio of surface roughness to skin depth reaches 2, the increased conductor resistivity could reach 75%.

It is evident from the previous discussion that signal loss, due to C_i and the transmission line, depends on signal frequencies. For a PCB trace length of less than 6 inches, signal loss due to C_i dominates. In contrast, for an 18-inch trace with FR4, the dielectric loss dominates. Conductor loss becomes much more significant when loss dielectric material, such as Rogers, is used.

2.2.2 Crosstalk

Crosstalk noise (or coupled noise) is a generic term that refers to signal-to-signal coupling, signal-to-supply coupling, or supply-to-signal coupling. Crosstalk is the direct result of the physical design of the chip, package, or PCB, as well as the connector where the electromagnetic coupling occurs. On-chip crosstalk is primarily capacitive (that is, electric field coupling). In contrast, crosstalk due to package wirebonds is primarily inductive (that is, magnetic field coupling). In general, particularly on transmission lines, both capacitive and inductive couplings exist. Crosstalk noise can be classified as either near-end crosstalk (NEXT), or far-end crosstalk (FEXT). NEXT refers to the

crosstalk noise observed at the side where the signal is sent. FEXT refers to crosstalk noise observed at the side where the signal is received. Chapter 5 discusses NEXT and FEXT in detail.

2.2.3 Reflections and Resonances

Reflections occur when impedance discontinuities exist in the transmission line. Impedance discontinuity often occurs when a mismatch exists in the physical dimensions of the vias, BGA balls, and connectors. Moreover, the manufacturing variations in the physical dimensions (such as dielectric thickness, trace width, and spacing) all contribute to impedance variations. For low-cost high-volume manufacturing, having +/–10% to +/–15% impedance variation in PCBs, and +/–20% impedance variation in the package impedance, is common.

2.2.4 Inter-Symbol Interference (ISI)

Inter-Symbol Interference (ISI) is not an independent noise source: It is the result of channel frequency dependent loss and reflections. As an example, Figure 2.11 shows a single-bit response at the receiver for a long backplane channel. Each circle in the plot indicates the center of the bit. As shown from this plot, the single-bit pulse is widened to occupy a bit before (or precursor) and four bits after (post cursors), due to the large channel frequency dependent loss. As in the case of reflections, the residue waves of past bits are superimposed onto the current bits. From the perspective of single-bit response, the ISI effect is the same, whether it is due to loss or reflections. However, reflections can occupy many more bits than attenuation for a long and low-loss channel.

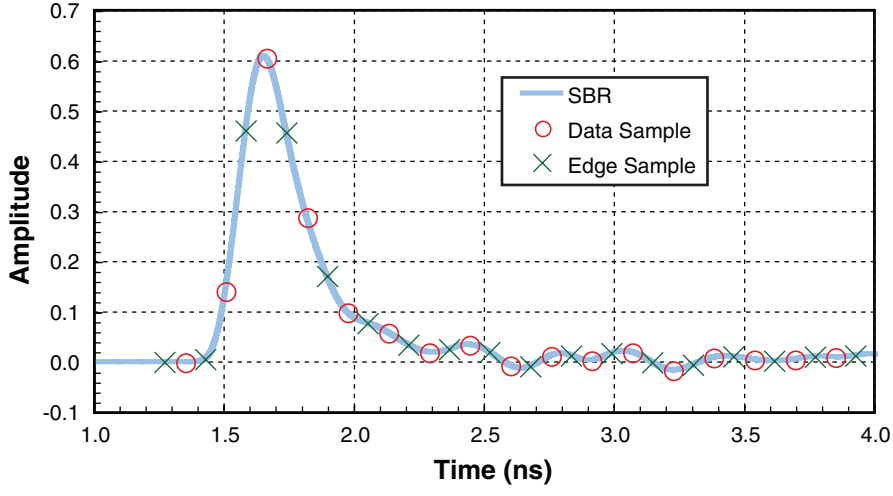


Figure 2.11 Single-Bit Response of a Backplane Channel

To illustrate how ISI generates bit errors, assume that Tx transmits a 101 pattern, as shown in Figure 2.12. Because the channel is a linear time invariant (LTV) system, the final wave at the

receiver is simply the superposition of time-shifted single-bit pulses. Due to the post cursors of the first single bit 1 pulse, and the precursors of the second single bit 1 pulse, the 0 bit becomes a 1 bit, resulting in a single-bit error.

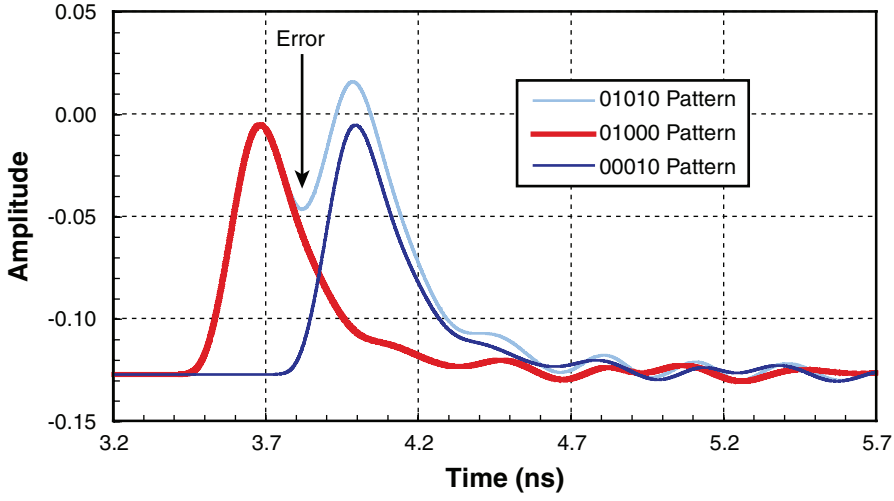


Figure 2.12 Large ISI-Induced Bit Errors

2.2.5 Power Supply Noise

Power supply noise, and its impact on link jitter, is one of the major bottlenecks in high-speed interfaces. The power distribution network (PDN) must be carefully designed, in order to limit power noise. Excellent textbooks have recently been published on the subject of designing power distribution networks [4–6]. However, their impact on link jitter has not been covered. Part III of this book addresses the topic of jitter induced by supply noise in detail. However, a few key issues related to power supply noise are introduced here.

2.2.5.1 Power Distribution Network (PDN) and Bypass Capacitors

A bypassing capacitor can provide the transient charges needed by the switching devices on silicon and reduce the PDN impedance. A power distribution system consists of three different hierarchies: silicon, package, and PCB. At each hierarchy, capacitors suppress supply noise at different frequency ranges. For example, the capacitors placed on the silicon suppress very high frequency noise (>500MHz). Modern high-speed designs must have on-chip decaps for reliable operation. Many system-level designers often neglect this on-chip capacitance and overdesign the PDN. Similarly, low to medium frequency noise is suppressed by capacitors of different sizes and types placed on the package and PCB. It is important to note that capacitors on the PCB are less effective at high frequencies, because of their parasitic inductance. Capacitors placed on the

package are more effective due to lower parasitic inductance. Package PDN inductance is often larger than that on the PCB. This is particularly true for wirebond packages. Furthermore, package inductance, in a power distribution network, interacts with the on-die bypass capacitors to create a resonance in the effective impedance of the power distribution network. Supply noise is the largest when a large switching current exists at resonant frequencies. Figure 2.13 plots typical PDN impedance versus frequency, where the package resonant frequency occurs most often between 100MHz–300MHz. The design goal is to set the target impedance of the PDN below a certain limit over a broad frequency band, given the amount of switching current and allowed voltage variations (including both DC drop and AC noise).

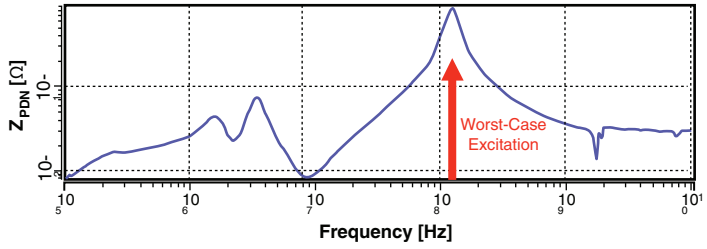


Figure 2.13 Effective Impedance of a Power Distribution Network

2.2.5.2 Power Supply Noise and On-Chip Circuitry

Unlike noise sources, such as crosstalk and reflections (which directly influence the signals on the channel), supply noise directly affects the circuit operations in terms of functionality and performance. Supply noise could degrade system performance in several different ways: First, supply noise increases signal delay uncertainty in the design. Signal delay could increase when the supply voltage falls below the nominal voltage, or decrease when supply voltage rises above the nominal supply voltage. The net effect of power supply noise on the propagation of clock and data signals is an increase in delay uncertainty, and the maximum delay of the data path. For example, +/–10% variation in supply voltage could cause –/+15% variation in the delay of clock buffer and clock distribution. This, of course, depends on the process and circuit design. As a result, power supply noise imposes a limit on the maximum frequency of the design.

Second, supply noise increases on-chip clock jitter. In a typical high-speed link, a phased-locked loop (PLL) generates the on-chip clock, based on a reference clock input. A PLL consists of a phase detector, charge pump, loop filter, VCO, and frequency divider (as shown in Figure 2.14). Supply voltage variations in these circuit blocks result in timing jitter on the PLL clock output. In particular, the supply noise in the VCO dominates the timing jitter contributions. Furthermore, supply noise in the clock buffers and distributions changes the clock delay dynamically, as previously described, introducing additional timing jitter. Figure 2.15 plots the PLL noise transfer functions versus frequency for various circuit blocks. Specifically, the noise in the reference clock is a low-pass filter, because the PLL feedback path completely tracks the input

phase noise, resulting in the same phase noise at the output. On the other hand, the noise in the clock buffer is a high-pass filter; it is directly routed to the output and the feedback path provides no filtering. The noise in the VCO is a band-pass filter, with the low frequency phase noise tracked out slowly (due to the integration effect) by the loopback path, and the high-frequency phase noise filtered out by the VCO supply noise sensitivity.

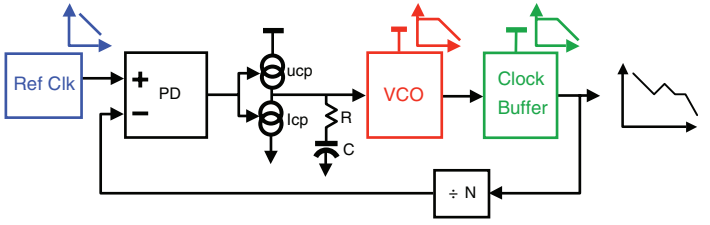


Figure 2.14 PLL Jitter Model

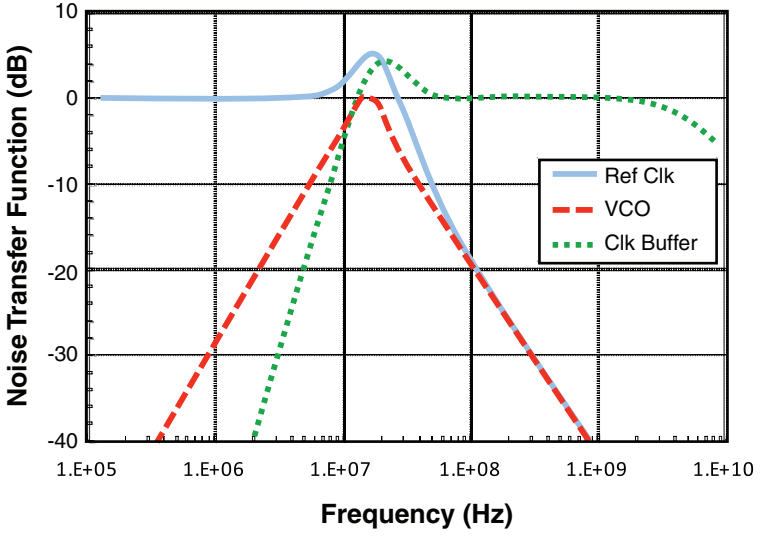


Figure 2.15 PLL Noise Transfer Functions

Third, supply noise could reduce circuit-noise margin and headroom. Verifying a circuit's supply-noise sensitivity is one of the major design goals for critical circuit blocks. A typical circuit design goal is to ensure robust operation under the worst case +/– 10% supply variations from a nominal supply voltage. This variation includes both DC (IR drop) and AC (dynamic noise) components. If the supply noise exceeds the designed target, such as at the worst-case PDN resonance, the circuit may not have enough voltage headroom to function properly and could fail. Unfortunately, this design goal is increasingly difficult to achieve. Chapter 14, "Supply Noise and Jitter Characterization," discusses this topic in detail.

Finally, supply noise can degrade gate oxide reliability. The gate oxide thickness is reduced, as process geometry scales, in order to improve device performance and to reduce power consumption. However, thin oxide poses the risk of electron tunneling and reliability. Particularly, high-supply voltages increase the stress on thin oxide, and reduce its long-term reliability. As a result, the overshoot in power supply voltage must be limited to minimize the risk of device breakdown.

2.3 Jitter Basics and Decompositions

Jitter characterization and simulation have become an integral part of high-speed link design. This section reviews jitter basics and decomposition and also categorizes the noise and jitter sources mentioned in the previous section, and applies them to different jitter types. M. P. Li's book is recommended for further reading on the subject of jitter in high-speed links [13].

2.3.1 Jitter Representations

Jitter is defined as the short-term variations of a digital signal's significant instants from their ideal positions in time [14]. Timing variations that occur slowly (less than 10Hz) are called *wander* or *drift*, whereas jitter describes more rapid variations. Jitter is measured using various devices, such as real-time or equivalent-time (sampling) oscilloscopes, time-interval analyzers (TIA), or a spectrum analyzer. Different measurement devices report jitter using different jitter representations. In this section, six commonly used jitter representations are reviewed: jitter sequence, jitter spectrum, autocorrelation, power spectrum density (PSD), probability density function (PDF), and phase noise.

2.3.1.1 Jitter Sequence and Spectrum

Jitter sequence is simply the sequence of the amount of jitter for a given data pattern. Figure 2.16 shows an example of data pattern along with some possible deviations. In this example, Δt_1, Δt_2, Δt_3, Δt_4, Δt_6, Δt_7, and Δt_9 represent the jitter sequence. As shown in this example, some of the jitter terms are missing, depending on the data pattern. For a clock pattern, all jitter terms are present. Jitter sequence is also referred to as Time Interval Error (TIE), or *phase jitter*. Jitter sequence can be plotted, in order to show jitter characteristics. Figure 2.17 shows a sinusoidal jitter in a clock signal, where the vertical axis is the jitter amount in time.

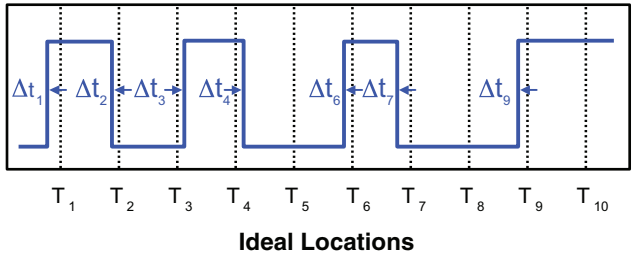


Figure 2.16 Jitter Sequence of the Sample Data Pattern

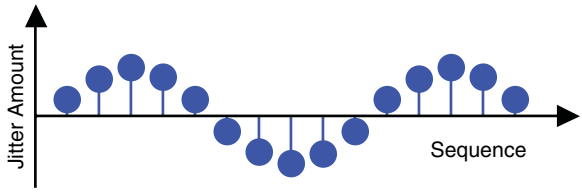


Figure 2.17 Plot of a Sinusoidal Jitter Sequence

For a clock pattern, *period jitter* is defined as the period deviation from the ideal period. The period jitter can be calculated by taking the difference of the jitter sequence. *Cycle-to-cycle jitter* is defined as the difference between two consecutive periods. Cycle-to-cycle jitter is calculated by taking the difference of period jitter. A more general definition for *n*th-cycle jitter is the difference between any two periods within *n* cycles. The *n*th-cycle jitter concept is used to define the jitter spec, to control the jitter spectrum contents. For instance, by increasing the order, more low-frequency jitter components can be accounted for.

Figure 2.18 shows phase, period, and cycle-to-cycle jitter measurements for sinusoidal jitter. As shown in the figure, the period jitter filters the significant low-frequency jitter components. Both the period and cycle-to-cycle jitter measurements significantly filter low-frequency jitter components, but do not filter the high-frequency jitter components. It is interesting to note that, even though the cycle-to-cycle jitter measurement filters the low-frequency content more than the periodic jitter measurement, it slightly amplifies the high-frequency jitter measurement.

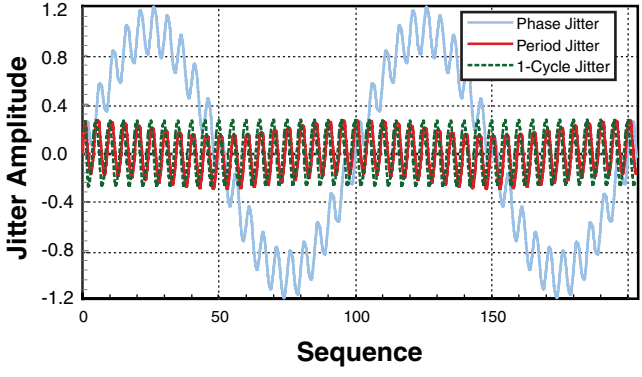


Figure 2.18 Various Jitter Measurements for a 1-GHz Clock with Sinusoidal Jitter of Magnitudes of 1.0 (at 10MHz) and 0.2 (at 200MHz)

Jitter sequence in the time domain can be converted to jitter spectrum in the frequency domain by applying a Fourier transform. Conversely, jitter sequence can be generated from jitter spectrum by using an inverse Fourier transform. For a jitter sequence generated from a data

pattern, not all the jitter terms are defined, as shown in Figure 2.16. To find the jitter spectrum for this case, the missing jitter terms can be interpolated before applying the Fourier transform.

Although jitter sequence can be measured in the time domain, jitter spectrum cannot be directly measured in the frequency domain. Consequently, jitter spectrum has a limited usage in practice. Nonetheless, it can be useful for a link-jitter simulator, as it provides an efficient way to simulate the jitter response by avoiding the time-domain convolution of the jitter sequence with the jitter transfer function.

2.3.1.2 Power Spectrum Density (PSD) and Autocorrelation

Power spectrum density (PSD), rather than jitter spectrum, is often used to characterize a stochastic process. PSD is directly measured using a spectrum analyzer with phase detector [15]. The PSD frequency plot shows various jitter characteristics. The jitter spikes at the multiples of data frequencies are data-dependent jitter. On the other hand, the jitter spikes at frequencies other than the data frequency are periodic jitter. Furthermore, the noise floor, after removing the jitter spikes, represents the random noise.

PSD is also related to the autocorrelation of jitter sequence through the Fourier transform:

$$S_{\Delta t}(f) = \int_{-\infty}^{\infty} R_{\Delta t}(t) e^{-j\omega t} dt \tag{2.14}$$

where

$$R_{\Delta t}(\tau) = \int_{-\infty}^{\infty} \Delta t(t + \tau) \Delta t(t) dt. \tag{2.15}$$

$\Delta t(t)$ is the TIE, or phase jitter, of the clock signal, and τ is the delay time. Hence, PSD or autocorrelation $R_{\Delta t}(\tau)$ can be calculated from TIE or jitter sequence data.

2.3.1.3 Phase Noise

Radio frequency (RF) designers studied jitter before it became a serious issue for digital designers. The RF community commonly uses the phase noise in place of phase jitter. This section reviews the differences between phase noise and phase jitter, or jitter sequence and PSD.

Phase noise is not voltage noise: It is timing jitter representing the phase deviation of the clock signal. To observe the difference between jitter and phase noise, consider the sine wave signal:

$$X(t) = \sin(2\pi f_0 t + \varphi(t)) \tag{2.16}$$

where f_0 is the nominal frequency, and $\varphi(t)$ is the phase noise or variations. To see the relationship to jitter ($\Delta t(t)$) the preceding equation is rewritten as

$$X(t) = \sin(2\pi f_0 (t + \frac{\varphi(t)}{2\pi f_0}). \tag{2.17}$$

Now the phase jitter and phase noise is related with

$$\Delta t(t) = \frac{\varphi(t)}{2\pi f_0}. \tag{2.18}$$

Phase noise, as used in RF applications, is often dealt with in the frequency domain rather than the time domain. The relationship of the PSD of phase noise to the PSD of jitter is determined by

$$S_{\Delta t}(f) = \frac{1}{(2\pi f_0)^2} S_{\varphi}(f). \tag{2.19}$$

The units of $S_{\Delta t}(f)$ and $S_{\varphi}(f)$ are sec^2/Hz and rad^2/Hz, respectively. If phase jitter is normalized to a unit interval (UI), the scaling factor simplifies to $1/(2\pi)^2$. RMS jitter is one of the popular jitter-measurement parameters and it is defined by

$$J_{RMS} = \frac{1}{2\pi f_0} \sqrt{\int_{f_L}^{f_H} S_{\varphi}(f) df}. \tag{2.20}$$

2.3.1.4 Jitter Probability Density Function (PDF) in Statistical Domain

Jitter can also be analyzed in the statistical domain by using histograms, probability-density functions (PDF), or cumulative-density functions (CDF). The histogram is created by plotting the frequency of occurrence versus the range of values of a particular parameter of interest. The height of the histogram represents the cases that fall within a particular unit interval. A PDF is then calculated by normalizing the histogram, by dividing the number cases within each interval by the total number of cases.

Similarly, one can obtain a timing histogram by binning edges at a reference voltage in the eye diagram, as shown in Figure 2.19. One can collect many statistics from the histogram plot, such as mean, median, standard deviation, and peak-to-peak values. If the eye diagram is obtained by averaging (as in Agilent's digital sampling scope [DCA] eyeline mode), the histogram removes RJ. Otherwise, both DJ and RJ are contained in the histogram plot. Histograms complement eye diagrams, and offer further insight into jitter components.

Although PDF provides a convenient way to describe jitter distribution, it is a rather simplified version of jitter representation, because it loses both time and frequency information. Hence, it cannot be converted to other, more general, jitter representations, nor can it be used to simulate accurate jitter amplification or tracking. As discussed in Chapter 6, "Channel Voltage and Timing Budget," link analysis based on statistical PDF models often results in pessimistic results when compared to more general models (such as jitter sequence, jitter spectrum, or PSD).

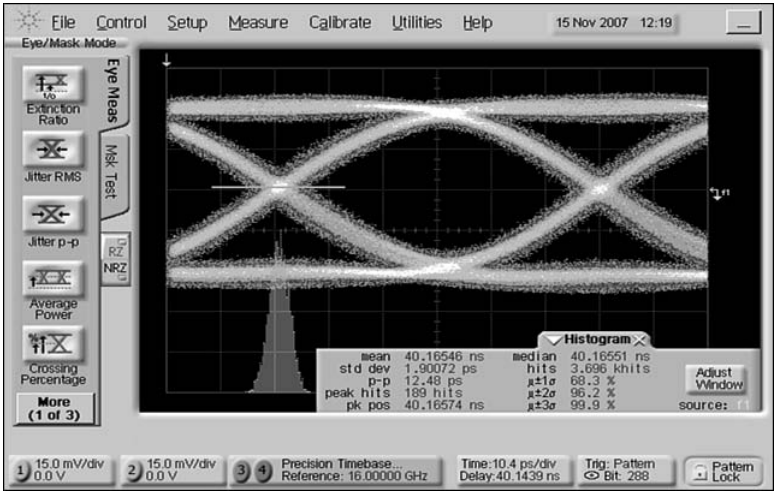


Figure 2.19 Sample Histogram Generated from an Eye Diagram at 16Gb/s

2.3.2 Jitter Decompositions

To facilitate identifying and understanding the sources of jitter, jitter is often decomposed into subcomponents, as shown in Figure 2.20. Specifically, the jitter is separated into two large categories: deterministic jitter (DJ) and random jitter (RJ). The DJ can be decomposed into data-dependent jitter (DDJ), and bounded uncorrelated jitter (BUJ). The duty cycle distortion (DCD) is

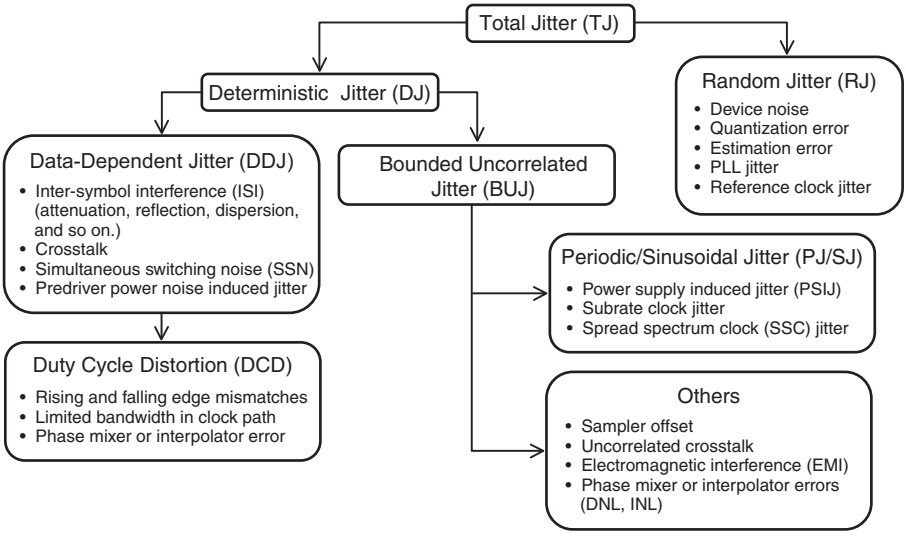


Figure 2.20 Jitter Component Decompositions

an example of DDJ, whereas periodic jitter (PJ) is an example of BUJ. Sinusoidal jitter (SJ) is a single-tone (frequency) case of PJ. Modern test equipment, such as Agilent's Digital Communication Analyzer (DCA-J), performs jitter decomposition as part of the post processing of measured time domain waveforms. Figure 2.21 shows a sample output from Agilent's DCA-J. It is important to keep in mind that the jitter separation method is an approximation, based on a mathematical model. The balance of this section provides short descriptions of the various jitter components.

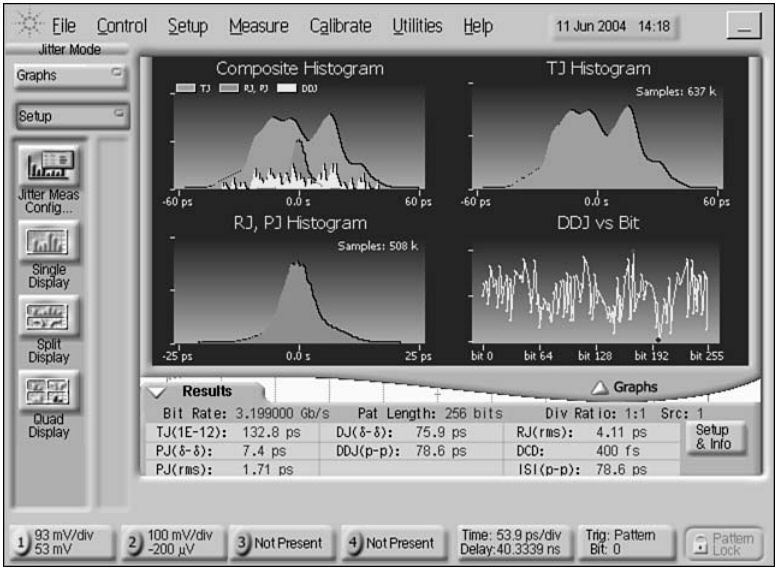


Figure 2.21 Jitter Decompositions Performed by Agilent's DCA-J

The deterministic jitter (or DJ) is repeatable, predictable, and bounded. DJ can be quantified in terms of a peak-to-peak value. As shown in Figure 2.20, DJ consists of DDJ and BUJ. DDJ is defined as any jitter components that are correlated with the data pattern in the link. Passive channel introduces several non-idealities that cause DDJ. These include channel attenuation, crosstalk, and reflections. Both inter-symbol interference (ISI) and duty cycle distortion (DCD) contribute to DDJ. (ISI was previously discussed in Section 2.2.4.) DCD is defined as the variation of the pulse width in the data pattern. This is most effectively illustrated using a 101010 pattern. The pulse width for 1 may be different than the width for 0. DCD is typically introduced by the mismatch of even and odd delays in the transmitter. For single-ended signaling, DCD can also be due to the mismatch of rising and falling edges, and if the decision threshold is higher or lower than it should be. Furthermore, DCD is amplified by a lossy channel, because the narrow pulse contains more high-frequency components, which can be attenuated by the channel.

BUJ refers to any unbounded jitter not correlated with the data stream. PJ is an example of BUJ, which represents jitter that repeats in a certain time period that is independent of the data pattern. PJ, in general, consists of multiple frequency contents (SJs). Power supply modulation is often the root cause of PJ in the system. Many other examples of BUJ exist, including any crosstalk that is not correlated with the data pattern, phase interpolation errors, and external noise-induced jitter due to EMI.

On the other hand, random jitter (RJ) is unbounded and unpredictable. RJ is often assumed to be a Gaussian distribution, because the primary cause of RJ is the thermal noise in the transistors. More importantly, a Gaussian distribution, based on the central limiting theorem, can approximate the composite effect of many uncorrelated random noise sources. Because Gaussian distribution is unbounded, the peak-to-peak value of RJ is theoretically infinite, so using peak-to-peak to quantify RJ is meaningless. Consequently, RJ is best represented by a standard deviation, or root mean square (rms) value. Depending on the BER, the RJ impact on the link's total jitter can be derived by multiplying the rms value by two times the Q factor, as described in Chapter 6.

2.4 Summary

This chapter briefly reviewed the fundamental components of I/O signaling: channel topology, driver options, termination choices, and clocking architectures. A successful channel design must consider a good balance of performance, power, and cost for target applications. In addition to these basic signaling blocks, various noise sources and mechanisms were presented, along with an introduction to jitter basics. A good signal integrity engineer needs to understand, and be able to quantify, the impact of these noise sources. A significant portion of this book is dedicated to modeling and simulating the impact of these noises.

References

1. W. Dally and J. Poulton, *Digital System Engineering*, Cambridge University Press, 1998.
2. B. C. Wadell, *Transmission Line Design Handbook*, Artech House, 1991.
3. A. E. Sanderson, "Effect of surface roughness on propagation of the TEM mode," *Advances in Microwave*, vol. 8, L. Young, ed., Academic Press, 1971, pp. 2–57.
4. A. V. Mezhiba and E. G. Friedman, *Power Distribution Networks in High Speed Integrated Circuits*, Kluwer Academic, 2004.
5. M. Swaminathan and A. E. Engin, *Power Integrity Modeling and Design for Semiconductors and Systems*, Prentice Hall, 2008.
6. I. Novak, Ed., *Power Distribution Network Design Methodologies*, IEC, 2008.
7. W. Hobbs, A. Muranyi, R. Rosenbaum and D. Telian, "IBIS: I/O buffer information specification, overview," http://www.vhdl.org, January 14, 1994.

8. S. B. Huq, “Effective signal integrity analysis using IBIS model,” presented at the IEC DesignCon, Santa Clara, CA, 2000.
9. C.C. Huang, K. S. Oh, and S. Rajan, “The interconnect design and analysis of Rambus memory channel,” *The Pacific Rim/International, Intersociety, Electronic Packaging Technical/Business Conference* (InterPack’01), Kauai, Hawaii, Jul. 8–13, 2001.
10. A. Varma, M. Steer, and P. Franzon, “SSN issues with IBIS models,” in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2004, pp. 87–90.
11. Z. Yang, S. Huq, V. Arumugham, and I. Park, “Enhancement of IBIS modeling capability in simultaneous switching noise (SSN) and other power integrity related simulations-proposal, implementation, and validation,” in *Proceedings of International Symposium on Electromagnetic Compatibility*, Aug. 8–12, 2005, vol. 2, pp. 672–677.
12. A. Varma, M. Steer, and P. Franzon, “Improving behavior IO buffer modeling based on IBIS,” *IEEE Transactions on Advanced Packaging*, vol. 31, no. 4, pp. 711–721, Nov. 2008.
13. M. P. Li, *Jitter, Noise, and Signal Integrity at High-Speed*, Prentice Hall, 2007.
14. Bell Communications Research, Inc (Bellcore), “Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria, TR-253-CORE,” issue 2, rev. no. 1, Dec. 1997.
15. Agilent App. Note 1432, “Jitter analysis techniques for high data rates” (2003).
16. H. Hatamkhani and C.-K. K. Yang, “Power analysis for high-speed I/O transmitters,” in *Proceedings of International Symposium on VLSI Circuits*, Jun. 17–19, 2004, pp. 142–145.

Part I

Channel Modeling and Design

This page intentionally left blank

CHAPTER 3

Channel Modeling and Design Methodology

Chuck Yuan, Ravi Kollipara, Dan Oh, and Hao Shi

The passive interconnect (or channel), along with the transmitter (Tx) and receiver (Rx), is one of the three primary components of an I/O system. The channel connects the Tx with an Rx. The signal from the Tx may be distorted by the passive channel due to channel loss, dispersion, reflection, and crosstalk. When the data rate is sufficiently low, we can ignore the impact of the passive channel. At multi-gigabit data rates, the channel becomes the limiting factor in achieving a target data rate. Furthermore, the impact of the channel is strongly dependent on the signaling method used. Specifically, for a given data rate, the channel has a greater impact on single-ended signaling than on differential signaling due to single-ended signaling's higher crosstalk and simultaneous switching noise. In modern designs, the impact of the passive channel cannot be neglected and must be evaluated.

Designing a high-speed channel presents many difficult challenges, especially when the data rate is in the multi-gigahertz range. First, one must develop a systematic methodology for channel simulation and modeling. The models must be accurate enough for the targeted data rate, and one must be able to simulate them in a system environment efficiently. Many iterations, involving hundreds (if not thousands) of simulations are required to optimize system yield in terms of channel topology, channel parameters and their variations, and signal-conditioning methods, if required. Second, one must adopt a signal integrity–driven physical design. Attention must be paid to crosstalk and impedance discontinuities in packages, PCBs, and connectors. One must carefully design the power distribution network, in order to minimize the impact of supply noise on system performance. Finally, one must rigorously correlate the channel model with hardware and verify the system performance over process, voltage, and temperature (PVT). Figure 3.1 illustrates these important aspects of channel design and lists the detailed steps involved.

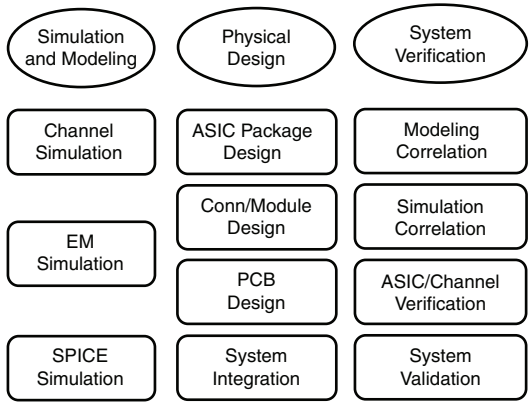


Figure 3.1 Passive Channel-Design Components

3.1 Channel Design Methodology

To demonstrate how to successfully manage the design challenges that high-speed channel design presents, this chapter describes the proven design process and methodology used for the Rambus® Direct RDRAM® and XDR™ channels [1–4]. The channel design and modeling methodology is, in essence, the process of balancing device and system characteristics with architectural constraints. This process enables the creation of component and channel specifications that guarantee the system operation, at the intended frequency, over all processing ranges and operating conditions. It is important to realize that not all SI engineers have the opportunity to define an I/O interface from the ground up. The typical case is that a standard-setting body defines the signaling methods and specifications. Sometimes, that body only defines a part of the system, as is often the case for memory interfaces, where only memory specifications are explicitly provided. Understanding how the standards are derived is very useful in performing trade-off analysis. The following discussion assumes that one has an option to define and design an I/O interface.

The Rambus channel design methodology considers system-level effects as key elements of the overall device and system requirements and specifications. We apply a top-down methodology, whereby the system-level specifications drive the component-level specifications, as illustrated in Figure 3.2. SI engineers work with architectural engineers to define the system requirements in the first step. Next, we perform the first-order analysis, where we create channel models, based on the extrapolated historical design and data. In the third step, we attempt to identify critical channel parameters, based on the proposed signaling topology and methods. We analyze these critical parameters to evaluate their impact on system performance and the design goals. After we are satisfied with the design, we proceed to the fourth step, which is to create a more detailed channel (signal integrity) model and power integrity model. We then use these models to define the initial specifications for various channel components, including the Tx and Rx parameters. These specifications define the I/O silicon/circuit design.

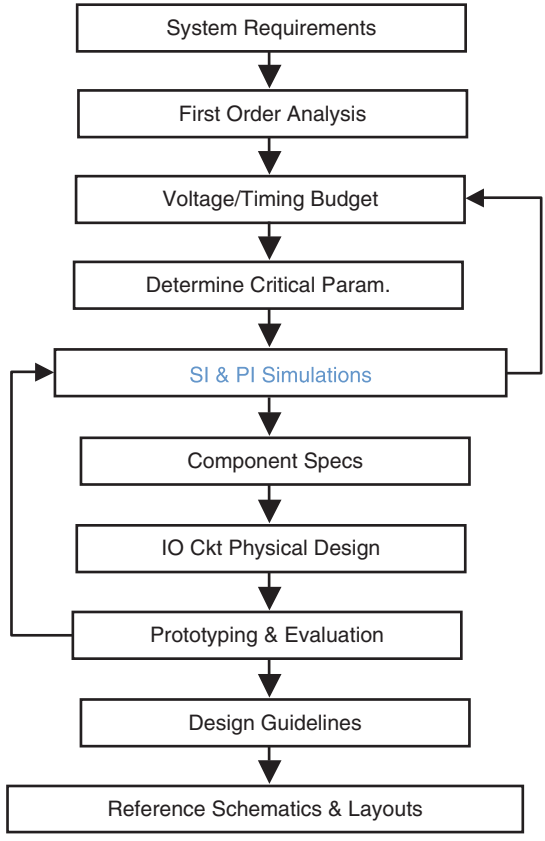


Figure 3.2 Top-Down Channel Design Methodology

The key design goal is to ensure robust system operation under the permitted worst-case process variations. This goal is achieved using balanced specifications for all the system components, and without overly constraining any particular component (which could result in significant or unnecessary yield loss). The process outlined in Figure 3.2 answers many of the following questions:

- What signaling scheme should be used, single-ended or differential signaling?
- What is the trade-off between point-to-point and bussed topology?
- What are the maximum data rates that need to be supported?
- What is the limiting factor of the design? Is it passive channel or device?
- What are the characteristics of Tx and Rx?
- What are the adequate Tx voltage swing and Rx voltage sensitivities?

- What are the acceptable Tx output timing (tQ) and Rx input timing (tS/tH)?
- Is active signal conditioning (Tx or Rx equalization) needed?
- Do we need per-bit timing calibration? Should it be one-time or periodic?
- Is on die termination (ODT) required? What are the ODT value and its tolerance?
- Do we need to calibrate ODT and driver rise time?
- What type of package do we use (wirebond, C4, or CSP)?
- What is the maximum trace length?
- What are the impedance tolerances of the packages and PCBs?
- What is the skew tolerance budget for the packages and PCB traces?
- What type of transmission lines do we use (microstrip or stripline)?
- What type of connectors do we use?

The preceding questions range from architectural to physical implementation details. Ultimately, one needs to balance the bandwidth, power, capacity, silicon area, and cost requirements at the system level. The answers to these questions depend on the targeted application. For the consumer market, cost is a major concern, whereas for the server market, performance (bandwidth and power) is the more important consideration. Figure 3.2 illustrates a systematic methodology that makes addressing these design trade-offs easier. The following sections use the Rambus XDR memory interface design to demonstrate many of the points described previously.

Specifically, why should one use point-to-point differential signaling when many other choices are available (as listed in Figure 2.1 in Chapter 2, "High-Speed Signaling Basics")? For many years, a popular method of connecting digital systems has been multi-point connections. The Direct Rambus DRAM (RDRAM) memory channel [2] is a good example. The Direct RDRAM memory channel started at 800Mb/s/pin and scaled to 1.6Gb/s/pin in later designs. However, the further performance improvement of multi-point single-ended topologies is very difficult due to a number of factors, including stub lengths, effective stub impedances, and device capacitive loading, as well as driver and termination implementations. Consequently, XDR data path (DQ) chooses point-to-point topology to achieve 3.2Gb/ps to 6.4Gb/s data rate. In contrast, the command/address (RQ) path still uses a bussed topology, similar to RDRAM signals, and operating at 800Mb/s. This bussed topology makes supporting larger memory capacity possible. As a result, there are two different channels to analyze, one for DQ and the other for RQ.

During the initial exploration phase in the design of the Rambus XDR memory system, the following system requirements were proposed:

- Supported data rates: 3.2Gb/s to 6.4Gb/s
- Controller package: C4 (Flip-chip) or wirebond package

- DRAM package: Chip scale package (CSP)
- Short channel: Maximum PCB trace of 5 inches without connectors
- Long channel: Maximum PCB trace of 8 inches with up to three connector crossings
- No package and PCB trace matching or skew requirements
- Controller package layer count: 2-2-2 or 3-2-3 build up package for DQ traces
- PCB layer count: Four layers with two signal-routing layers
- Microstrip lines are used for differential DQ signals
- Strip lines are used for single ended RQ signals

After building channel models based on past designs, the simulations are performed to determine the channel topology and signaling method. In the end, a differential point-to-point signaling topology (shown in Figure 3.3) is chosen to ensure data rate scalability. The key design features of this topology are:

- On-chip termination to minimize reflections
- Low voltage swing (200mV) to minimize power consumption
- Bi-directional data transfer to maximize channel efficiency

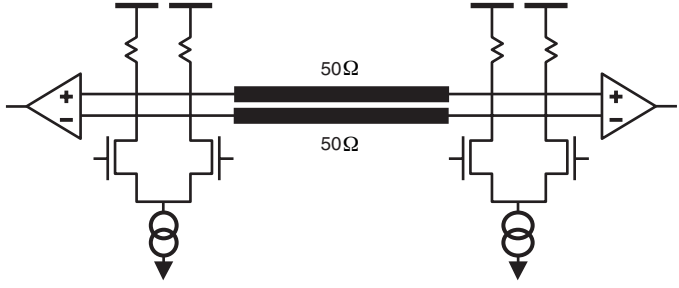


Figure 3.3 XDR Signaling Technology

The next step is to identify the critical channel parameters. These include the die parasitic capacitance, ODT impedance tolerance, channel impedance tolerance, package via impedance, package-plating stub for wirebond package, dielectric loss (as a function of temperature and humidity), as well as PCB trace length. For example, a clear trade-off exists between Tx voltage swing and channel attenuation introduced by die parasitic capacitance and PCB trace loss. One could also consider introducing signal conditioning into the I/Os, such as Tx finite impulse response (FIR), or Rx linear equalization. In this case, a design trade-off exists between silicon complexity and power/area impact versus system performance enhancement. One must have an accurate channel model in order to quantify the system performance gain due to various channel parameters and design options. After extensive simulations, it is clear that signal conditioning

provides marginal performance improvement at 3.2Gb/s, in particular, for a short memory channel (< 5 inches). For example, Figure 3.4 shows the channel parameter variations for a short XDR channel.

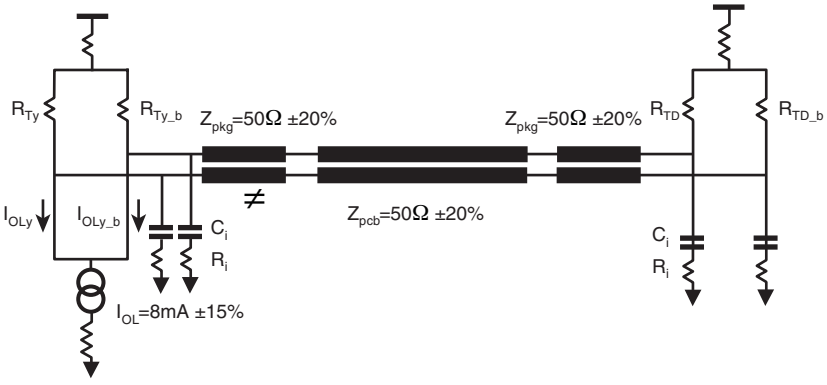


Figure 3.4 Channel Parameter Variations for a Short XDR Channel

Because XDR is targeted for consumer products, such as game consoles or DTVs, conventional packaging and PCB technologies are used. This, in turn, imposes low-cost manufacturing constraints on the design. The design challenges and goals include:

- Use of conventional package and PCB design rules
- Use of 100Ω differential impedance PCB traces in a microstrip line or stripline configuration
- Support for four-layer conventional PCBs with +/−15% impedance variation
- Support for wirebond or flip-chip BGA packages, with +/−20% impedance variation
- Generation of accurate broadband interconnect models that capture high-frequency effects
- Robust system operation under worst-case process variations
- 100% system and component yield at 3.2Gb/s

The last goal of achieving 100% yield is very important in the game console market, because the system performance does not vary. This means that every part produced can be used in the system. Similarly, every system built should be robust and suitable for shipment to customers. In contrast, one can tolerate a small yield initially, if the market permits different performance levels. This is typically true in the graphics market, where high-performance products are first introduced in high-end products, and later cost reduced for use in lower-end products. To achieve 100% yield, one must apply a statistical simulation technique, as described in Chapter 7, "Manufacturing Variation Modeling," to consider the impact of process variations. In addition,

one must use a reliable voltage and timing budget methodology to ensure robust system operation under the worst-case operating conditions (as described in Chapter 6, "Channel Voltage and Timing Budget"). The voltage and timing budget, derived from the channel simulations, drive the specifications for I/O silicon designs.

To build confidence in the proposed system design, we build prototype silicon and systems to validate our assumptions and models. We then apply the verification results from the prototype system to the silicon and PCB/package design to create the next-generation design. Therefore, the channel design methodology, shown in Figure 3.2, is an iterative process. With each iteration of the design, we gain more confidence in the signaling technology, and eventually drive the design into production. We use the channel simulation results to derive/optimize the physical design rules and guidelines for the silicon, package, and PCB.

In the previous discussion, the assumption is that one has an option to define an I/O interface from scratch, which is obviously not very common. In most cases, the signaling architecture is pre-defined by a standard-setting body, which performs an analysis similar to the one described. The SI engineer's job is to verify that his or her particular design meets the specification. In the case of memory systems, specifications are explicitly given only for DRAM, not for the controller. The SI engineer must back-calculate the specification for the passive channel. You can use the aforementioned methodology to facilitate this process. The difference is that the design space is much smaller, and the number of variables is restricted to package and PCB design parameters; the silicon related parameters are fixed. If, however, one ignores the potential impact of the package and PCB (as is done in many practical situations), the risk of having an unstable system increases dramatically as data rates increase. The signal integrity driven design outlined in Figure 3.2 minimizes such a risk and should be adopted.

3.2 Channel Modeling Methodology

As previously discussed, obtaining accurate channel models that one can use to predict channel behavior reliably is essential. Accurate channel modeling is particularly important when operating at multi-gigabits per second data rates, where the channel becomes the bottleneck. Channel non-idealities, such as reflections and attenuation at high frequency, become much more dominant. Ultimately, the simulation results offer very little value, if one does not trust the underlying model. One cannot reliably make design trade-offs without having confidence in the accuracy of the model. In this chapter, we specifically refer to the channel model as a passive channel model for traces, vias, wirebonds, and connectors.

Figure 3.5 shows a commonly used modeling methodology for designing high-speed memory interfaces [2–4]. This methodology is based on the fact that the channel is made of many subcomponents. You create an equivalent circuit model for each subcomponent, using an electromagnetic solver. The electromagnetic solver could be 2D, 2.5D, or 3D. It can also be either a quasi-static solver or a full-wave solver. You generate the channel model by combining the equivalent circuit models for all the subcomponents. Field solver simulation is an important area in channel modeling and is discussed in Section 3.3.

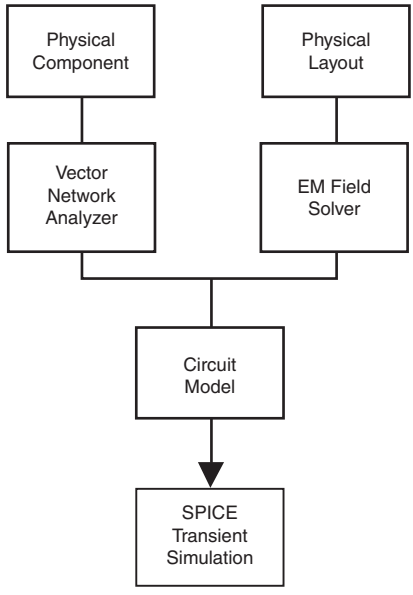


Figure 3.5 Example of a Common Channel-Modeling Methodology

The process of modeling a channel subcomponent starts with the physical design database, using electromagnetic field solvers. The first decision is whether to use a quasi-static or full-wave field solver. The choice depends on the electrical size of the structure. The quasi-static solvers produce the equivalent circuit model in the form of inductances, capacitances, and resistances. They are good for electrically small structures, such as C4 bumps or BGA balls. On the other hand, the full-wave solvers generate scattering (or S) parameters. S-parameters can be used directly in modern circuit simulators. Chapter 4, "Network Parameters," covers using S-parameters in a circuit simulator. However, the S-parameter is a "black box" model that offers very little physical insight into how to improve the design. Converting the S-parameter to an equivalent physical circuit model offers the opportunity to study design sensitivity to certain parameters. It is important to note that the key phrase here is equivalent physical circuit model. Because an equivalent physical circuit model is only an approximation, there could be many equivalent models for a given S-parameter. Because an equivalent physical model relates the design to common circuit parameters, such as capacitance, inductance, and impedance, it is more convenient to work with than a raw S-parameter model.

Section 12.2.2, in Chapter 12, "SSN Modeling and Simulation," describes a simple equivalent circuit-generation method for electrically short interconnect structures, such as vias, wirebonds, and connectors. For more complex structures (such as a complete package model), one can generate equivalent circuit models by using both lumped and distributed circuit elements, in conjunction with a general numerical optimization [5]. We can compare the accuracy of various equivalent circuit models—in this example, using coupled vias (Figure 3.6 shows the geometry).

We use both full-wave and quasi-static solvers to generate frequency dependent S-parameters, and static models using RLGC lumped elements, respectively. Then, an additional model is generated using a one-segment PI model, by curve fitting the full-wave model. Figure 3.7 shows the resulting S-parameter simulation data. Note that the figure plots S11, S21, S31, and S41, which represent the reflection, near-end crosstalk, insertion loss, and far-end crosstalk, respectively. In general, the RLGC model generated from the S-parameter fits well with the full-wave model at lower frequency range, as expected. In Figure 3.7, the following RLGC values are used to fit data: $C_1 = C_3 = 0.289\text{pF}$, $C_2 = C_4 = 0.28\text{pF}$, $C_{12} = C_{34} = 5.14\text{fF}$, $L_{13} = L_{24} = 950\text{pH}$, and $K = 0.29$. Section 3.3 discusses the accuracy of field solver modeling.

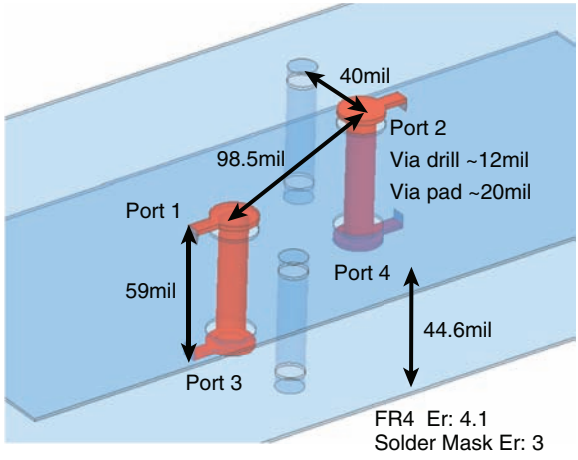


Figure 3.6 Geometry of Two Single-Ended Vias

Generating a model from a physical design involves simplifying the physical design, because the physical design contains many details, which may or may not influence the accuracy of the model. Including everything in the design often slows down the EM solver dramatically. Including all the physical details is neither practical nor necessary. Deciding what to include and what to simplify is an important decision. In addition, different EM solvers may offer different levels of accuracy. As a result, verifying a model based on prototype hardware is an essential part of ensuring the integrity of the model.

The model needs to be verified in both the time and frequency domains. There have been many debates over the pros and cons of using *time-domain reflectometer* (TDR) versus *vector network analyzer* (VNA) to accomplish this verification. The truth is that both methods are useful and needed to derive a complete picture of the passive channel. The time domain measurement using (TDR) offers a direct measurement of impedance discontinuities and crosstalk. One can easily identify the type, location, and the amount of discontinuities in the physical design. VNA measures the high-frequency behavior accurately, whereas TDR provides direct measurements

of the channel impedance profile and of the amount of crosstalk (far end or near end) in the channel. Section 4.2.1 discusses a few key issues in comparing these time- and frequency-domain methods.

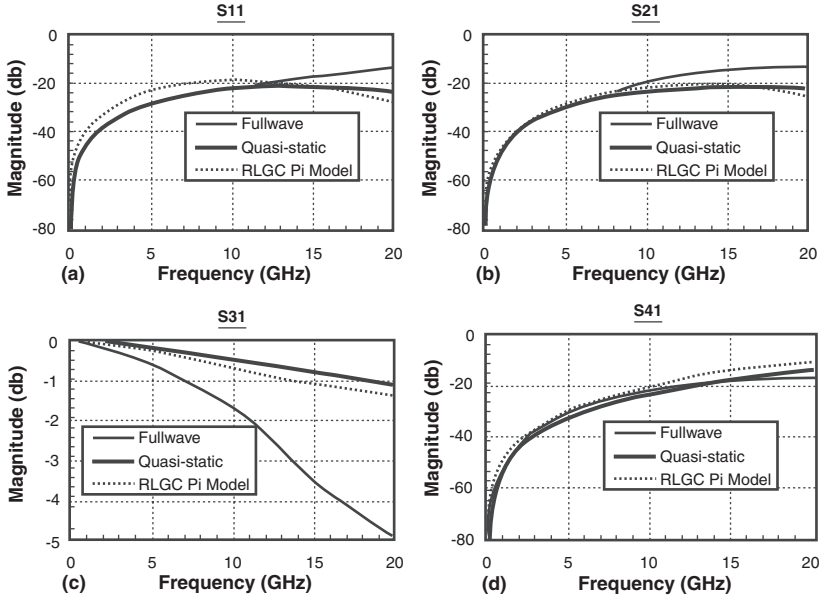


Figure 3.7 Comparison of Full-Wave, Quasi-Static, and RLGC Pi Models of S-Parameters

3.3 Modeling with Electromagnetic Field Solvers

We can generate an equivalent circuit model, using electromagnetic solvers, for a given physical design, or the layout of a package and PCB. We then include these models in system-level transient simulations using SPICE, which may include the non-linear behaviors of drivers and receivers. At a very high level, this sounds like a straightforward procedure. However, one must consider a number of factors in order to generate accurate models. First, one must determine the desired level of accuracy and the frequency of interest. The answer to these questions determines which EM field solver to use, and the complexity of the equivalent circuit model. Second, how will one use this model in the analysis? The answer to this question determines whether the model is purely mathematical and does not correspond to the physical design. Finally, one must trade off the computational costs versus accuracy. The answer to this question determines what is included in the physical model for the EM solvers, and how much computational resource is required. A physical design database usually contains many physical details that may have negligible influence on the electrical behavior of the design. Determining what to include in EM simulations involves experience and judgment. This, in turn, creates the need to correlate the model with lab measurements.

As mentioned previously, the two types of electromagnetic field solvers are *quasi-static* and *full-wave*. Quasi-static field solvers solve the Poisson or Laplace equation. The corresponding extracted physical parameters are inductances and capacitances, or transmission-line parameters. The equivalent circuit models consist of inductances and capacitances. The quasi-static solver is appropriate for interconnect structures that are much smaller than the wavelength of the highest frequency of interest. Typically, you can apply the 1/10 wavelength rule; that is, the largest dimension of the structure is smaller than 1/10 of the wavelength of the highest frequency. Figure 3.8 illustrates this, using a lossy transmission line with different lumped element models. The transmission line is 1 inch, with $R = 3.332\Omega/\text{m}$, $L = 500\text{nH}/\text{m}$, $G = 0$, and $C = 200\text{pF}/\text{m}$, which has about a 254-ps flight time. Applying the 1/10 wavelength rule, a one-segment model is accurate up to 400MHz in this case (as shown in Figure 3.8). However, this simple rule does not work for high-order lumped models, as demonstrated by the 5- and 10-segment cases. In general, a significant number of segments are required to model high-frequency distributed effects.

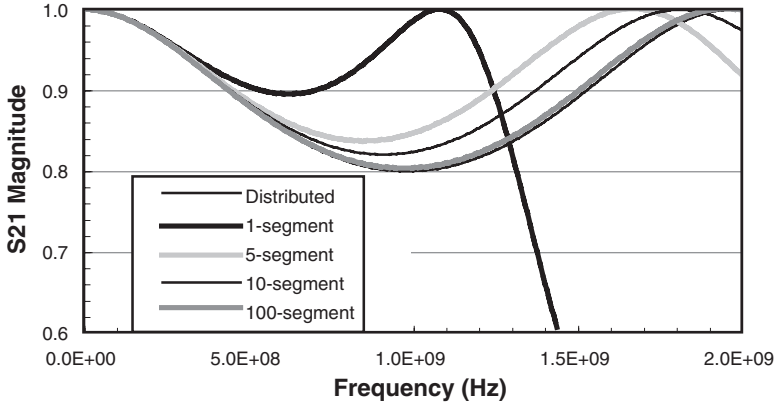


Figure 3.8 Lumped vs. Distributed Transmission-Line Models

Full-wave solvers solve the vector wave equation of the Maxwell equations. The extracted corresponding physical parameters are scattering parameters (or S-parameters). Theoretically, there is no limit to what structure these solvers can handle. However, most full-wave field solvers have a low-frequency limit, with the accuracy of the solver degrading at lower frequencies. At higher frequencies, more unknowns are required to obtain accurate solutions, and the computing resources limit the size of the problem. We can derive analytical formulae, through conformal mapping and/or eigenfunction expansion, but for simple geometries, we can often use numerical techniques, such as Finite Element Method (FEM), Finite Difference Time-domain Method (FDTD), or the Method of Moments (MoM). With increased capabilities, and the ease-of-use of

commercial EDA tools (such as ANSYS® Q3D Extrtactor® [6] and HFSS™ [7]), the majority of models are now generated through these field solvers.

In addition to different types of EM solvers, we can choose to model in either two-dimensions (2D), two-and-a-half dimensions (2.5D), or three dimensions (3D). A 2D solver is appropriate for package and PCB traces, whereas a 3D solver is often the right choice for wirebond and vias. we can use 2.5D solvers to model package/PCB traces and vias, but not wirebonds.

Finally, we can choose to model the entire interconnect structure, or break it into smaller pieces. The advantages of the first approach include convenience and accuracy, as it accounts for the interactions of the different interconnect segments. As the data rate increases, full-wave solvers are often required to simulate the entire interconnect structure. The disadvantage of the whole model approach is the large computational cost due to the large structures. In addition, we cannot easily evaluate the impact of individual segments, which is necessary for design optimization. In contrast, breaking the interconnect structures into smaller pieces makes them easier for the EM solver to handle, and enables optimization of each segment of the interconnect structure.

3.4 Backplane Channel Modeling Example

This section demonstrates a typical modeling process, using a backplane system as an example. Backplanes are used in networking, telecom, and server applications to route signals. The example backplane link is a high-speed differential, point-to-point serial link between a chip on a line card, and a chip on a switch card, connected through a backplane with ten independent components, as shown in Figure 3.9. The links are full duplex links, and have various trace lengths and via stub-lengths on the line, switch and backplane PCB modules, and chip packages. The links also go through various connector pair combinations, which result in various impedance and crosstalk profiles. Because low bit-error rates must be ensured on all the links in the system, the task is a challenging one [8–12]. Models are needed of all the components in order to predict the channel loss, impedance profile, crosstalk, and delay. These, in turn, predict the data rate that can be supported with a given set of Tx and Rx equalization techniques, and at a specified BER.

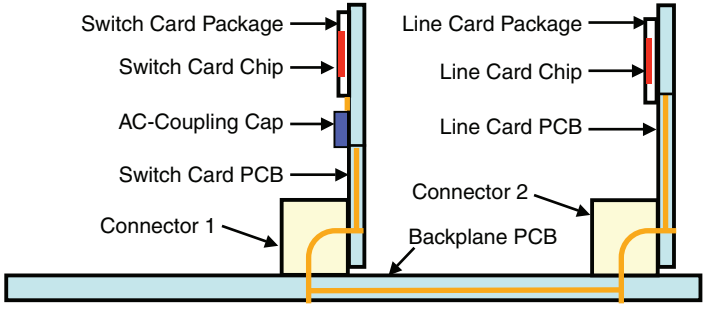


Figure 3.9 Example of a Backplane Channel with Various Components

The Tx and Rx parasitics may be considered part of the passive channel. The model, at a minimum, consists of the Tx or Rx on-die termination, and the parasitic capacitive loading of the Tx or Rx. This capacitive loading results from the ESD structure, die pad, on-chip interconnect, and driver or receiver circuit. We extract these parasitics using on-chip extraction tools, or by probing the pad with a simulated VNA, and then curve fitting the resulting S11 to a simple parallel RC circuit (or a more complex model). Alternatively, the die pads can be directly probed using a high-speed microwave probe and a VNA.

The most commonly used packages in backplane systems are flip-chip packages. We can use the 3-D field solver to model the package, from the flip-chip bumps to the BGA ball pads of the PCB. We can also characterize the package using a VNA and microwave probes, which land on the flip-chip package pads on one side, and close to the BGA ball pads (using very short transmission lines) on the PCB side. Probing a packaged chip is much easier than a bare die, because the chip can be powered and clocked. Consequently, the Tx or Rx die parasitics are extracted by probing from outside the package near the BGA balls, using short transmission lines, and de-embedding the package S-parameters from the measured S-parameters.

We model the line card, switch card, and backplane PCB traces using 2-D tools. However, we use a 3-D field solver to model the via pin fields of the through-hole backplane connector, the signal vias under the line card and switch card chip packages, and the vias near the AC-coupling cap. The backplane connector has three to six columns of differential pairs, with each pair having a different delay. Typically, the connector is modeled using a 3-D field solver. Another alternative is to extract the S-parameters of the connector using connector-characterization test boards and a VNA. Special test fixtures with TRL (Through, Reflection, and Length) calibration structures are used for this purpose. Another approach is to use simple connector test boards, probe the backside of the vias (on the card and the backplane sides) with differential probes, and then de-embed the card and backplane via models from the measured S-parameters in order to extract the connector model. In this case, the card and the backplane test board signal vias are not connected to any signal layers on the test boards.

3.4.1 Data-Rate Limitations from Tx and Rx Blocks

Link data rates are limited, not only by the purely passive components, but also by the active components. The main limitation comes from the low-pass filtering effect of the Tx or Rx parasitic capacitive loading. Figure 3.10 illustrates this effect for a few parasitic capacitance values of either Tx or Rx. Even if the channel between Tx and Rx is perfect, or completely absent, the large parasitic capacitance values still seriously limit the link's data rates. Consequently, minimizing the capacitive loading of Tx and Rx as the data rates increase is critical. Other limitations to the data rate, introduced by Tx and Rx, include duty cycle error, intra-pair skew, power supply noise–induced jitter (PSIJ), and limited receiver sensitivity (due to voltage noise sources and receiver input offset). Correcting some of these limitations, introduced by the channel, is possible by using Tx and Rx blocks with active equalization and crosstalk cancellation. However, these features increase the link's power consumption. In some cases, one can even minimize the impact

of the Tx and Rx lumped parasitic capacitive loading, by using on-chip inductive structures, and by distributing the capacitance across the inductors to make the LC loading look more like a transmission line loading of 100 Ω differential. The downside of this technique is that inductors take up space on the chip interconnect layers and introduce resistive losses.

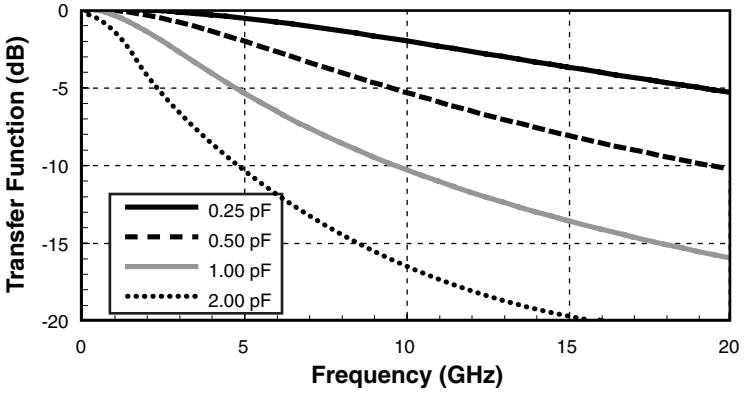


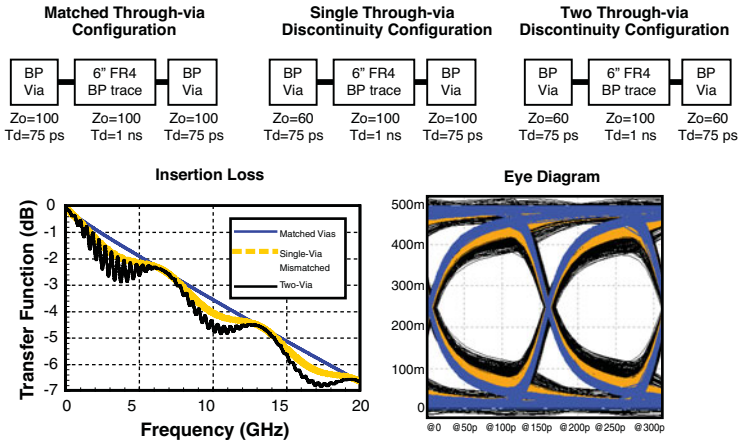
Figure 3.10 Impact of Tx/Rx Capacitive Loading

3.4.2 Impact of Through Via Reflections at the Connector Pin Fields

Backplane connectors have pins that protrude beyond the seating plane of the connector. The part of the backplane where the connectors are installed is called the *connector footprint area*. It contains plated through-hole vias with diameters large enough to accommodate the protruding connector pins. The connector is inserted into these via holes using a press, and both the electrical and mechanical connections to the backplane are made by press fit. This eliminates the soldering process from the backplane assembly, and facilitates easy field repairs, because we can remove the connector with special tools.

One drawback to press-fitted backplane connectors is that the protruding pin size requires large-diameter backplane vias. With some older generation connector footprints, this lowered the backplane differential through-via impedance to 50−60Ω. As a result, though the actual physical length of these vias is less than a centimeter, they had a significant impact on the link data rates. To illustrate this effect, we simulated the insertion loss of a 6-inch FR4 backplane trace with two impedance-matched vias, one impedance-matched via and one impedance-mismatched via, and two impedance-mismatched vias (see Figure 3.11). Figure 3.11 shows the corresponding eye diagrams for the three cases at a data rate of 6.4Gb/s. In this particular case, the loss at the Nyquist frequency doubled for a 6-inch FR4 backplane trace with a 60-ohm differential impedance press-fit connector via on each end, compared to the matched differential impedance vias on each end, and the eye opening is significantly reduced. Even worse, the insertion loss is non-monotonic, due to the resonance introduced by the large impedance discontinuities at the two locations. We

can see from the eye diagram that this contributes to reduced eye timing margin. The situation, presented by the lumped parasitic capacitances of Tx and Rx, is similar to that of the low-impedance backplane through-vias, in that the fast data edges see a low impedance termination, despite the matched resistive termination. Additionally, the insertion loss will demonstrate modulating behavior, similar to that in Figure 3.11, if the channel length is short.



The newer generations of backplane connectors have smaller diameter protruding pins, and so the diameter of the backplane connector pin field vias is smaller. This improves the differential impedance of these backplane vias to $80-95\,\Omega$. This is a significant improvement over the older generation connector footprint via impedance, though still not a perfect match to the channel impedance of 100Ω. If a better-matched channel is desirable, we could design the differential impedance of the channel to match the through-via impedance of the backplane connector vias. However, the connector impedance must also match this lowered channel impedance. An example of this approach is the recent PCIe Gen3 channel impedance specification, which is $85\,\Omega$ differential.

3.4.3 Impact of Via Stub Reflections at the Connector Pin Fields

For thick backplanes, vias attached to the stripline layers in the mid- to upper layers of the backplane will have significant stub lengths. We can model these via stubs as open-ended transmission lines. An open-ended transmission line presents a short, at the input of the transmission line, at frequencies when the stub length equals an odd multiple of the quarter-wave lengths. At these

frequencies, all the incident power is reflected back to the transmitter (Tx), and no power is delivered to the receiver (Rx). To illustrate this effect, we attached a 6-inch long FR4 backplane trace to one side of a single backplane via, as shown in Figure 3.12, then routed the trace on the top stripline layer (see Table 3.1 for details). Figure 3.12 shows the simulated transfer function for the backplane trace with the via on one end. The stub causes resonance at 5GHz, severely limiting the bandwidth of the channel, and the data rate it can support. For a 10Gb/s NRZ data rate, the Nyquist frequency is 5GHz. This means that the link's bandwidth is significantly less than 10Gb/s, even though the link can easily support this data rate with a trace routed on a bottom stripline layer. Unfortunately, all the links in a backplane system run at the same data rate, so the worst-case link determines the entire system's data rate. Figure 3.12 also illustrates the transfer function, but with the transmission line model of the via stub replaced by a lumped element capacitance, as is often done at much lower data rates. As the figure shows, it is completely ineffective in predicting the channel cutoff at 5GHz. Chip packages often have plating stubs in order to lower package cost. The effect of plating stubs on the channel transfer function is similar to that of via stubs, and may limit the data rate of the system.

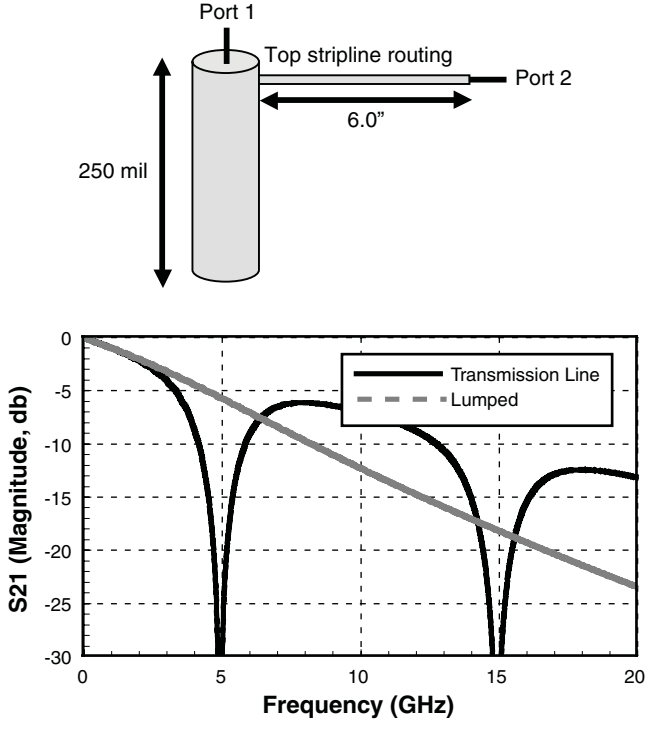


Figure 3.12 Backplane Via Attached to Trace Routed on Top Stripline Layer and Transfer Function with Via Modeled as Transmission Line and Lumped Capacitance

Table 3.1 Backplane Parameters

Parameter Type	Value
Backplane thickness	6.35 mm (250 mil)
Backplane trace length	6-inch long FR4 trace
Backplane via impedance and delay	50 Ω and 54.4 ps
Stub length for top stripline routing without back drilling	5.84 mm (230 mil)
Stub delay for top stripline routing	50.0 ps
Stub length for top stripline routing with back drilling	2.03 mm (80 mil)
Stub delay for top stripline routing with back drilling	17.4 ps

Back drilling is an effective way to remove the bottleneck presented by the via stubs. Back drilling has proven to be reliable and cost-effective [13], and can leave a stub as small as 5 mil. Another way to eliminate via stubs is to use surface-mount connectors in conjunction with blind vias. However, this option is not popular, as it adds significant cost to the system. One disadvantage of press-fit through-hole connector vias is that a stub length of up to 80 mils is still present, in order for the top-most stripline layer-routed vias to accommodate the ~100-mil long press-fit connector pin and provide a reliable contact. However, the resonant dip in the transfer function, caused by the 80-mil stub, occurs well beyond 10GHz, and does not present a problem below a 12.5-Gb/s data rate. Figure 3.13 shows a top stripline trace, attached to the two backplane vias, and the impact of back drilling on the transfer function. As shown in Figure 3.13, the resonance at 5GHz (without back drilling the via stub) is pushed to near 15GHz with back drilling, allowing the link to operate at a data rate of ~ 12.5Gb/s. The smooth transfer function of the channel (shown in Figure 3.12) is modulated by the secondary resonance (shown in Figure 3.13), which is caused by the reflections at the two vias. This resonance frequency is at multiples of the inverse of the roundtrip time of the backplane trace. The stub resonance frequency dominates the channel cut-off, rather than the secondary resonance frequency.

3.4.4 Impact of Crosstalk in Backplane Channels

Far-end crosstalk (FEXT) is the crosstalk of concern in memory channels. This is because memory channels are bi-directional links, and at any given time, the signal flow is only in one direction (that is, WRITE or READ). Typically, we route the memory signals on the top or bottom surface, as microstrip lines. FEXT is non-zero, due to the inhomogenity of the microstrip medium. In addition, the memory signals are single-ended, which also exacerbates the crosstalk. On the other hand, near-end crosstalk (NEXT) is the primary concern in backplane channels, because the links are full duplex. The chips are transmitting and receiving at the same time on different links. The transmitting and receiving links must be handled properly, so that NEXT (from

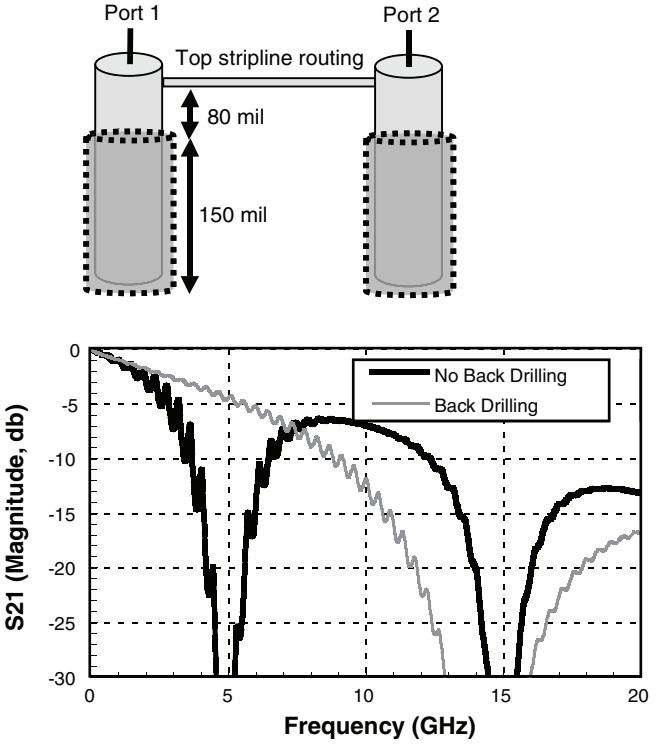


Figure 3.13 Back-Drilled Backplane Vias Attached to Trace Routed on Top Stripline Layer and Transfer Functions with and without Back Drilling

a full swing Tx) does not couple to a highly attenuated Rx signal, and degrade the signal to crosstalk ratio, resulting in a high BER. FEXT is of less concern with backplanes, line/switch cards, and package traces, as the differential pairs are routed in internal layers as edge-coupled striplines. FEXT is negligible for striplines, as long as the dielectric mediums, above and below the striplines, have nearly the same dielectric constant. However, connectors and vias still contribute to FEXT in the backplane links. NEXT is always present for both microstrip and striplines.

Figure 3.14 provides an example of connector-pin assignments and routing that result in greater crosstalk at the receiver. If a differential signal pair in the connector-pin field region is not fully surrounded by ground vias, then the connector-pin assignments result in Tx and Rx backplane via pairs facing each other, as shown in Figure 3.14(a). This results in higher crosstalk at the receiver, due to NEXT. Here, the Tx and Rx references are with respect to either the switch card chip, or the line card chip. The crosstalk at the receiver is also higher, if the Tx and Rx links are routed in the same stripline layer, due to NEXT being non-zero, as shown in Figure 3.14(b).

The via crosstalk is proportional to the via coupling length. Consequently, two through-via differential pairs, as shown in Figure 3.14(c), will have higher crosstalk. Figure 3.15 is an example of connector pin assignments and routing that result in lower crosstalk at the receiver. The Tx and Rx pairs are better isolated, as shown in Figure 3.15(a) and (b), resulting in lower NEXT at the receiver. If the via pairs alternate between through-vias and stub vias, or better yet, between through-vias and back drilled vias, the crosstalk is further reduced, as shown in Figure 3.15(c).

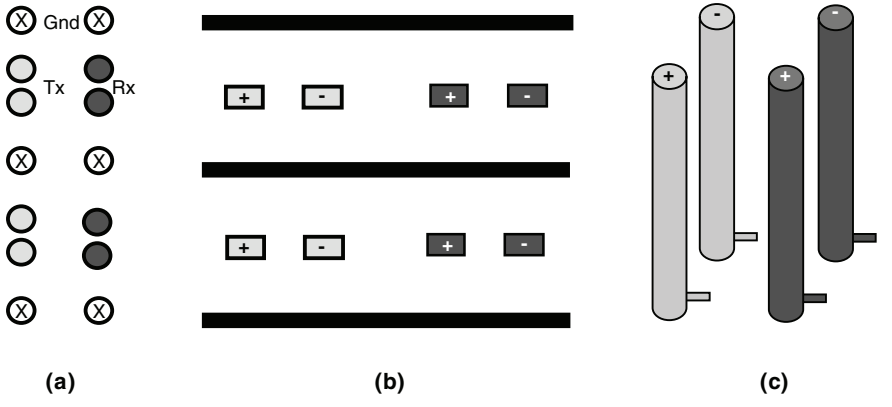


Figure 3.14 Poor Crosstalk Isolation: (a) Connector-Pin Field Assignment, (b) Stripline Routing, and (c) Two Through-Via Pairs Next to Each Other

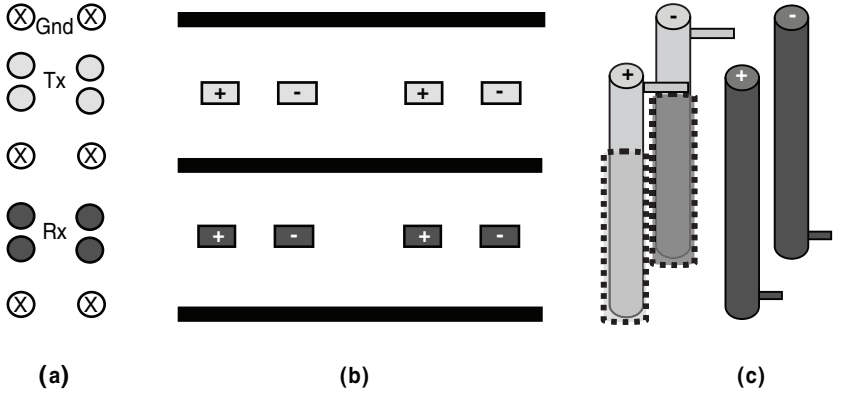


Figure 3.15 Good Crosstalk Isolation: (a) Connector-Pin Field Assignment, (b) Stripline Routing, and (c) Through-Via Pair Next to Back-Drilled Via Pair

3.4.5 Impact of Intra-Pair Skew

Controlling the intra-pair skew of a differential pair is important at high data rates. For example, a 1% mismatch in the single-ended delay of a 30" channel, with 5ns of transit time, translates to a skew of 5% UI at a data rate of 1.0Gb/s. The same amount of delay mismatch translates to a skew of 50% UI at 10Gb/s. The former case will have minimal impact on the eye diagram and the link BER. The latter case will have significant impact on the eye diagram, as the rise-time degrades, and results in a high BER (see Figure 3.16). Consequently, the absolute skew needs to scale in proportion to the UI, so that the intra-pair skew, as a percentage of the UI, remains the same. Sometimes, matching delays across all the components can be difficult; for example, in a high-density package. In such cases, one can compensate for the skew introduced in a package by introducing an intentional skew in the opposite direction, on the line or switch card. We must cancel the skew in delay, not the skew in length, as the propagation velocities could be different in the two components. Even length matching within a component may not guarantee zero skew, if the two traces have different numbers of bends. Again, it is the delay that we must match, not necessarily the length.

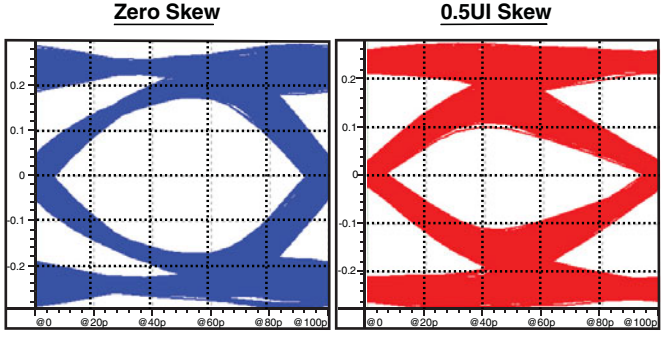


Figure 3.16 Receiver Eye Diagrams at 10Gb/s: Zero Skew and 0.5UI Skew

Another deleterious effect of skew at high data rates is the increased conversion of the differential signal to a common signal, as the rise times shorten. Typically, a differential impedance of 100 Ω is targeted across all the components; there is no equivalent common impedance. Consequently, unlike the differential signal, the generated common signal does not propagate smoothly across the components, and is reflected at the component interfaces. This contributes to jitter and radiation, which may cause EMI.

3.4.6 Impact of Manufacturing Variations

The backplane link designer must consider the impact of manufacturing, environmental, and voltage variations on system performance to ensure satisfactory system operation under the

required conditions. Typically, we design SerDes circuits to minimize the impact of process, voltage, and temperature variations on the performance of the transmitters and receivers. If the operating environmental conditions (for example, temperature and humidity) are not tightly controlled, the impact on the channel performance could be significant [11] [14]. Chapter 7 covers various methods of handling the channel manufacturing variations. This section presents a brief summary of an analysis, performed on a backplane channel, to account for manufacturing variations [10]. We analyzed four links of a dual-star 14-slot ATCA backplane. The first two links have backplane trace lengths of 10" that go through different connector row pairs. The second two links have backplane trace lengths of 1.1" that also go through different connector row pairs. All four links have maximum stub lengths of 75 mil on the line and switch cards and 100 mil on the backplane. Typically, the links with the longest stub lengths have the worst margins. We estimated both the nominal and worst-case voltage margins, at a data rate of 6.4 Gb/s and a BER of 1e-15 (see Table 3.2). We estimated the worst-case margin using the Monte-Carlo method, choosing nominal minus 3-sigma as the limit, and simulating for the absolute worst-case margin by aligning the worst-case loss of each component. In general, this is sufficient to ensure margin at the Monte-Carlo corner, because many components are in the system, and the probability of all the components aligning in a worst-case loss configuration is negligible.

Table 3.2 Voltage Margin at a BER of 1e-15 for Links Operating at 6.4Gb/s

Link	Nominal	Worst-Case Monte-Carlo	Simulated Worst Case
1	25 mV	15 mV	6 mV
2	16 mV	17 mV	6 mV
3	79 mV	66 mV	50 mV
4	86 mV	74 mV	53 mV

3.5 Summary

This chapter covered the passive channel design goals and challenges, along with methodologies for developing channel models. The key lessons of this chapter are:

- Channel design is critical to the robust operation of a high-speed system.
- A systematic methodology is required for successful channel design.
- Developing an accurate channel model, and verifying it with hardware, is crucial.

References

1. *Rambus Developer Forum*, San Jose, CA, October 22–23, 2001.
2. A. Moncayo, S. Hindi, C.-C. Huang, R. Kollipara, H.-J. Liaw, D. Nguyen, D. Perino, A. Sarfaraz, C. Yuan, M, Leddige, J. McCall, X. Moua, and J. Salmon, "Physical layer design of 1.6 GB/s DRAM bus," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 1999, pp. 11–14.
3. W. Beyene, C. Yuan, N. Cheng, and H. Wu, "Interconnect design of a 3.2 Gbps bi-directional memory system," presented at the IEC DesignCon, Santa Clara, CA, 2002.
4. X. Yuan, W. Beyene, N. Cheng, and H. Wu, "Design and modeling of a 3.2Gbps memory channel," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2002, pp. 227–230.
5. C.-C. Huang and J. Feng, "Optimizing VNA measurements by cascaded transmission lines for interconnect characterization," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2000, pp. 107–110.
6. *User's Guide*, Ansoft Maxwell Quick3D, 1999.
7. *User's Guide*, Ansoft, High Frequency Structure Simulator, 1999.
8. R. Kollipara, G. Yeh, B. Chia, and A. Agarwal, "Design, modeling and characterization of high speed backplane interconnects," presented at the IEC DesignCon, Santa Clara, CA, 2003.
9. R. Kollipara and B. Chia, "Modeling and verification of backplane press-fit PTH vias," presented at the IEC DesignCon, Santa Clara, CA, 2004.
10. R. Kollipara, B. Chia, Q. Lin, and J. Zerbe, "Impact of manufacturing parametric variations on backplane system performance," presented at the IEC DesignCon, Santa Clara, CA, 2005.
11. J. Zerbe, Q. Lin, V. Stojanovic, A. Ho, R. Kollipara, F. Lambrecht, and C. Werner, "Comparison of adaptive and non-adaptive equalization techniques in high performance backplanes over temperature, humidity, and impedance variations," presented at the IEC DesignCon, Santa Clara, CA, 2005.
12. R. Kollipara, B. Chia, F. Lambrecht, C. Yuan, J. Zerbe, G. Patel, T. Cohen, and B. Kirk, "Practical design considerations for 10 to 25 Gbps copper backplane serial links," presented at the IEC DesignCon, Santa Clara, CA, 2006.
13. T. Cohen, "Practical guidelines for the implementation of back drilling plated through hole vias in multi-Gigabit board applications," presented at the IEC DesignCon East, Santa Clara, CA, 2003.
14. G. Sheets and J. D'Ambrosia, "The impact of environmental conditions on channel performance," presented at the IEC DesignCon, Santa Clara, CA, 2004.

CHAPTER 4

Network Parameters

Dan Oh

Typically, the passive components of an I/O channel consist of various lumped elements and transmission lines, which introduce significant signal dispersion, reflection, attenuation, and crosstalk. One of the main tasks of signal integrity engineers is to provide accurate passive-channel models for high-speed channel analysis. RF engineers used network parameters (such as Y-, Z-, ABCD-, T-, and S-parameters) for decades as a convenient way to model a complex passive channel. Network parameters are becoming increasingly popular with circuit and SI engineers. Most of the modern SPICE simulators now support network parameters (such as S-parameter) as a circuit model for both time-domain and frequency-domain simulations. With these modern SPICE simulators, network parameters can represent an entire passive channel. For instance, in high-speed link BER simulation (described in Chapter 8, "Link BER Modeling and Simulation"), the channel simulation does not require a detailed topology based on circuit models. A single network-parameter model for the entire link is sufficient for performance analysis.

Building accurate network models requires a thorough understanding of network parameters, including precise definitions and conversion formulae. In SI analysis, preparing network models for time-domain simulation poses additional challenges that are not concerns for RF engineers. This chapter reviews the fundamentals of network parameters, along with accurate methods with which to generate time-domain network models. In particular, the chapter provides various formulae for converting between different network parameters, along with clear definitions and assumptions. It also describes the conversion between mixed-mode and single-ended parameters. The formulae derived are for general multi-conductor cases, so there are no restrictions on applications. The chapter also reviews the fundamentals behind causality and passivity checks, and presents a few practical tips to help ensure the accuracy of network parameter models.

4.1 Generalized Network Parameters for Multi-Conductor Systems

4.1.1 N-Port versus General 2-Port Network Parameter Representations

Before we delve into the details of network parameters, let us first define a general representation for network parameters. Figure 4.1 illustrates two different network representations of a two-line transmission line. Figure 4.1(a) is a traditional *N*-port representation, where each terminal is defined as an independent port, resulting in four ports for four terminals. Figure 4.1(b) is a generalized 2-port representation, where the terminals are grouped into two ports.

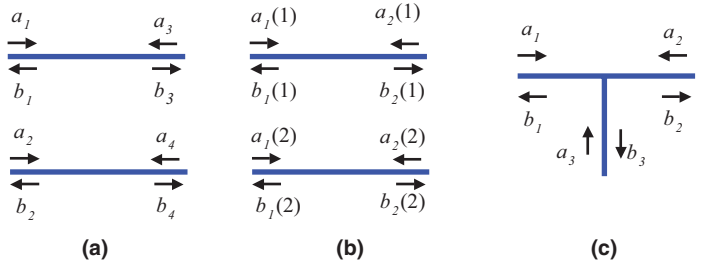


Figure 4.1 Representations of Coupled Transmission Lines: (a) *N*-Port, (b) Generalized 2-Port, and (c) *N*-Port representation for T Junction

The *N*-port representation is the most general form, because it can model arbitrary structures where input and output ports are not explicitly defined [1]. (The T-junction structure shown in Figure 4.1(c) is one such example.) The *N*-port representation also results in a very compact formula for converting between different network parameters. Most textbooks use this representation.

However, the *N*-port representation is not convenient for merging with, or post-processing to, other ABCD or transmission line parameters, because the input and output ports are not clearly defined. Moreover, the conversion process can be numerically unstable for certain cases, where no explicit connection exists between any two terminals. The two-line transmission in Figure 4.1(a) is an example of this stability issue. Near the DC point, the coupling between a_1 and b_4, or a_2 and b_3, becomes small. The formulae used to convert the S-parameter to other parameters (such as Y- or ABCD-parameters) can be unstable due to the inversion of the ill-conditioned matrix. On the other hand, the T-junction structure does not suffer from this problem, because all the terminals are physically connected.

For transmission-line type structures where the input and output ports are clearly defined, the generalized 2-port representation (consisting of *N*-terminal for each port) always produces numerically stable formulae. This chapter covers both definitions and describes the conversion

formulae for each. For other general derivations, (offered to prove a concept, or for the purpose of illustration), the *N*-port representation is used because of its generality and simplicity.

4.1.1.1 *N*-Port Network Parameters

The *N*-port network parameters for impedance, admittance, and S-parameters, are defined by

$$\vec{v} = \mathbf{Z}\vec{i} \tag{4.1a}$$

$$\vec{i} = \mathbf{Y}\vec{v} \tag{4.1b}$$

$$\vec{b} = \mathbf{S}^p\vec{a} \tag{4.1c}$$

where $\mathbf{Z} = \mathbf{Y}^{-1}$. *For the N-port representation, all voltages are measured with, or defined by, local references*. Therefore, in general, there could be *N* signal conductors with *N* reference conductors. The conversion between the S- and Z-parameters is

$$\mathbf{Z} = \mathbf{Z}_o^{1/2}(\mathbf{I} - \mathbf{S}^p)^{-1}(\mathbf{I} + \mathbf{S}^p)\mathbf{Z}_o^{1/2} \tag{4.2a}$$

$$\mathbf{S}^p = \mathbf{Z}_o^{-1/2}(\mathbf{Z} - \mathbf{Z}_o)(\mathbf{Z} + \mathbf{Z}_o)^{-1}\mathbf{Z}_o^{1/2}. \tag{4.2b}$$

The reference impedance conversion of the S-parameter is often useful. The conversion can be performed indirectly, using the preceding equations, by converting to Z and then converting back to S, using a different reference impedance.

To demonstrate the potential numerical instability of (4.2), we replace the two transmission lines in Figure 4.1(a) with two resistors. Assuming that both the resistor and reference impedance values are 50Ω, the resulting S-parameter is:

$$\begin{bmatrix} 1/3 & 0 & 2/3 & 0 \\ 0 & 1/3 & 0 & 2/3 \\ 2/3 & 0 & 1/3 & 0 \\ 0 & 2/3 & 0 & 1/3 \end{bmatrix}. \tag{4.3}$$

The matrix $(\mathbf{I} - \mathbf{S}^p)$ has an eigenvalue of zero, indicating that it is ill conditioned. In practice, multi-conductor transmission lines have significant coupling terms between neighboring lines, and these coupling terms help improve the matrix condition number. Nonetheless, one should avoid using this general *N*-port representation for uniform transmission lines, as a more stable equivalent 2-port representation is available.

4.1.1.2 Generalized 2-Port Network Parameters

Figure 4.2 shows a generalized 2-port network, for a multi-conductor system. The figure also includes the definitions of the port variables. Assuming that the reference characteristic impedance matrices ($\mathbf{Z}_{o,1}$ and $\mathbf{Z}_{o,2}$) for the wave variables are diagonal matrices, then Kurokawa [2] defines the power waves ($\vec{a}_j$ and $\vec{b}_j$) in the frequency domain, as

$$\vec{a}_j(i) \equiv \frac{1}{2\sqrt{\mathrm{Re}[\mathbf{Z}_{o,j}(i)]}} \left[\vec{v}_j(i) + \vec{i}_j(i)\mathbf{Z}_{o,j}(i) \right] \tag{4.4a}$$

$$\vec{b}_j(i) \equiv \frac{1}{2\sqrt{\mathrm{Re}[\mathbf{Z}_{o,j}(i)]}} \left[\vec{v}_j(i) - \vec{i}_j(i)\mathbf{Z}^*_{o,j}(i) \right] \tag{4.4b}$$

where * represents the Hermitian conjugate. This popular definition of power waves has been demonstrated to be incompatible with measurement data, and a new definition has been proposed [3]. However, when the reference characteristic impedances are real, both definitions are identical. We stay with the earlier definitions of power waves, because this real reference impedance assumption is quite reasonable in practice.

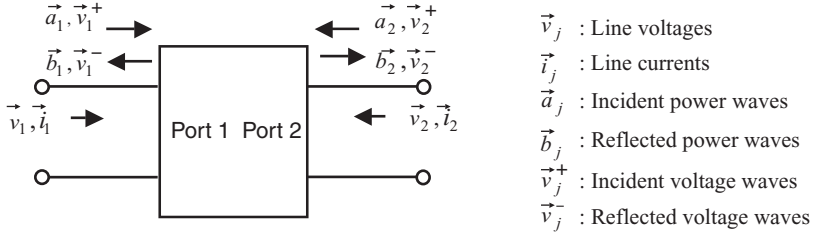


Figure 4.2 Generalized 2-Port Network Input and Output Definitions

Similar to the *N*-port representation, *N* terminals in each port can be associated with *N* distinct reference conductors. The number of the reference conductor can also be one, when there is a well-defined single return path for all the signals. In such cases, this return path is defined as a ground conductor, in field solver modeling.

The conversion formulae for the network parameters, based on the general 2-port representation, are quite complicated; the following four sections describe them. The first of these sections covers the conversion formulae for network parameters, based on line voltage and current, such as Z-, Y-, and ABCD-parameters. The next section deals with network parameters based on

voltage and current waves, such as S- and T-parameters. The third section describes the conversion between line and wave parameters. Finally, the fourth section examines the conversion between a normal (single-ended) mode and a mixed mode, which is applicable to both the *N*-port and generalized 2-port representations.

4.1.2 Line Voltage and Current Network Parameters: Z, Y, and ABCD

The line voltage and current represent measurable quantities at transmission lines, and are the sum of both forward and backward waves. They are also the conventional voltage and current used in circuit analysis. This section examines the conversion process for the network parameters associated with line parameters. The line voltage vectors ($\vec{v}_1$ and $\vec{v}_2$) and line current vectors ($\vec{i}_1$ and $\vec{i}_2$) on a multi-conductor system are related by the impedance matrix $\mathbf{Z}$, the admittance matrix $\mathbf{Y}$, and the **ABCD** matrix, as follows:

$$\begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \end{bmatrix} = \mathbf{Z}\begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{bmatrix}\begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \end{bmatrix} \tag{4.5a}$$

$$\begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \end{bmatrix} = \mathbf{Y}\begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{bmatrix}\begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \end{bmatrix} \tag{4.5b}$$

$$\begin{bmatrix} \vec{v}_1 \\ \vec{i}_1 \end{bmatrix} = \mathbf{ABCD}\begin{bmatrix} \vec{v}_2 \\ -\vec{i}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}\begin{bmatrix} \vec{v}_2 \\ -\vec{i}_2 \end{bmatrix} \tag{4.5c}$$

where a minus sign is inserted in the second entry of the input vector for **ABCD**, in order to allow them to be cascaded. For instance, the **ABCD** matrix of two cascaded ABCD systems, from port 1 to port 3, is

$$\begin{bmatrix} \vec{v}_1 \\ \vec{i}_1 \end{bmatrix} = \mathbf{ABCD}_{(1,3)}\begin{bmatrix} \vec{v}_3 \\ -\vec{i}_3 \end{bmatrix} = \mathbf{ABCD}_{(1,2)} \cdot \mathbf{ABCD}_{(2,3)}\begin{bmatrix} \vec{v}_3 \\ -\vec{i}_3 \end{bmatrix}. \tag{4.6}$$

Due to the cascading nature of the ABCD-parameter, it is very useful for merging and dividing network models. For instance, one can use it to de-embed port discontinuities in modeling transmission lines based on measurements, as described in Section 5.4.2. Table 4.1 provides the conversions between the Z-, Y-, and ABCD-parameters.

Table 4.1 Conversion Formulae for **Z**, **Y**, and **ABCD** Matrix Parameters

	Z	**Y**	**ABCD**
Z	$\begin{bmatrix}\vec{v}_1\\ \vec{v}_2\end{bmatrix}=\begin{bmatrix}\mathbf{Z}_{11} & \mathbf{Z}_{12}\\ \mathbf{Z}_{21} & \mathbf{Z}_{22}\end{bmatrix}\begin{bmatrix}\vec{i}_1\\ \vec{i}_2\end{bmatrix}$	$\mathbf{Y}_{11}=(\mathbf{Z}_{11}-\mathbf{Z}_{12}\mathbf{Z}_{22}^{-1}\mathbf{Z}_{21})^{-1}$ $\mathbf{Y}_{22}=(\mathbf{Z}_{22}-\mathbf{Z}_{21}\mathbf{Z}_{11}^{-1}\mathbf{Z}_{12})^{-1}$ $\mathbf{Y}_{12}=-\mathbf{Z}_{11}^{-1}\mathbf{Z}_{12}\mathbf{Y}_{22}$ $\mathbf{Y}_{21}=-\mathbf{Z}_{22}^{-1}\mathbf{Z}_{21}\mathbf{Y}_{11}$	$\mathbf{A}=\mathbf{Z}_{11}\mathbf{Z}_{21}^{-1}$ $\mathbf{B}=\mathbf{Z}_{11}\mathbf{Z}_{21}^{-1}\mathbf{Z}_{22}-\mathbf{Z}_{12}$ $\mathbf{C}=\mathbf{Z}_{21}^{-1}$ $\mathbf{D}=\mathbf{Z}_{21}^{-1}\mathbf{Z}_{22}$
Y	$\mathbf{Z}_{11}=(\mathbf{Y}_{11}-\mathbf{Y}_{12}\mathbf{Y}_{22}^{-1}\mathbf{Y}_{21})^{-1}$ $\mathbf{Z}_{22}=(\mathbf{Y}_{22}-\mathbf{Y}_{21}\mathbf{Y}_{11}^{-1}\mathbf{Y}_{12})^{-1}$ $\mathbf{Z}_{12}=-\mathbf{Y}_{11}^{-1}\mathbf{Y}_{12}\mathbf{Z}_{22}$ $\mathbf{Z}_{21}=-\mathbf{Y}_{22}^{-1}\mathbf{Y}_{21}\mathbf{Z}_{11}$	$\begin{bmatrix}\vec{i}_1\\ \vec{i}_2\end{bmatrix}=\begin{bmatrix}\mathbf{Y}_{11} & \mathbf{Y}_{12}\\ \mathbf{Y}_{21} & \mathbf{Y}_{22}\end{bmatrix}\begin{bmatrix}\vec{v}_1\\ \vec{v}_2\end{bmatrix}$	$\mathbf{A}=-\mathbf{Y}_{21}^{-1}\mathbf{Y}_{22}$ $\mathbf{B}=-\mathbf{Y}_{21}^{-1}$ $\mathbf{C}=\mathbf{Y}_{12}-\mathbf{Y}_{11}\mathbf{Y}_{21}^{-1}\mathbf{Y}_{22}$ $\mathbf{D}=-\mathbf{Y}_{11}\mathbf{Y}_{21}^{-1}$
ABCD	$\mathbf{Z}_{11}=\mathbf{A}\mathbf{C}^{-1}$ $\mathbf{Z}_{12}=\mathbf{A}\mathbf{C}^{-1}\mathbf{D}-\mathbf{B}$ $\mathbf{Z}_{21}=\mathbf{C}^{-1}$ $\mathbf{Z}_{22}=\mathbf{C}^{-1}\mathbf{D}$	$\mathbf{Y}_{11}=\mathbf{D}\mathbf{B}^{-1}$ $\mathbf{Y}_{12}=\mathbf{C}-\mathbf{D}\mathbf{B}^{-1}\mathbf{A}$ $\mathbf{Y}_{21}=-\mathbf{B}^{-1}$ $\mathbf{Y}_{22}=\mathbf{B}^{-1}\mathbf{A}$	$\begin{bmatrix}\vec{v}_1\\ \vec{i}_1\end{bmatrix}=\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\vec{v}_2\\ -\vec{i}_2\end{bmatrix}$

4.1.3 Voltage and Current Wave Network Parameters: S and T

Typically, high-frequency measurements are performed using wave parameters, rather than line parameters. This section describes the conversion between two wave network parameters (S- and T-parameters). The incident and reflected voltage-wave vectors ($\vec{v}_j^{+}$ and $\vec{v}_j^{-}$), and the incident and reflected power-wave vectors ($\vec{a}_j$ and $\vec{b}_j$) are related by the voltage-mode S-parameter matrix $\mathbf{S}^v$, and the power-mode S-parameter matrix $\mathbf{S}^p$, as follows:

$$\begin{bmatrix}\vec{v}_1^{-}\\ \vec{v}_2^{-}\end{bmatrix}=\mathbf{S}^v\begin{bmatrix}\vec{v}_1^{+}\\ \vec{v}_2^{+}\end{bmatrix}=\begin{bmatrix}\mathbf{S}_{11}^v & \mathbf{S}_{12}^v\\ \mathbf{S}_{21}^v & \mathbf{S}_{22}^v\end{bmatrix}\begin{bmatrix}\vec{v}_1^{+}\\ \vec{v}_2^{+}\end{bmatrix} \tag{4.7a}$$

$$\begin{bmatrix}\vec{b}_1\\ \vec{b}_2\end{bmatrix}=\mathbf{S}^p\begin{bmatrix}\vec{a}_1\\ \vec{a}_2\end{bmatrix}=\begin{bmatrix}\mathbf{S}_{11}^p & \mathbf{S}_{12}^p\\ \mathbf{S}_{21}^p & \mathbf{S}_{22}^p\end{bmatrix}\begin{bmatrix}\vec{a}_1\\ \vec{a}_2\end{bmatrix}. \tag{4.7b}$$

Here, $\mathbf{S}^p$ is the conventional S-parameter obtained from VNA measurement, and used in most circuit simulators. The voltage-mode parameter $\mathbf{S}^v$ is more useful, from the link analysis point of view, because typical receivers work with voltage instead of power. Distinguishing between these two parameters is important, as they differ when the termination loads from two ends are not identical. For real-valued diagonal reference characteristic matrices, $\mathbf{S}^p$ and $\mathbf{S}^v$ are related by [2]:

$$\mathbf{S}^v(i,j)=\mathbf{S}^p(i,j)\sqrt{\mathbf{Z}_o(j)/\mathbf{Z}_o(i)}. \tag{4.8}$$

Note that $\mathbf{S}^v$ is asymmetric when the reference characteristic impedance values are different. In such cases, the magnitude of $\mathbf{S}^v$ can be even greater than unity; unlike $\mathbf{S}^p$, which is always smaller than or equal to unity.

Yet another useful network parameter based on wave quantities is the scattering transmission matrix parameter ($\mathbf{T}$). It relates the incident and reflected input-port wave vectors to the incident and reflected output-port wave vectors as follows:

$$\begin{bmatrix} \vec{b}_1 \\ \vec{a}_1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \vec{a}_2 \\ \vec{b}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_2 \\ \vec{b}_2 \end{bmatrix}. \tag{4.9}$$

Similar formulae for the voltage-mode scattering transmission matrix can be derived, but the details are omitted from this section. Table 4.2 shows the conversion formula between $\mathbf{T}$ and $\mathbf{S}^p$. The advantage of the T-parameter matrix over the S-parameter matrix is that it can be easily cascaded, similar to the ABCD case. For instance, the T-parameter matrix of two cascaded T-parameter systems, from port 1 to port 3, is

$$\begin{bmatrix} \vec{b}_1 \\ \vec{a}_1 \end{bmatrix} = \mathbf{T}_{(1,3)} \begin{bmatrix} \vec{a}_3 \\ \vec{b}_3 \end{bmatrix} = \mathbf{T}_{(1,2)}\, \mathbf{T}_{(2,3)} \begin{bmatrix} \vec{a}_3 \\ \vec{b}_3 \end{bmatrix}. \tag{4.10}$$

Table 4.2 Conversion Formulae for $\mathbf{S}^p$ and $\mathbf{T}$ Matrix Parameters

	$\mathbf{S}^p$	$\mathbf{T}$
$\mathbf{S}^p$	$\begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{S}^p_{11} & \mathbf{S}^p_{12} \\ \mathbf{S}^p_{21} & \mathbf{S}^p_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \end{bmatrix}$	$\mathbf{T}_{11} = -\mathbf{S}^p_{11}\mathbf{S}^{p\,-1}_{21}\mathbf{S}^p_{22} + \mathbf{S}^p_{12}$ $\mathbf{T}_{12} = \mathbf{S}^p_{11}\mathbf{S}^{p\,-1}_{21}$ $\mathbf{T}_{21} = -\mathbf{S}^{p\,-1}_{21}\mathbf{S}^p_{22}$ $\mathbf{T}_{22} = \mathbf{S}^{p\,-1}_{21}$
$\mathbf{T}$	$\mathbf{S}^p_{11} = \mathbf{T}_{12}\mathbf{T}^{-1}_{22}$ $\mathbf{S}^p_{12} = \mathbf{T}_{11} - \mathbf{T}_{12}\mathbf{T}^{-1}_{22}\mathbf{T}_{21}$ $\mathbf{S}^p_{21} = \mathbf{T}^{-1}_{22}$ $\mathbf{S}^p_{22} = -\mathbf{T}^{-1}_{22}\mathbf{T}_{21}$	$\begin{bmatrix} \vec{b}_1 \\ \vec{a}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_2 \\ \vec{b}_2 \end{bmatrix}$

4.1.4 Conversion between Line and Wave Parameters

Because modern circuit simulators can directly accept the S-parameter as a circuit model, the need for converting the S-parameter to other line parameters (such as the Z- or Y-parameter) is not as important as it used to be. However, converting wave-based network parameters (such as the S-parameter) to line-based network parameters is still quite useful, because line-based parameters are often more intuitive, and can be approximated with equivalent circuits in a straightforward

manner. For example, we can generate a wirebond package model with a full-wave EM modeling tool, which results in an accurate broadband S-parameter model. However, if the operating frequency is sufficiently low, this model can be replaced with an inductance matrix by first converting to a **Z** matrix, and then fitting it with a linear function, producing an equivalent inductance matrix. Similar approximation processes can be applied to other small discontinuities, such as vias. (Section 12.2.2 illustrates this approximation process.)

The conversion formulae between the line voltage and voltage wave are quite complicated when compared to the previous conversion formulae used within the same domain. Consequently, this section presents only one conversion formula (between $\mathbf{S}^v$ and **ABCD**), rather than deriving various conversion formulae for different combinations of line parameters versus wave parameters. One can easily perform the conversions between other parameters by first converting to either $\mathbf{S}^v$ or **ABCD**, and then converting to the other parameters in the same domain.

The line voltage and current vectors relate to the incident and reflected voltage and current wave vectors, as follows:

$$\vec{v}_1 = \vec{v}_1^{+} + \vec{v}_1^{-},\ \vec{v}_2 = \vec{v}_2^{+} + \vec{v}_2^{-} \tag{4.11a}$$

$$\vec{i}_1 = \mathbf{Z}_{o,1}^{-1}(\vec{v}_1^{+} - \vec{v}_1^{-}),\ \vec{i}_2 = \mathbf{Z}_{o,2}^{-1}(\vec{v}_2^{+} - \vec{v}_2^{-}) \tag{4.11b}$$

where $\mathbf{Z}_{o,1}$ and $\mathbf{Z}_{o,2}$ are the reference characteristics impedance matrices for two ports. Table 4.3 [4] provides the conversion formula between **ABCD** and $\mathbf{S}^v$. This formula is valid, even for full $\mathbf{Z}_{o,1}$ and $\mathbf{Z}_{o,2}$ matrices, and they do not have to be real matrices. Note that the numerical instability issue, with the two-resistor example in Section 4.1.1, does not appear in this new conversion expression. For instance all the matrices associated with inversion in Table 4.3 are diagonal matrices that are nonsingular. Therefore, the equations in Table 4.3 are always stable for any generalized two-port parameters that have direct connections between the input and output ports, such as transmission lines.

4.1.5 Conversion to Mixed-Mode Parameters

Differential signaling is commonly used for high-speed I/O interfaces. The system response of the differential mode provides a good picture of the channel quality for the first-order analysis. Blockelman and Einsenstadt [5] first introduced the concept of a mixed-mode S-parameter, which is a conventional (single-ended) mode S-parameter converted to differential and common-modes S-parameters. This section reviews this mixed-mode conversion (described in [5]), and generalizes it to handle other network parameters.

For the purpose of the mixed mode conversion, we use the general *N*-port representation of network parameters [1], based on (4.1) to derive a more general expression.

Table 4.3 Conversion Formulae for $\mathbf{S}^v$ and **ABCD** Matrix Parameters

	$\mathbf{S}^v$	**ABCD**
$\mathbf{S}^v$	$\begin{bmatrix}\vec{v}_1^- \\ \vec{v}_2^-\end{bmatrix} = \begin{bmatrix}\mathbf{S}_{11}^v & \mathbf{S}_{12}^v \\ \mathbf{S}_{21}^v & \mathbf{S}_{22}^v\end{bmatrix}\begin{bmatrix}\vec{v}_1^+ \\ \vec{v}_2^+\end{bmatrix}$	$\mathbf{A} = -\Delta[\Omega - \mathbf{S}_{21}^{v\,-1}(\mathbf{S}_{22}^v - \mathbf{I})]$ $\mathbf{B} = \Delta[\Omega - \mathbf{S}_{21}^{v\,-1}(\mathbf{S}_{22}^v + \mathbf{I})]\mathbf{Z}_{o,\mathbf{2}}$ $\mathbf{C} = -\Theta[\Psi - \mathbf{S}_{21}^{v\,-1}(\mathbf{S}_{22}^v - \mathbf{I})]$ $\mathbf{D} = \Theta[\Psi - \mathbf{S}_{21}^{v\,-1}(\mathbf{S}_{22}^v + \mathbf{I})]\mathbf{Z}_{o,\mathbf{2}}$ $\Omega = (\mathbf{I} + \mathbf{S}_{11}^v)^{-1}\mathbf{S}_{12}^v$ $\Psi = (\mathbf{S}_{11}^v - \mathbf{I})^{-1}\mathbf{S}_{12}^v$ $\Delta = [(\mathbf{I} + \mathbf{S}_{11}^v)^{-1}(\mathbf{S}_{11}^v - \mathbf{I}) - \mathbf{I}]^{-1}$ $\Theta = \mathbf{Z}_{o,\mathbf{1}}^{-1}[(\mathbf{S}_{11}^v - \mathbf{I})^{-1}(\mathbf{S}_{11}^v + \mathbf{I}) - \mathbf{I}]^{-1}$
ABCD	$\mathbf{S}_{21}^v = 2(\mathbf{A} + \Delta + \Theta + \Omega)^{-1}$ $\mathbf{S}_{22}^v = 0.5\mathbf{S}_{21}^v(-\mathbf{A} + \Delta - \Theta + \Omega)$ $\mathbf{S}_{11}^v = 0.5(\mathbf{A} + \Delta - \Theta - \Omega)\mathbf{S}_{12}^v$ $\mathbf{S}_{12}^v = 0.5(\mathbf{A} - \Delta - \Theta + \Omega)$ $+0.5(\mathbf{A} + \Delta - \Theta - \Omega)\mathbf{S}_{22}^v$ $\Delta = \mathbf{B}\mathbf{Z}_{o,\mathbf{2}}^{-1}$ $\Theta = \mathbf{Z}_{o,\mathbf{1}}\mathbf{C}$ $\Omega = \mathbf{Z}_{o,\mathbf{1}}\mathbf{D}\mathbf{Z}_{o,\mathbf{2}}^{-1}$	$\begin{bmatrix}\vec{v}_1 \\ \vec{\imath}_1\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\vec{v}_2 \\ \vec{\imath}_2\end{bmatrix}$

First, let us define the differential- and common-mode voltages and currents as follows:

$$\vec{v}^d(i) \equiv \vec{v}(i) - \vec{v}(j),\ \vec{\imath}^{\,d}(i) \equiv \frac{1}{2}[\vec{\imath}(i) - \vec{\imath}(j)] \tag{4.12a}$$

$$\vec{v}^c(i) \equiv \frac{1}{2}[\vec{v}(i) + \vec{v}(j)],\ \vec{\imath}^{\,c}(i) \equiv \vec{\imath}(i) + \vec{\imath}(j) \tag{4.12b}$$

where the *i*th and *j*th lines are assumed to be differential terminals. For a two-line transmission line, the differential and common-mode characteristic impedances (Z_o^d and Z_o^c) are related to the even- and odd-mode characteristic impedances (Z_o^{even} and Z_o^{odd}) as follows:

$$Z_o^d = 2Z_o^{odd},\ Z_o^c = \frac{Z_o^{even}}{2}. \tag{4.13}$$

Using (4.4), and assuming real characteristic impedance matrices, the mixed-mode power waves can be written as:

$$\vec{a}^{\,d} = \frac{1}{\sqrt{2}}[\vec{a}(i) - \vec{a}(j)],\ \vec{b}^{\,d} = \frac{1}{\sqrt{2}}[\vec{b}(i) - \vec{b}(j)] \tag{4.14a}$$

$$\vec{a}^c = \frac{1}{\sqrt{2}}[\vec{a}(i)+\vec{a}(j)], \vec{b}^c = \frac{1}{\sqrt{2}}[\vec{b}(i)+\vec{b}(j)]. \tag{4.14b}$$

Based on the relations given in (4.11–4.13), one can convert any network parameter to a mixed-mode parameter (and back), by first scaling, and then adding or subtracting the corresponding rows and columns of the differential pair, as described in the following paragraphs. These column and row operations produce a fast conversion between a single-ended parameter and mixed-mode parameter.

In general, a multi-conductor system can contain both differential and single-ended lines. In such systems, the coupling between the differential and single-ended lines can be an interesting subject. For instance, a memory I/O interface often uses single-ended signaling for data transmission, but routes the clock signal differentially in order to reduce the clock jitter due to coupling from the data signal. In this case, the coupling factor from the single-ended signal to the differential-mode clock signal is an important design parameter. The generalized conversion formula (similar to the formulation given in [5]), and including both differential and single-ended lines, is provided in the following paragraphs.

First, we define the transformation matrix that converts single-ended variables to differential variables, and vice versa:

$$\vec{v}^m \equiv \begin{bmatrix} \vec{v}^d \\ \vec{v}^c \\ \vec{v}^s \end{bmatrix} = \mathbf{M}_v^{SM}\vec{v} \Leftrightarrow \vec{v} \equiv \begin{bmatrix} \vec{v}^p \\ \vec{v}^n \\ \vec{v}^s \end{bmatrix} = \mathbf{M}_v^{MS}\vec{v}^m \tag{4.15a}$$

$$\vec{i}^m \equiv \begin{bmatrix} \vec{i}^d \\ \vec{i}^c \\ \vec{i}^s \end{bmatrix} = \mathbf{M}_i^{SM}\vec{i} \Leftrightarrow \vec{i} \equiv \begin{bmatrix} \vec{i}^p \\ \vec{i}^n \\ \vec{i}^s \end{bmatrix} = \mathbf{M}_i^{MS}\vec{i}^m \tag{4.15b}$$

$$\vec{a}^m \equiv \begin{bmatrix} \vec{a}^d \\ \vec{a}^c \\ \vec{a}^s \end{bmatrix} = \mathbf{M}_{a,b}^{SM}\vec{a} \Leftrightarrow \vec{a} \equiv \begin{bmatrix} \vec{a}^p \\ \vec{a}^n \\ \vec{a}^s \end{bmatrix} = \mathbf{M}_{a,b}^{MS}\vec{a}^m \tag{4.15c}$$

$$\vec{b}^m \equiv \begin{bmatrix} \vec{b}^d \\ \vec{b}^c \\ \vec{b}^s \end{bmatrix} = \mathbf{M}_{a,b}^{SM}\vec{b} \Leftrightarrow \vec{b} \equiv \begin{bmatrix} \vec{b}^p \\ \vec{b}^n \\ \vec{b}^s \end{bmatrix} = \mathbf{M}_{a,b}^{MS}\vec{b}^m \tag{4.15d}$$

where $\vec{x}^p$ and $\vec{x}^n$ represent the single-ended parameters for positive and negative terminals, and $\vec{x}^s$ represents the stand-alone single-ended parameters. $\vec{x}^d$ and $\vec{x}^c$ represent the differential and common-mode parameters, respectively. The transformation matrices ($\mathbf{M}_v^{SM}$, $\mathbf{M}_i^{SM}$, and $\mathbf{M}_{a,b}^{SM}$) convert the single-ended line voltage, line current, and power waves to the equivalent mixed-mode variables. The counterpart transformation matrices ($\mathbf{M}_v^{MS}$, $\mathbf{M}_i^{MS}$, and $\mathbf{M}_{a,b}^{MS}$) convert the mixed-mode

line voltage, line current, and power waves to the equivalent single-ended variables. These matrices can be constructed based on (4.12) and (4.14).

The conversion formula, used to convert a single-ended network parameter to the mixed-mode parameter, is derived as follows:

$$\mathbf{Z}^M = \mathbf{M}_v^{SM}\mathbf{Z}\mathbf{M}_i^{MS} \Leftrightarrow \mathbf{Z} = \mathbf{M}_v^{MS}\mathbf{Z}^M\mathbf{M}_i^{SM} \tag{4.16a}$$

$$\mathbf{Y}^M = \mathbf{M}_i^{SM}\mathbf{Y}\mathbf{M}_v^{MS} \Leftrightarrow \mathbf{Y} = \mathbf{M}_i^{MS}\mathbf{Y}^M\mathbf{M}_v^{SM} \tag{4.16b}$$

$$\mathbf{S}^{p^M} = \mathbf{M}_{a,b}^{SM}\mathbf{S}^p\mathbf{M}_{a,b}^{MS} \Leftrightarrow \mathbf{S}^p = \mathbf{M}_{a,b}^{MS}\mathbf{S}^{p^M}\mathbf{M}_{a,b}^{SM} \tag{4.16c}$$

$$\mathbf{S}^{v^M} = \mathbf{M}_v^{SM}\mathbf{S}^v\mathbf{M}_v^{MS} \Leftrightarrow \mathbf{S}^v = \mathbf{M}_v^{MS}\mathbf{S}^{v^M}\mathbf{M}_v^{SM}. \tag{4.16d}$$

The final mixed-mode parameter can be put into the following form:

$$\mathbf{X}^M = \begin{bmatrix} \mathbf{X}^{dd} & \mathbf{X}^{cd} & \mathbf{X}^{sd} \\ \mathbf{X}^{dc} & \mathbf{X}^{cc} & \mathbf{X}^{sc} \\ \mathbf{X}^{ds} & \mathbf{X}^{cs} & \mathbf{X}^{ss} \end{bmatrix} \tag{4.17}$$

where the superscript *ij* indicates the conversion from the *i*-mode to *j*-mode. For instance, $\mathbf{X}^{dc}$ indicates the conversion of the differential mode to common mode. Table 4.4 summarizes how one can fill these matrices for converting single-ended parameters to mixed-mode parameters. For stand-alone single-ended parameters, the entry is simply 1 for the corresponding diagonal term.

Figure 4.3 shows an example of six transmission lines, which consist of four differential conductors (two pairs), and two single-ended conductors. The following equations show the conversion matrices $\mathbf{M}_v^{SM}$ for the column- and row-wise orders. (4.18a) is the column-wise case, and (4.18b) is the row-wise case.

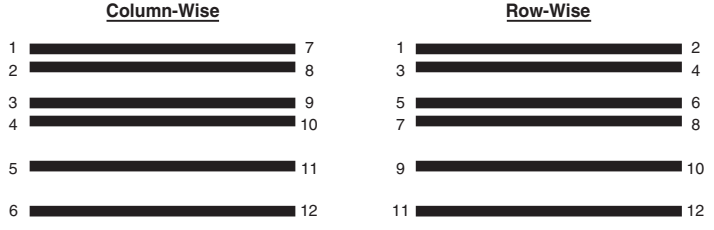


Figure 4.3 Sample Mixed-Mode Conversion with Six Conductors: S-Parameters Are Measured Using Both Column-Wise, and Row-Wise Orders

Table 4.4 Transform Matrices to Convert Single-Ended Variables to Mixed-Mode Variable and Vice Versa

	Single-Ended to Mixed-Mode	Mixed-Mode to Single-Ended
	i: single-ended positive index j: single-ended negative index m: differential-mode index n: common-mode index	i: differential-mode index j: common-mode index m: single-ended positive index n: single-ended negative index
$\vec{v}$	$\begin{matrix} & i & j \\ m & 1 & -1 \\ n & 1/2 & 1/2 \end{matrix} \Rightarrow \mathbf{M}_v^{SM}$	$\begin{matrix} & i & j \\ m & 1/2 & 1 \\ n & -1/2 & 1 \end{matrix} \Rightarrow \mathbf{M}_v^{MS} = \mathbf{M}_v^{SM^{-1}}$
$\vec{i}$	$\begin{matrix} & i & j \\ m & 1/2 & -1/2 \\ n & 1 & 1 \end{matrix} \Rightarrow \mathbf{M}_i^{SM}$	$\begin{matrix} & i & j \\ m & 1 & 1/2 \\ n & -1 & 1/2 \end{matrix} \Rightarrow \mathbf{M}_i^{MS} = \mathbf{M}_i^{SM^{-1}}$
$\vec{a},\vec{b}$	$\begin{matrix} & i & j \\ m & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ n & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{matrix} \Rightarrow \mathbf{M}_{a,b}^{SM}$	$\begin{matrix} & i & j \\ m & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ n & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{matrix} \Rightarrow \mathbf{M}_{a,b}^{MS} = \mathbf{M}_{a,b}^{SM^T}$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{bmatrix} \rightarrow \begin{bmatrix} v_1^d = v_1 - v_2 \\ v_2^d = v_3 - v_4 \\ v_3^d = v_7 - v_8 \\ v_4^d = v_9 - v_{10} \\ v_1^c = v_1 + v_2 \\ v_2^c = v_3 + v_4 \\ v_3^c = v_7 + v_8 \\ v_4^c = v_9 + v_{10} \\ v_5 \\ v_6 \\ v_{11} \\ v_{12} \end{bmatrix} \Rightarrow \mathbf{M}_v^{SM} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.18a)$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{bmatrix} \rightarrow \begin{bmatrix} v_1^d = v_1 - v_3 \\ v_2^d = v_2 - v_4 \\ v_3^d = v_5 - v_7 \\ v_4^d = v_6 - v_8 \\ v_1^c = v_1 + v_3 \\ v_2^c = v_2 + v_4 \\ v_3^c = v_5 + v_7 \\ v_4^c = v_6 + v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{bmatrix} \Rightarrow \mathbf{M}_v^{SM} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18b}$$

4.2 Preparing an Accurate S-Parameter Time-Domain Model

Although the S-parameter was used for several decades in RF applications, it was used mainly in the frequency domain. The recent application of the S-parameter to time-domain simulations has raised several numerical issues. For example, in a typical frequency-domain analysis, the knee frequency F_{knee}, (defined as $0.5/t_{rise}$) covers sufficient energy for digital signals [6]. However, as demonstrated in a later example, this knee frequency could still lead to significant inaccuracy in a time-domain simulation. This section covers several issues associated with broadband modeling and presents several tips for building accurate broadband S-parameter models.

4.2.1 Accuracy of Time-Domain (TDR/T) and Frequency-Domain (VNA) Measurements

The signal-integrity community has had many debates about comparing the accuracy of the S-parameter model, based on time-domain and frequency-domain measurement methods [7] [8]. Many of these discussions focused on the dynamic range of the measurement instruments used to compare the accuracy of measured data. This type of accuracy comparison is useful in RF applications, but it is less useful for digital applications, because both TDR and VNA devices can provide sufficiently accurate measurements. With reasonable instrument bandwidths, and proper calibration procedures, any error introduced by either of these two devices is negligible, when compared to the process or manufacturing variations typically observed in digital systems. A more critical issue, and one often neglected, is the model's numerical stability in transient time-domain simulations.

In digital applications, we should compare the measurement accuracy in the time domain, rather than in the frequency domain, because measured models are ultimately used for transient simulation. For instance, the comparison of step or single-bit responses from the two measurement methods is more meaningful. Under this condition, the time-domain measurement could be more advantageous, whereas the frequency domain data would suffer additional errors, such as inaccurate low-frequency measurement data and numerical errors associated with the inverse Fourier transformation. These drawbacks of frequency-domain models are often overlooked.

Unfortunately, most modern circuit simulators do not support network models using time-domain representation, such as impulse or step responses. As a result, you must convert even time-domain measurement data to the frequency domain first. This results in two domain transformations for transient analysis. These two domain transformations result in an error that is worse than the error in the frequency-domain measurement. Agilent recognized this problem, and added support for the time-domain impulse model [9]. The balance of this section focuses on improving frequency-domain models, because most models are still generated in the frequency domain.

4.2.2 Maximum Frequency Range

The specification of the maximum frequency range for the S-parameter model is perhaps one of the most controversial issues in the field of frequency-domain modeling and measurements. Johnson's proposal of the knee frequency F_{knee} seems to cover enough of the energy spectrum for digital signals [6]. Yet, many different sources advocate for an even higher frequency, and claim that F_{knee} is no longer valid for high-speed operation. Three, or even five times, the Nyquist frequency has been proposed as the maximum range. Occasionally, one can observe that, even with the same input-signal frequency content, the required maximum frequency range changes, depending on the characteristics of the passive channels.

The confusion arises from the fact that, when engineers were trying to determine the maximum frequency, they focused solely on capturing the energy content of the digital signals, and often ignored the numerical side effect related to transforming the frequency-domain data to the time domain. The knee frequency is perfectly valid, if we perform all the analyses solely in the frequency domain. It is also valid if we can transform the frequency-domain channel model to the time domain without a numerical side effect. However, in practice, we must consider the numerical side effect in the broadband model; the knee frequency is no longer sufficient. This numerical side effect is even more severe for electrically short circuit elements due to their slowly decaying nature (as demonstrated in the next example). Furthermore, if some of subcomponent models (such as vias, connectors, pads, and balls) are generated without a specific target data rate, the knee frequency cannot even be defined.

To demonstrate the modeling issues associated with electrically short elements, consider a simple example of two S-parameter models for very short (0.5 cm), and long (30 cm), lossy

transmission lines. The short line response could be representative of a short via, solder ball, or connector pins. The long line represents a long backplane trace. The HSPICE W-element is used for simulation with the following parameters: $L_o = 30\text{nH/m}$, $C_o = 120\text{pF/m}$, $R_o = 1.74\,\Omega/\text{m}$, $G_o = 0$, $R_s = 14.7\text{m}\Omega/\sqrt{\text{Hz}}/\text{m}$, and $G_d = 12.8\text{S/Hz/m}$. Figure 4.4 shows the S21 responses. Assuming the final channel consists of these two transmission lines in series, 50GHz is a large enough frequency range to capture this channel response, regardless of the signal spectrum contents, because the channel effectively filters out all the high-frequency contents. To increase the time-domain resolution, we pad the frequency responses with zeros, up to 150GHz.

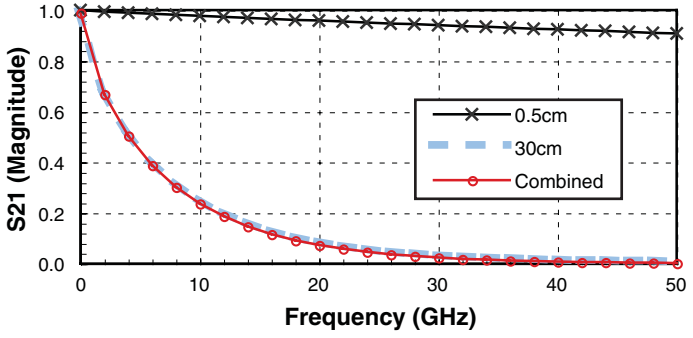


Figure 4.4 S21 Responses of Short (0.5cm), Long (30cm), and Combined Case

Consider two methods of computing the time-domain responses. The first method computes the overall frequency-domain response by multiplying the two transmission line responses. Then, the time-domain response is obtained by taking IFFT of the overall response. The second method computes the impulse responses of the short and long transmission lines by taking IFFT of the individual frequency responses. Figure 4.5(a) shows the time-domain response of the short transmission line. Clearly, the impulse response of the short line suffers from the aliasing effect and windowing. Padding zeros is effectively the same as applying the rectangular window, and causes ringing (Gibbs phenomena) in the time-domain waveform. However, this ringing is eventually filtered out after the response passes through the long transmission line, as shown in Figure 4.5(b).

The major issue with a short element is that the frequency response decays extremely slowly, and even with truncation at relatively high frequency, it can lead to aliasing problems. Due to ringing, this aliasing effect is hard to detect, and we can observe it accurately only after filtering out the ringing. Figure 4.6 shows the impulse responses of the combined channel response, using frequency-domain multiplication and time-domain convolution. Although a discrepancy exists in the two impulse responses, predicting the impact on the real signal is difficult. To better understand this impact, the single-bit (pulse) responses are calculated. Figure 4.7 shows the single-bit

responses of the short, long, and combined channel. Figure 4.8 compares the frequency-domain multiplication and time-domain convolution. Considering the fact that the impact of the short transmission line must be small in this case, the magnitude of error shown in Figure 4.8 is fairly significant. Moreover, the error can increase, as we add more short S-parameter elements corresponding to a channel with many connectors, via transitions, and small interconnections.

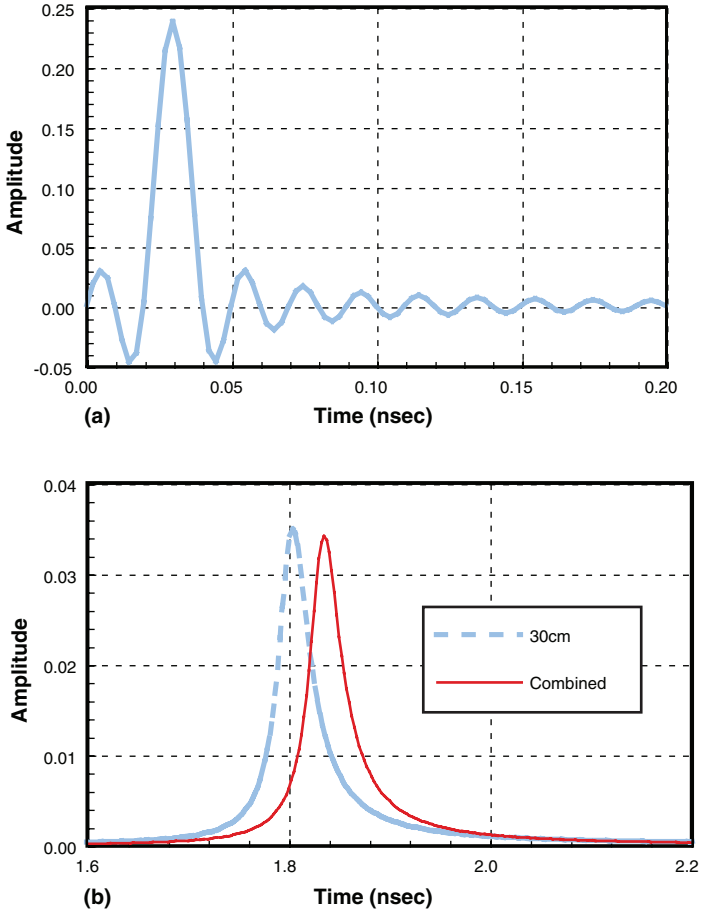


Figure 4.5 Impulse Responses of (a) Short Transmission Line, and (b) Long Transmission Line and Combined Case

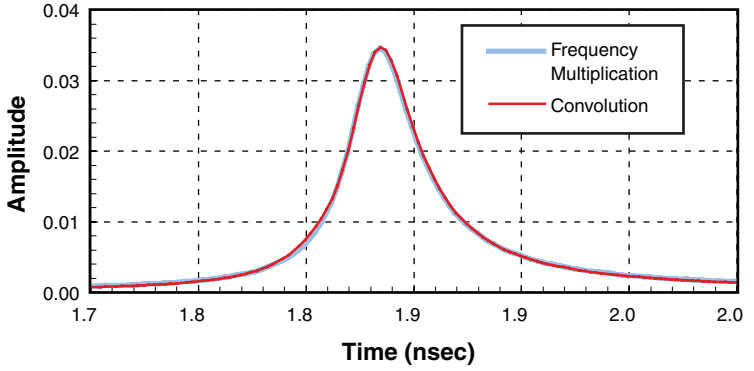


Figure 4.6 Impulse Responses Based on IFFT of Overall Response and Convolution of Two Impulse Responses

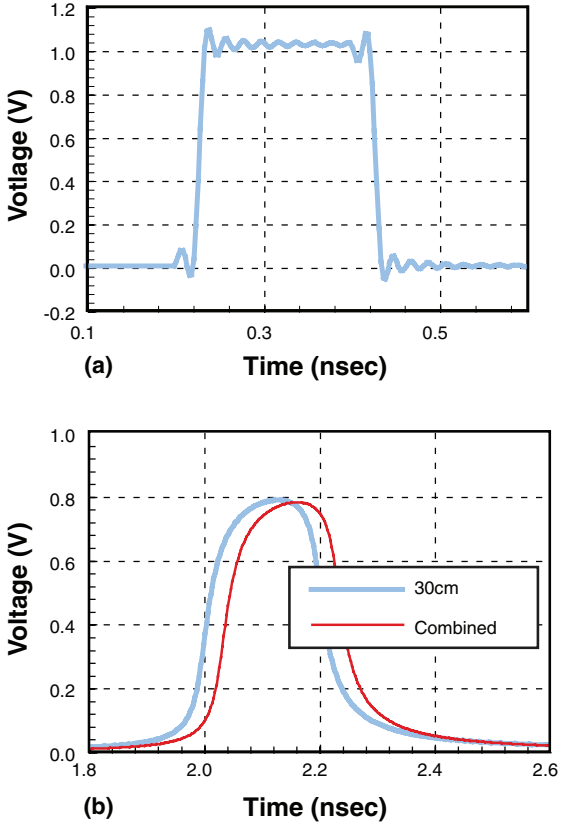


Figure 4.7 Single-Bit Responses of (a) Short Transmission Line, and (b) Long Transmission Line and Combined Case

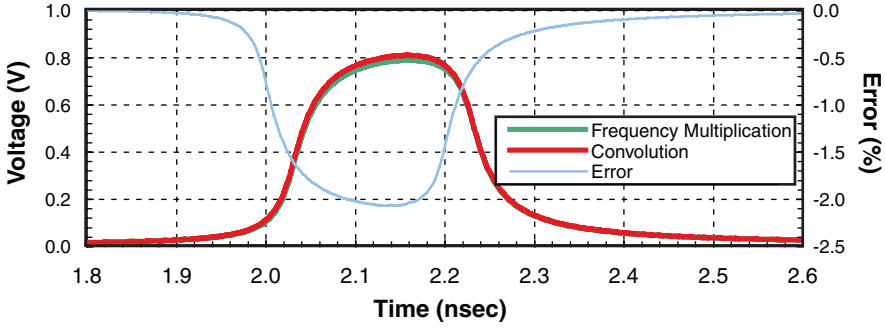


Figure 4.8 Single-Bit Responses Based on IFFT of Overall Response and Convolution of Two Impulse Responses

This aliasing error cannot be avoided, because circuit simulators do not combine the frequency-domain models before converting to the time-domain model. Consequently, we must provide an S-parameter that captures the complete frequency behavior, regardless of the actual signal data rate.

To summarize, the following practices are recommended when using the S-parameter for time-domain simulation:

- If only frequency-domain analysis is used, then F_{knee} can be used as the baseline for determining the maximum frequency range. The remaining recommendations apply to time-domain analysis.
- Use the cutoff frequency of the S-parameter response, which captures all the dynamics of the frequency response, as the baseline for determining the maximum frequency range (rather than the signal bandwidth).
- If possible, try to merge the S-parameter models in the frequency domain to avoid an aliasing problem. This will also speed up the transient simulation.
- Try to generate the S-parameter model for the entire channel in the frequency domain. (Many of today's circuit simulators support S-parameter generation.)
- If the frequency response is still flat or not band limited, apply a filter (window) [10] to limit the bandwidth. Apply the filter only to the high-frequency region beyond F_{knee} to minimize the aliasing and windowing effect.
- Consider the lumped element representation when the S-parameter has a flat response. Small (short) elements do not need to be modeled using an S-parameter. Use the formulae in Section 4.1.4 to convert the S-parameter to other Z- or Y-parameters, and fit with lump elements. Do not use the S-parameter for everything. (This is particularly true for crosstalk responses. In fact, the frequency response of near-end crosstalk has a non-zero steady value.)

- For S-parameter models with both long delays and a flat response, consider using transmission lines to approximate the model. The transmission-line simulation algorithm does not suffer from the instability issue associated with S-parameter models (as discussed in Chapter 5, "Transmission Lines").
- When an S-parameter is measured or generated from simulation, use the linear frequency steps to guarantee a smooth phase response. The non-linear frequency steps are good for capturing slow magnitude response, but may result in data points that are too coarse for phase response. If the existing measurement data does not have enough points for phase response, add interpolated points to aid the simulator.
- Provide accurate DC values (see the following section for more details).
- Filter measurement noise and check for the passivity. (Section 4.3 reviews the necessary passivity condition.)

So far, the assumption has been that the S-parameter models are used in the time domain using direct convolution. Due to advances in the macro-modeling field, a recursive convolution is considered as an alternative solution for time-domain simulation. A recursive convolution relies on the rational function approximation of the frequency domain response. It has been widely used for transmission-line modeling (see Chapter 5). Although the concept of fitting the frequency data with a known function may not suffer from aliasing and truncation problems, it has issues that are more complex. These issues are related to its approximation accuracy, which involves fitting a highly oscillatory S-parameter response using a smooth rational function. Much work has been done to improve the accuracy of approximation over the last decade, but it still remains challenging due to numerical stability issues. The current research in this area can be found in Triverio and Grivet-Talocia [11].

4.2.3 Accurate DC Modeling

One of the most critical bottlenecks for frequency domain-based techniques is lack of DC (zero frequency) values. This section demonstrates the importance of DC values using step responses. For the sake of simplicity, the system is assumed to be linear without loss of generality. The digital system response in terms of step responses can be written as follows:

$$y(t) = \sum_k (a_k - a_{k-1})u(t - kT) \tag{4.19}$$

where a_k is the input symbol, and $u(t)$ is the step response. This expression emphasizes that the step response has a direct impact on the final accuracy of the system response. Applying the initial and final theorems to the step response, we have:

$$\lim_{t\to 0+} u(t) = \lim_{s\to\infty}[sU(s)] = \lim_{s\to\infty}[H(s)] \tag{4.20a}$$

$$\lim_{t\to\infty} u(t) = \lim_{s\to 0}[sU(s)] = \lim_{s\to 0}[H(s)]. \tag{4.20b}$$

It is now clear that the infinity value of the frequency response of the channel directly determines the instant time response, whereas the DC value of the frequency response influences the steady-state value of the step response in time.

Now we use the example shown in the previous section once again, to demonstrate the impact of DC values. Modeling DC or very low frequency points using frequency-domain techniques is challenging, because the wavelength is large and the coupling between the electrical and magnetic fields is weak. This example uses the long transmission line, only to isolate the accuracy issue with the short line. Two different DC values are estimated using the step and linear extrapolation methods. The calculated DC values are 0.98 and 1.0, for the step and linear extrapolations, respectively. Figure 4.9 and Figure 4.10 show the corresponding time-domain impulse and step responses. Although the impulse response does not show a significant difference, due to its high-frequency nature, the step response clearly shows a considerable difference.

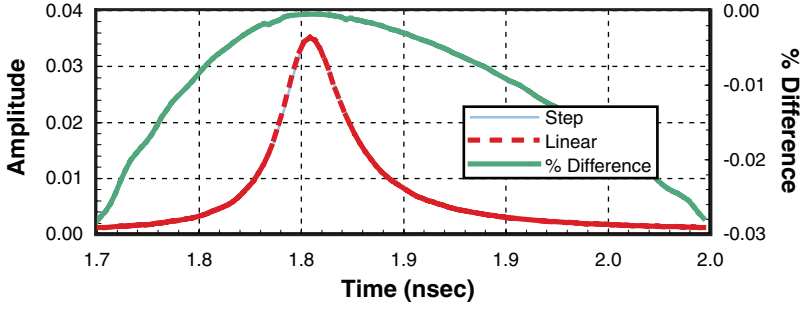


Figure 4.9 Impulse Responses Based on Step and Linear DC Point Interpolations

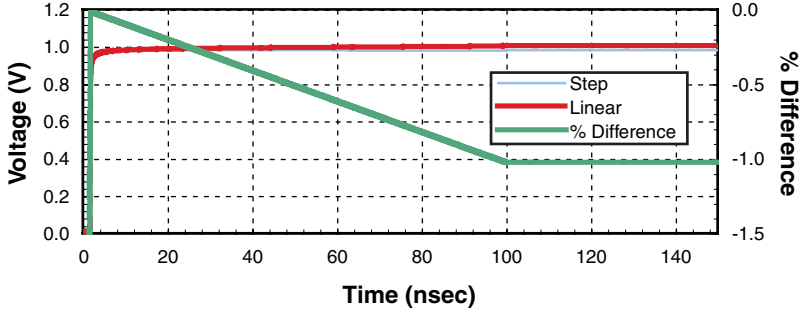


Figure 4.10 Step Responses Based on Step and Linear DC Point Interpolations

As expected (and shown in Figure 4.10), the DC value difference causes the steady-state offset in the step response. A direct correlation exists between the difference in DC values and the

steady-state offset. In fact, the final steady-state value is the DC value illustrated in (4.20b). Therefore, a better estimation of S-parameter DC values is obtained using the time-domain step responses, TDR and TDT. To verify the results, Figure 4.11 provides a closer view of the transition region; where one finds no significant error.

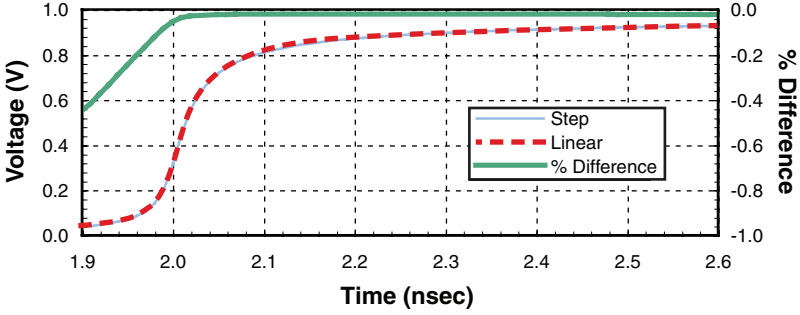


Figure 4.11 Enlarged View of Step Responses Near the Transition

When the time-domain measurement data are not available, more elegant extrapolation methods (based on causality or passivity conditions) can be used instead of simple numerical extrapolation techniques, such as linear and spline interpolations [12] [13]. Section 4.4.5 briefly describes this approach, which is based on the generalized dispersion relations [20] [21]. Transmission-line parameter extraction also has a similar DC modeling issue (discussed in Section 5.4.4). (The actual measurement example is also found in that section.)

4.3 Passivity Conditions

Measurement data containing noise is common, and it can even violate the passivity condition. This is particularly true for small coupling terms, which are more subject to numerical error (as discussed in Section 4.2.2). Repairing the passivity violations in the measurement data or macro model has become a subject of special interest [14] [15]. In this section, the basic passivity conditions are informally derived and reviewed. The passivity conditions for both the Z- and S-parameters are discussed. We start with the S-parameter first, because it provides a better physical model with which to define the passivity condition. We use an N-terminal representation of the S-parameter, defined in (4.1), in this section, because the passivity conditions are applicable to any passive structure other than transmission lines. More detailed derivations and descriptions are available [11] [16] [17].

4.3.1 Passivity Conditions for S-Parameter

The word *passivity*, in general, denotes that a system is incapable of generating energy. Such systems can only absorb energy from external sources. In other words, the net energy (sum or integral of power) produced by the difference in the input (incident) and output (reflected) powers

must be non-negative. This leads to the following definition of passivity for the S-parameter for the time domain:

$$\sum_{i=1}^{N} \int_{-\infty}^{t} [a_i^2(\tau) - b_i^2(\tau)]d\tau = \int_{-\infty}^{t} [\vec{a}^T\vec{a} - \vec{b}^T\vec{b}]d\tau \geq 0. \tag{4.21}$$

This definition can be applied to general systems, including both lumped and distributed cases. The integral equation allows the system to generate energy after it absorbs it, but not the other way around. Consequently, it is implicit that passive systems are causal. In fact, it can be explicitly shown that a passive system satisfying (4.21) is also causal [11].

Alternatively, this more general version of the passivity condition is used in [16]:

$$\int_{-\infty}^{\infty} [\vec{a}^T\vec{a} - \vec{b}^T\vec{b}]d\tau \geq 0. \tag{4.22}$$

The main difference between (4.21) and (4.22) is that the passive system satisfied by (4.21) is always causal, whereas the passive system under (4.22) can still be non-causal, particularly for timing varying or non-linear systems. We use (4.21) in this section, because it is safe to assume that most physical systems are causal. In addition, for linear timing invariant (LTI) systems, all passive systems are causal, and both definitions lead to the same passivity conditions [16]. By definition, any non-causal system is not passive; however, not all causal systems are passive, so enforcing causality does not guarantee passivity.

For LTI systems, the passivity definition (4.21) can be written as follows [16]:

- each element of $\mathbf{S^p}(s)$ is analytic in $\text{Re}\{s\} > 0$
- $\mathbf{S^p}(s^*) = \mathbf{S^{p*}}(s)$
- $\mathbf{I} - \mathbf{S}^{pH}(s)\mathbf{S^p}(s)$ a nonnegative definite matrix for all $\text{Re}\{s\} > 0$

The superscript * denotes the complex conjugate, and the superscript H denotes conjugate-transpose. $\mathbf{S^p}$ is the scattering parameter matrix. A matrix satisfying the preceding conditions is called *bounded-real*. A matrix $\mathbf{B}$ is nonnegative if $\vec{x}^H\mathbf{B}\vec{x} \geq 0$ for any constant vector $\vec{x}$.

By definition, passive systems map real inputs to real outputs, so their impulse responses must be real, as implied by the second condition. To informally prove the third condition, consider the lossless system first. Under this assumption, we have $\sum|a_i|^2 = \sum|b_i|^2$ which leads to:

$$\vec{b}^H\vec{b} = (\mathbf{S}^p\vec{a})^H\mathbf{S}^p\vec{a} = \vec{a}^H\mathbf{S}^{pH}\mathbf{S}^p\vec{a} \tag{4.23}$$

and then:

$$\mathbf{I} = \mathbf{S}^{pH}\mathbf{S}^{p} \Leftrightarrow \mathbf{I} - \mathbf{S}^{pH}\mathbf{S}^{p} = 0. \tag{4.24}$$

Now, in the case of a lossy system, we have $\sum |a_i|^2 > \sum |b_i|^2$ which leads to:

$$\vec{a}^{H}\vec{a} > \vec{b}^{H}\vec{b} \Rightarrow \vec{a}^{H}\vec{a} > (\mathbf{S}^{p}\vec{a})^{H}\mathbf{S}^{p}\vec{a} \Rightarrow \vec{a}^{H}\vec{a} > \vec{a}^{H}\mathbf{S}^{pH}\mathbf{S}^{p}\vec{a} \tag{4.25}$$

and then:

$$\mathbf{I} > \mathbf{S}^{pH}\mathbf{S}^{p} \Leftrightarrow \mathbf{I} - \mathbf{S}^{pH}\mathbf{S}^{p} > 0. \tag{4.26}$$

Both lossless and lossy systems that are passive satisfy the third non-negative definite matrix condition. Note that the positive definiteness can be checked using the following relation:

$$\mathrm{Re}[Eig(\mathbf{I} - \mathbf{S}^{pH}\mathbf{S}^{p})] > 0. \tag{4.27}$$

The preceding Laplace-domain conditions are general, and they can be applied to both lumped and distributed systems under the LTI assumption. However, these conditions are not practical, because they require the verification of the entire half-plane. Fortunately, similar conditions for the frequency domain can be derived as follows [11]:

- $\mathbf{S^p}(j\omega)$ is causal, or equivalently, satisfies the Kramers-Kronig dispersion relations
- $\mathbf{S^p}(-j\omega) = \mathbf{S^{p^*}}(j\omega)$
- $\mathbf{I} - \mathbf{S^{pH}}(j\omega)\mathbf{S^p}(j\omega)$ is a non-negative definite matrix for all ω

Similar to the Laplace-domain case, one can apply these conditions to both lumped and distributed systems.

4.3.2 Passivity Conditions for Z- and Y-Parameters

This section describes the passivity conditions for impedance and admittance matrices **Z** and **Y**. As demonstrated later, the final frequency-domain expression is limited to lumped systems, and is somewhat more inconvenient than the S-parameter case. Consequently, use the S-parameter for passivity checking, instead of the impedance or admittance matrices.

Substituting (4.11) for the passivity definition (4.21):

$$\sum_{i=1}^{N} \int_{-\infty}^{t} v_i(\tau) i_i(\tau) d\tau = \int_{-\infty}^{t} \vec{v}^{T}\vec{i}\, d\tau \geq 0. \tag{4.28}$$

The vectors $\vec{v}$ and $\vec{i}$ represent line voltage and currents, respectively. This definition can be applied to both lumped and distributed cases.

For LTI systems, the passivity definition (4.28) can be written as [16]:

- each element of $\mathbf{Z}(s)$ is analytic in $\mathrm{Re}\{s\} > 0$
- $\mathbf{Z}(s^*) = \mathbf{Z}^*(s)$
- $\mathbf{Z}^H(s) + \mathbf{Z}(s)$ is a non-negative definite matrix for all s such that $\mathrm{Re}\{s\} > 0$

A matrix satisfying the above conditions is called *positive-real*. The third expression can be readily derived from the S-parameter condition ($\mathbf{I} - \mathbf{S}^{pH}\mathbf{S}^p = 0$) by using the conversion formula shown in (4.2a). A similar definition can be derived for the admittance matrix $\mathbf{Y}$. The earlier Laplace-domain conditions can be applied to both lumped and distributed systems under the LTI assumption. However, again, it is not practical, as it requires the verification of the entire half-plane.

The corresponding frequency domain conditions are [11]:

- each element of $\mathbf{Z}(s)$ is analytic in $\mathrm{Re}\{s\} > 0$
- $\mathbf{Z}(-j\omega) = \mathbf{Z}^*(j\omega)$
- $\mathbf{Z}^H(j\omega) + \mathbf{Z}(j\omega)$ is a non-negative definite matrix for all ω except for simple poles $j\omega_o$ of $\mathbf{Z}(s)$, where the residue matrix must be nonnegative definite
- asymptotically, $\mathbf{Z}(s) \rightarrow \mathbf{A}s$ in $\mathrm{Re}\{s\} > 0$, where A is a real, constant, symmetric, non-negative-definite matrix

Unlike the S-parameter case, the preceding conditions are not completely free from the verification of the entire half plane. In addition, these conditions are valid only for lumped systems; they cannot be used for distributed systems, such as transmission lines. This is a critical drawback, because macro modeling has been widely used, even for distributed systems. The fact that macro modeling uses a rational function, which represents lumped systems, does not change the underlying property of distributed systems; so, it does not justify using the preceding passivity conditions. However, many macro modeling papers have misused the preceding conditions to measure data representing distributed systems. It is also important to note that a perfectly passive distributed system, which passes the S-parameter passivity test, can be detected as non-passive by using the earlier test for impedance. The possibility exists that such data could result in rational function impedance models that are non-passive.

The main difficulty in verifying the passivity of distributed systems is due to a delay that causes oscillation in an impedance or admittance matrix, or even in the S-parameter. Passivity testing or correction is more effective, after the delay is explicitly removed from such parameters [14]. Delays can be extracted analytically for transmission lines, as described in the next chapter. However, such delay extraction is impossible with general S-parameters, or even with non-uniform transmission lines.

4.4 Causality Conditions

In the previous section, we informally showed that the passivity condition inherently assumes that the system is causal. The S-parameter passivity condition was defined only along the imaginary axis, assuming the system is causal. This section discusses the causality condition, as well as numerical issues that are associated with checking the causality condition.

A LTI system is causal if, and only if, the impulse response $h(t)$ is vanishing for $t < 0$ [16], or, in other words:

$$h(t) = 0, t < 0. \tag{4.29}$$

The frequency-domain counterpart of the preceding equation is referred to as a *dispersion relation*. It can be derived by expressing $h(t)$ as:

$$h(t) = \text{sgn}(t)h(t) \tag{4.30}$$

where $\text{sgn}(t)$ is 1 for $t > 0$ and –1 for $t < 0$. Now, by applying the Fourier transform, we have:

$$\mathbb{F}\{h(t)\} \equiv H(j\omega) = \frac{1}{2\pi} \Im\{\text{sgn}(t)\} * \Im\{h(t)\}$$

$$= \frac{1}{j\pi} P.V. \int \frac{H(j\omega')}{\omega - \omega'} d\omega' \equiv \mathbb{H}\{H(j\omega)\}. \tag{4.31}$$

Here, $\mathbb{F}\{\cdot\}$ and $\mathbb{H}\{\cdot\}$ denote the Fourier and Hilbert transformations, respectively. P.V. stands for Cauchy's principal integral:

$$P.V. \int = \lim_{\varepsilon \to 0^+} \left[\int_{-\infty}^{\omega-\varepsilon} + \int_{\omega+\varepsilon}^{+\infty} \right]. \tag{4.32}$$

Now, let $H(j\omega) = U(\omega) + jV(\omega)$, where $U(\omega)$ and $V(\omega)$ are the real and imaginary parts of $H(j\omega)$, respectively. Then, equating the real and imaginary parts of (4.31), we have:

$$U(\omega) = \frac{1}{\pi} P.V. \int \frac{V(\omega')}{\omega - \omega'} d\omega' + U_\infty \tag{4.33a}$$

$$V(\omega) = -\frac{1}{\pi} P.V. \int \frac{U(\omega') - U_\infty}{\omega - \omega'} d\omega'. \tag{4.33b}$$

Note that $U(\omega)$ must be an even function, and $V(\omega)$ must be an odd function. The preceding equations are known as the Kramers-Kronig dispersion relations, or Hilbert transform. Now, before proceeding with the causality condition, we describe a useful property for a minimum-phase system, where both the system and its inverse are causal and stable. The magnitude and phase of a minimum-phase system are related by:

$$\arg[H(j\omega)] = -\mathbb{H}\{\log|H(j\omega)|\} \tag{4.34a}$$

$$\log|H(j\omega)| = \log|H(\infty)| + \mathbb{H}\{\arg[H(j\omega)]\}. \tag{4.34b}$$

The causality violation can be checked by measuring the error between the original data and the Hilbert transformed data, as follows:

$$\Delta(j\omega) = H(j\omega) - \hat{H}(j\omega) \tag{4.35}$$

where $\hat{H}(j\omega)$ is the reconstructed transfer function of $H(j\omega)$ using the Hilbert transformation. The Hilbert transformation can be performed using any numerical integration scheme, or the discrete Hilbert transformation. In [18], Young and Bhandal derive a direct transformation expression, based on a piece-wise linear approximation that bypasses numerical quadrature schemes.

Although the preceding condition (4.35) is conceptually straightforward, it can be quite challenging, due to the numerical side effects. Typical measurement or simulation data provides only limited frequency information, whereas the frequency responses of the network parameters are not necessarily band-limited, or could be wide-band in nature. This is particularly true for coupling terms (as described in Section 4.2.2). In addition to this truncation error, a discretization error can also be significant, when compared to the causality violation amount. After all, the causality violation may have resulted from inaccuracies in the numerical modeling or measurements. Consequently, any numerical errors due to frequency truncation and discretization could be interpreted as causality violations in (4.35). This is why checking a causality violation is numerically very challenging. Moreover, one of the conditions for checking the passivity of the S-parameter requires checking the causality, as described in the previous section. In fact, the causality condition is perhaps the most difficult condition to test of the three conditions for the passivity test.

A rigorous approach to bound the numerical errors, associated with testing the causality condition, is presented by Triverio and Grivet-Talocia [19] and Asgari, Lalgudi, and Tsuk [20] and discussed in the following section.

4.4.1 Generalized Dispersion Relations

To minimize the numerical error, the generalized dispersion relations (or generalized Hilbert transform) is introduced [19]:

$$H_N(j\omega) = L_H(j\omega) + \frac{\prod_{q=1}^{N}(\omega - \omega_q)}{j\pi} \cdot P.V. \int \frac{H(j\omega') - L_H(j\omega')}{\prod_{q=1}^{N}(\omega' - \omega_q)} \frac{d\omega'}{\omega - \omega'} \tag{4.36}$$

where the so-called *subtraction points* $\{\omega_q\}_{q=1}^{N}$ are spread over the available frequency range Ω. $L_H(j\omega)$ represents the Lagrange interpolation polynomial for $H(j\omega)$:

$$L_H(j\omega) = \sum_{q=1}^{N} H(j\omega_q) \prod_{p=1;\, p\neq q}^{N} \frac{(\omega - \omega_p)}{(\omega_q - \omega_p)}. \tag{4.37}$$

Equation (4.36) reduces to the original Hilbert transformation for $N = 0$ as $L_H(j\omega) = 0$, and the Π terms disappear. The interpolation points are referred to as subtraction points, and the earlier dispersion relations as *dispersion relations with subtractions*. Now, the causality error expression (4.35) can be written as:

$$\Delta_N(j\omega) = H(j\omega) - \hat{H}_N(j\omega) \tag{4.38}$$

where $\hat{H}_n(j\omega)$ is the reconstructed transfer function $H(j\omega)$ using the Nth-order generalized Hilbert transformation (GHT). The denominator terms in (4.36) attenuate the frequency response. The generalized Hilbert transformation is numerically more stable than the original transformation, because the integrand in (4.36) contains the polynomial denominator terms that significantly attenuate $H(j\omega)$, further reducing the impact of high-frequency truncation.

Finally, the Lagrange interpolation polynomial, just like any other polynomial interpolation scheme, suffers from a polynomial oscillation between interpolation points, known as Runge's phenomenon [21]. This oscillation increases the truncation error of the GHT, but it can be reduced by using Chebyshev nodes [19]. For the frequency range of $[-\omega_{\max}(1-\varepsilon), \omega_{\max}(1-\varepsilon)]$, the Chebyshev nodes are given by:

$$\omega_q = -\omega_{\max}(1-\varepsilon)\cos\left(\frac{q-1}{n-1}\pi\right),\ q = 1,...,N. \tag{4.39}$$

As demonstrated in the following section, the truncation error in (4.35) generally reduces as the order of Lagrange interpolation increases.

4.4.2 Truncation Error Bound

Although the generalized dispersion relations help to reduce the sensitivity to frequency truncation, (4.38) is still subject to both frequency truncation and discretization errors. It is critical that we have an upper bound error estimation of these errors to avoid false causality violations. This section reviews the error bound analysis described in Triverio and Grivet-Talocia's paper [19], and discusses the recent accuracy and speed enhancements shown in Asgari, Lalgudi, and Tsuk's work [20].

Applying an integration in (4.36) using the existing data over the frequency set Ω, we have:

$$\hat{H}_N(j\omega) = L_H(j\omega) + \frac{\prod_{q=1}^{N}(\omega-\omega_q)}{j\pi}.$$

$$\left[P.V.\int_{\Omega}\frac{H(j\omega') - L_H(j\omega')}{\prod_{q=1}^{N}(\omega' - \omega_q)}\frac{d\omega'}{\omega - \omega'} + \int_{\Omega^C}\frac{-L_H(j\omega')}{\prod_{q=1}^{N}(\omega' - \omega_q)}\frac{d\omega'}{\omega - \omega'} \right] \tag{4.40}$$

where Ω^C is the complement set that covers the frequency range not covered by data. Comparing the preceding equation with (4.36), the truncation error $T_N(j\omega)$ is given by:

$$T_N(j\omega) = \frac{\prod_{q=1}^{N}(\omega - \omega_q)}{j\pi}\int_{\Omega^C}\frac{H(j\omega')}{\prod_{q=1}^{N}(\omega' - \omega_q)}\frac{d\omega'}{\omega-\omega'}. \tag{4.41}$$

A tight bound of this error term is rigorously derived in Triverio and Grivet-Talocia's paper [19]. Assuming that:

$$|H(j\omega)| \le M\left|\omega^{\alpha}\right| \quad \text{for } \omega \in \Omega^{C} \text{ and } \alpha = 0, 1, 2,... \tag{4.42a}$$

The bound for the truncation error is given by:

$$|T_N(j\omega)| \le \frac{M}{\pi}\sum_{q=1}^{N}\left\{ (\omega_q)^{\alpha}\left[\left| \ln\left(\frac{\omega_{\max} - \omega_q}{\omega_{\max} - \omega}\right)\right| \right]\right.$$

(*continues*)

$$-(-1)^{\alpha+N}\left|\ln\left(\frac{\omega_{\max}+\omega_q}{\omega_{\max}+\omega}\right)\right| \times \prod_{\substack{p=1\\p\neq q}}^{N}\frac{\left|(\omega-\omega_p)\right|}{(\omega_q-\omega_p)}\Bigg\}. \tag{4.42b}$$

In Section 4.3, the S-parameter was recommended for checking the passivity condition. The transfer terms and far-end crosstalk terms of S-parameters have low-pass filter characteristics, making the impact of frequency truncation small. On the other hand, the reflection terms and near-end crosstalk terms of S-parameters have high-pass filter characteristics with α being zero. Figure 4.12 shows the truncation errors for a few different sampling points for a case where $M = 1$ and $\alpha = 0$.

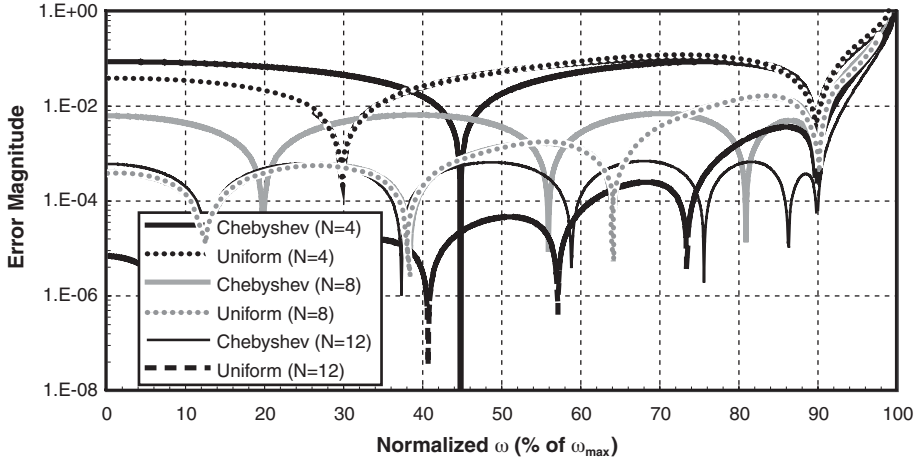


Figure 4.12 Truncation Error Bound with Various Numbers of Interpolation Nodes

As Figure 4.12 illustrates, the uniform distributions result in larger errors near the end for all cases. On the other hand, the Chebyshev distribution produced evenly spaced errors between the nodes, and the maximum errors are smaller than the uniform case. In all cases, the truncation error is largest at the end that is beyond the interpolation interval. Figure 4.12 also demonstrates the fact that as the interpolation order increases, the truncation error is reduced in both the Chebyshev and uniform cases.

For typical applications, the order of the Lagrange interpolation polynomial should be at least eight to minimize the truncation error. In fact, one can arbitrarily choose to make the order of the interpolation large, so that the maximum error bound, due to the truncation, is smaller than the discretization error associated with numerical integration. The upper bound parameters, M and α, can be estimated based on existing data (see Section 4.4.4).

4.4.3 Discretization Error Bound

Because one must calculate the Hilbert transformation of tabulated data through numerical quadrature, it inherently suffers from discretization error, in addition to the truncation error discussed in the previous section. Now, the causality equation (4.38) can be rewritten as:

$$\tilde{\Delta}_N(j\omega) = H(j\omega) - \hat{\tilde{H}}_N(j\omega) = \Delta_N(j\omega) + T_N(j\omega) + D_N(j\omega) \tag{4.43}$$

where $\tilde{\Delta}_N(j\omega)$ is the numerically calculated reconstruction error, and $D_N(j\omega)$ is the discretization error. Now, a tabulated $H_N(j\omega)$ is causal only if the following condition is met:

$$\left|\tilde{\Delta}_N(j\omega_k)\right| \leq |T_N(j\omega_k)| + |D_N(j\omega_k)| \qquad \forall k. \tag{4.44}$$

This condition guarantees the causality of the data, but it can lead to false violations, if the bounds for $T_N(j\omega)$ and $D_N(j\omega)$ are not estimated conservatively. On the other hand, the resolution (or sensitivity) of the preceding test depends on the amplitude of the numerical error bounds. The resolution of the causality check is improved if $T_N(j\omega)$ and $D_N(j\omega)$ are small. A very tight bound was given for $T_N(j\omega)$ in the previous section. After obtaining a good bound estimation for $D_N(j\omega)$, you the causality check can be accurately performed. In addition to a good bound estimation, we need to minimize the discretization or numerical integration error.

Both regular and generalized Hilbert transformations contain a singular kernel. This singularity can be analytically extracted and integrated to reduce numerical quadrature errors [19]. The bound for the integration error can be estimated using two distinct quadrature rules with different accuracy levels. Alternatively, one can use a more strict error bound, based on the worst-case bound for the given quadrature scheme to eliminate any false violations [19].

A simple, but perhaps a numerically more robust approach, is proposed by Asgari, Lalgudi, and Tsuk [20]. The remaining part of this section reviews this approach. Because the integrals containing $L_H(j\omega)$ in (4.40) cancel, it is rewritten as:

$$\hat{H}_N(j\omega) = L_H(j\omega) + \frac{\prod_{q=1}^{N}(\omega - \omega_q)}{j\pi} \times P.V. \int_{\Omega} \frac{H(j\omega')}{\prod_{q=1}^{N}(\omega' - \omega_q)} \frac{d\omega'}{\omega - \omega'}. \tag{4.45}$$

To perform the preceding integral in closed form, $H_N(j\omega)$ is interpolated by a spline-based interpolating function, as follows:

$$\overline{H}_N(j\omega) = \sum_{k=-N_\omega}^{N_\omega - 1} \sum_{l=0}^{L} \alpha_{k,l}(\omega - \omega_k)^l \cdot 1_{\Omega_k}(\omega) = \sum_{k=-N_\omega}^{N_\omega - 1} \overline{H}_{(k)}(j\omega) \cdot 1_{\Omega_k}(\omega) \tag{4.46}$$

where N_ω is the number of non-negative frequency points, L is the maximum order of the interpolation polynomial, Ω_k represents the internal $[\omega_k, \omega_{k+1}]$, $1_{\Omega_k}(\omega)$ is one for $\omega \in \Omega_k$ and zero elsewhere, and $\alpha_{k,l}$ is the spline coefficient for the kth internal and the lth power of ω.

Substituting the preceding approximation for (4.45), and performing partial fraction and analytical integration, we have:

$$\hat{\overline{H}}_N(j\omega) = L_{\overline{H}}(j\omega) + \frac{1}{j\pi}\sum_{k=-N_\omega}^{N_\omega-1}\left[\overline{H}_{(k)}(j\omega)\ln\left|\frac{\omega_k - \omega}{\omega_{k+1} - \omega}\right|\right.$$

$$\left. - \sum_{q=1}^{N}\overline{H}_{(k)}(j\omega_q)\ln\left|\frac{\omega_k - \omega_q}{\omega_{k+1} - \omega_q}\right|\prod_{p=1;\, p\neq q}^{N}\frac{\omega - \omega_p}{\omega_q - \omega_p}\right]. \tag{4.47}$$

Now, the causality condition (4.45) can be rewritten as:

$$\overline{\Delta}_N(j\omega) = H(j\omega) - \hat{\overline{H}}_N(j\omega) = \Delta_N(j\omega) + T_N(j\omega) + D_N(j\omega) + I_N(j\omega) \tag{4.48}$$

where $\overline{\Delta}_N(j\omega)$ is the numerically calculated reconstruction error, $D_N(j\omega)$ is the discretization error, and $I_N(j\omega)$ is the spline-based interpolation error. Similar to (4.44), a tabulated $H_N(j\omega)$ is causal only if the following condition is met:

$$|\overline{\Delta}_N(j\omega_k)| \leq |T_N(j\omega_k)| + |D_N(j\omega_k) + I_N(j\omega_k)| \qquad \forall k. \tag{4.49}$$

As mentioned previously, the resolution and sensitivity of the causality test can be improved by using accurate error bounds for $|D_N(j\omega_k) + I_N(j\omega_k)|$. (Various ways to estimate the error bound of $|D_N(j\omega_k) + I_N(j\omega_k)|$ are discussed in [20].) The first, and most rigorous approach, is to calculate the error bound based on the error analysis of the spline interpolation. This bound guarantees no false violations. Other approaches are based on the numerical estimation of two different interpolation schemes. For instance, one could perform the first estimation of $\hat{\overline{H}}_N(j\omega)$ using the cubic spline, and perform the second estimation using another interpolation scheme. According to Asgari, Lalgudi, and Tsuk [20], a linear interpolation resulted in a highly conservative bound, while a quadratic interpolation produced an inaccurate bound. The same third-order interpolation, based on a cubic Hermite spline (CHS), produced a reasonable value.

4.4.4 Lossless Coupled Transmission Line Example

In this section, coupled lossless transmission lines are considered to test the algorithm described in the previous section. An AC analysis of the microstrip structure (shown in Figure 4.13) is performed to generate a four-port S-parameter. The characteristic impedance is approximately 50Ω.

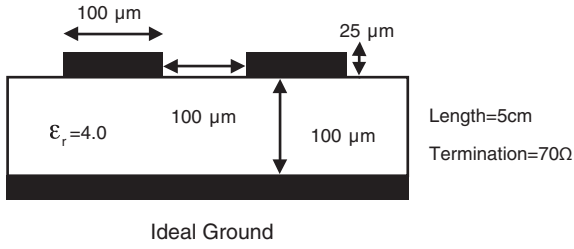


Figure 4.13 Lossless Coupled Microstrip Example

The following setup is used to check the causality: The 20th order Lagrange interpolation, with the Chebyshev node distribution described in (4.39), with $\varepsilon = 0.01$. $\widehat{H}_N(j\omega)$ is estimated using the cubic spline. The truncation and numerical interpolation errors are estimated using the cubic Hermite spline, as recommended in Asgari, Lalgudi, and Tsuk's paper [20]. To make a system response non-causal, we alter the S-parameter of the coupled line system by adding Gaussian noise with an amplitude of 0.01, a 0.5-GHz center frequency, and a 50-MHz sigma. (Noise is added to only the imaginary term to make the resulting response non-causal.) To determine the bound of the truncation error, we use $M = 0.5$ and $\alpha = 0$ in (4.42a). $\alpha = 0$ works for most of the S-parameter data. However, determining a reasonably accurate M value can be quite challenging for arbitrary data, and may require trial and error to achieve a fair estimation.

Figure 4.14(a) shows the calculated error, due to reconstruction through the generalized dispersion relationships for S11 data. This error term contains both of the errors due to numerical and causality violation. To test causality violation, the bound of the numerical error is plotted also. The original data is non-causal if it exceeds the numerical error bound. As shown in the figure, there is a causality violation, due to the added Gaussian noise, around the 0.5-GHz center frequency. The numerical error is decomposed into the truncation error in numerical integration, and the error due to interpolation and discretization, as shown in Figure 4.14(b). Figure 4.15, Figure 4.16, and Figure 4.17 show the analysis repeated for S12, S13, and S14, respectively.

4.4.5 Causality-Enforced Interpolation

Interpolation of S-parameter data is often necessary, especially for DC points. However, a simple numerical interpolation to real and imaginary parts could violate a causality condition. Consequently, the generalized dispersion relations (described previously) can be used to interpolate S-parameter points in Triverio and Grivet-Talocia's paper [12]. First, the imaginary part is interpolated using a standard numerical interpolation scheme. Then, the real part is derived through the generalized dispersion relations. Significant improvement is seen when using this scheme, as compared to a conventional interpolation, or to using dispersion relations without subtractions. In fact, applying dispersion relations without subtractions resulted in the worst-case approximation.

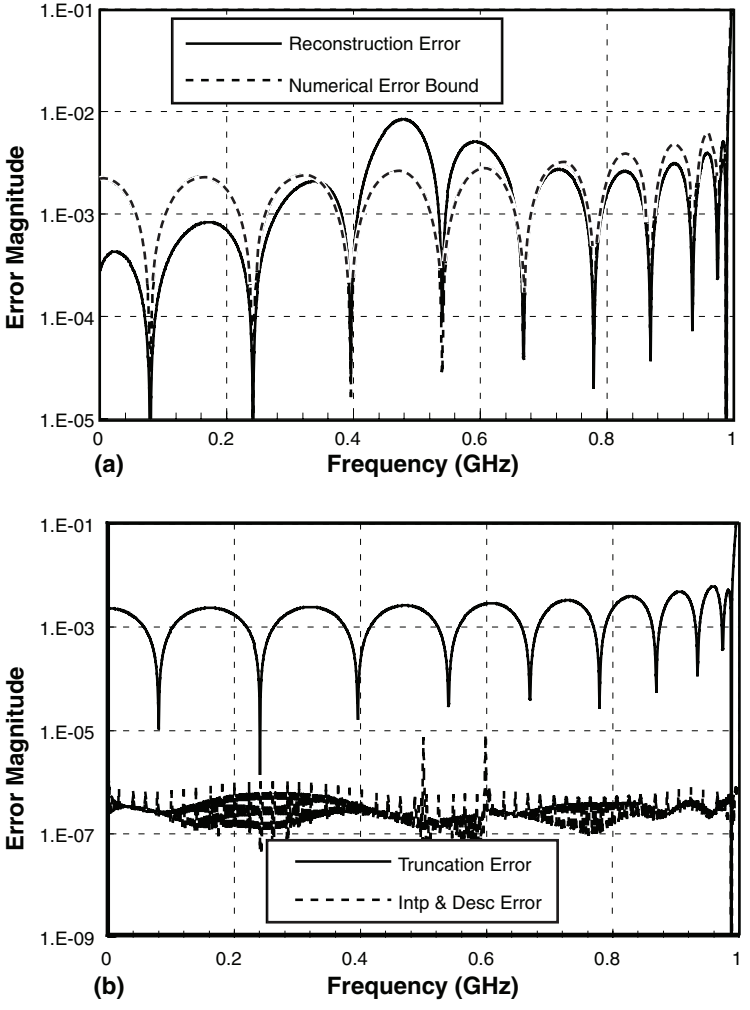


Figure 4.14 (a) Causality-Checking Plot with Numerical Error Bound, and (b) Truncation Error and Numerical Interpolation and Discretization Error for S11

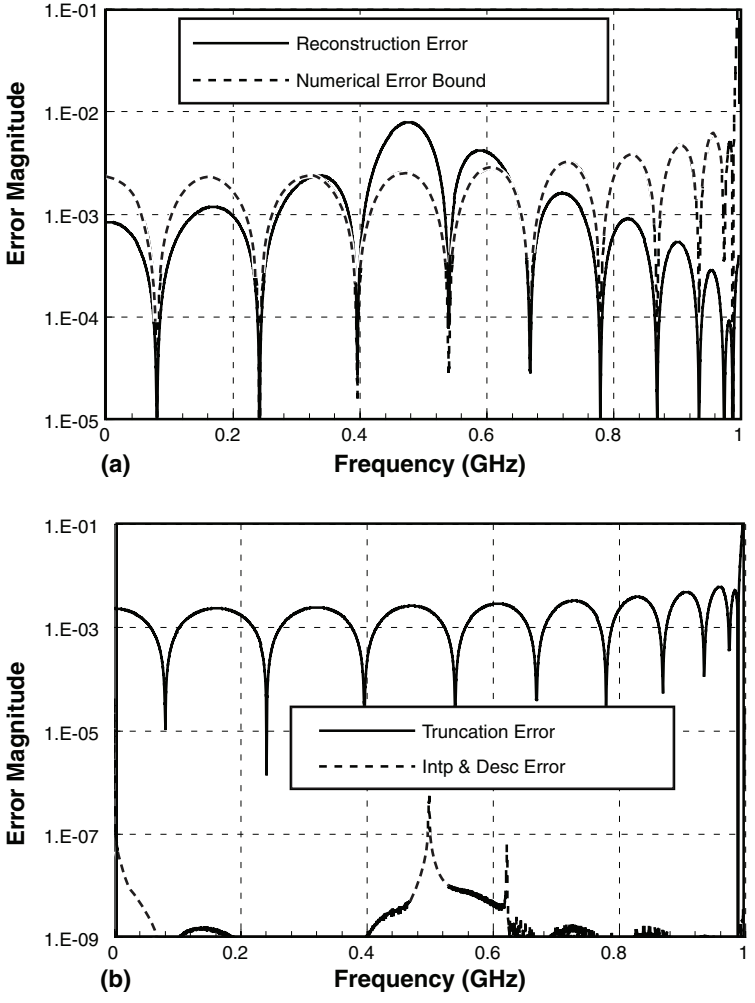


Figure 4.15 (a) Causality-Checking Plot with Numerical Error Bound, and (b) Truncation Error and Numerical Interpolation and Discretization Error for S12

4.5 Summary

This chapter reviewed the conversion formulae for various network parameters, including the conversion between single-ended and mixed-mode parameters. Modeling frequency domain data for time-domain simulation requires special attention, in order to guarantee an accurate channel response for broadband digital signals. This chapter described a technique to properly determine

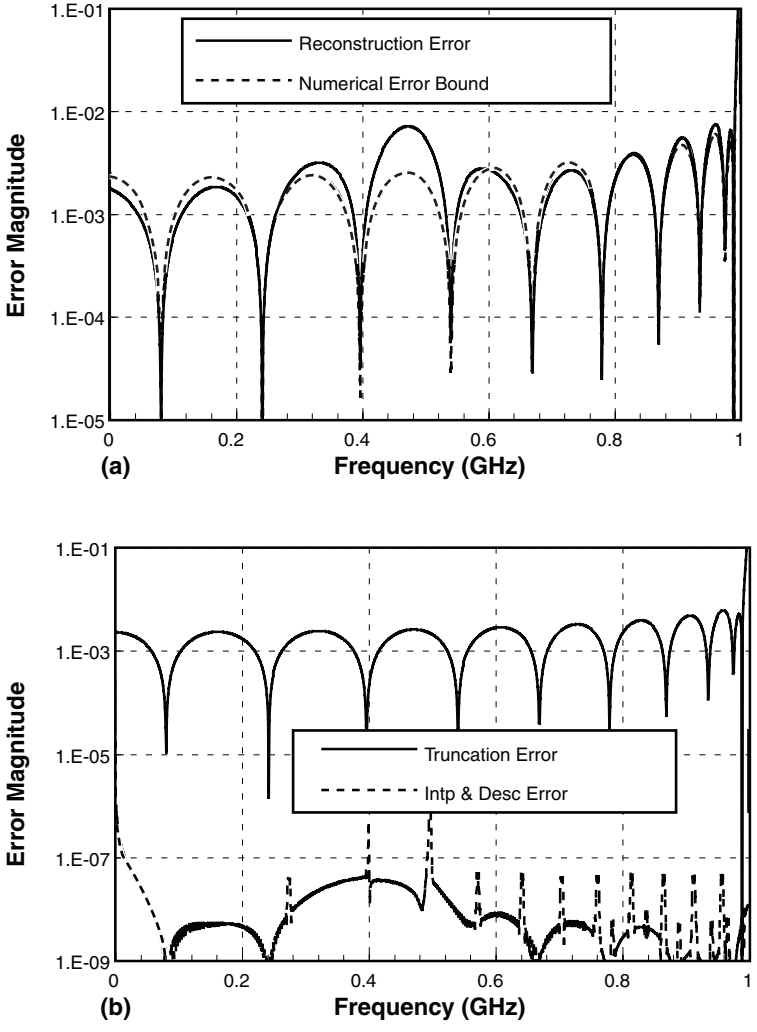


Figure 4.16 (a) Causality-Checking Plot with Numerical Error Bound, and (b) Truncation Error and Numerical Interpolation and Discretization Error for S13

the maximum frequency range, along with other tips to improve numerical stability. The passivity and causality conditions are crucial to a reliable time-domain simulation. The chapter also presented an intuitive and informal derivation for passivity and causality conditions, along with potential numerical issues.

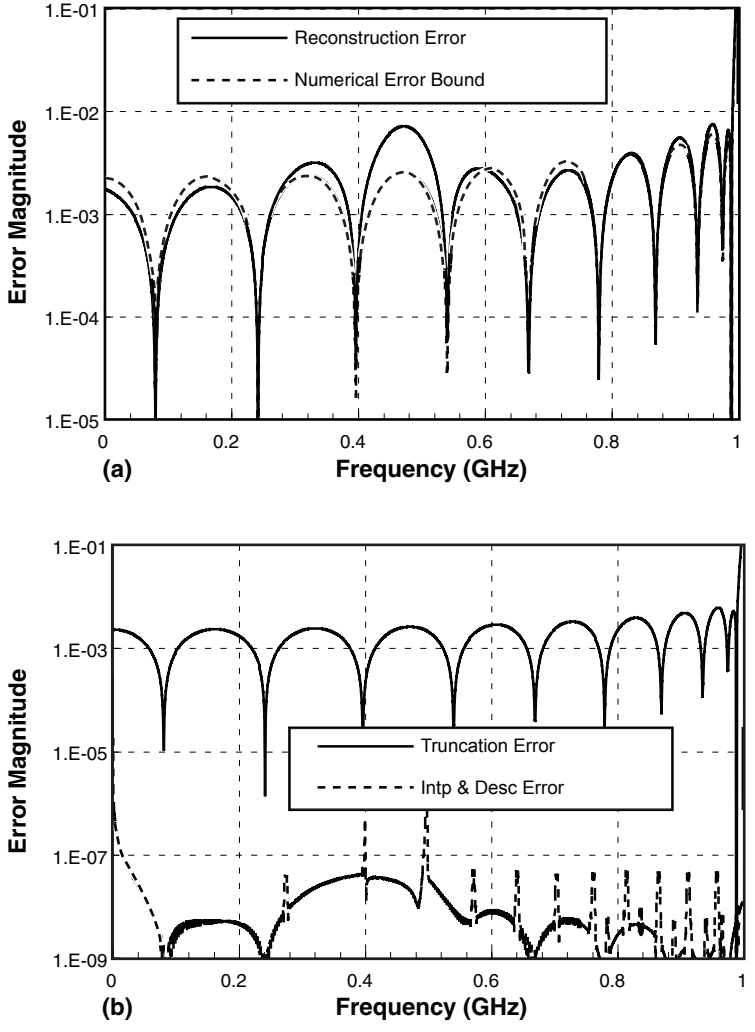


Figure 4.17 (a) Causality-Checking Plot with Numerical Error Bound, and (b) Truncation Error and Numerical Interpolation and Discretization Error for S14

References

1. T. Edwards, *Foundations for Microstrip Circuit Design*, 2nd ed., Wiley, ap. C, pp. 392–395, 1981.
2. K. Kurokawa, "Power waves and the scattering matrix," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-13, no. 3, pp. 194–202, Mar. 1965.
3. R. Marks and D. Williams, "A general waveguide circuit theory," *Journal of Research of the National Institute of Standards and Technology*, vol. 97, pp. 533–561, Sep. 1992.
4. D. Oh, F. Lambrecht, S. Chang, Q. Lin, J. Ren, J. Zerbe, C. Yuan, C. Madden, and V. Stojanovic, "Accurate method for analyzing high-speed I/O system performance," presented at the IEC DesignCon, Santa Clara, CA, 2007.
5. D. Bockelman and W. R. Einsenstadt, "Combined differential and common-mode scattering parameters: theory and simulation," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-43, no. 7, pp. 1530–1539, Jul. 1995.
6. H. Johnson and M. Graham, *High-Speed Digital Design, A Handbook of Black Magic*, Prentice Hall, pp. 1–5, 1993.
7. R. Schaefer, "Discussing the limitations and accuracies of time and frequency domain analysis of physical layer devices," presented at the IEC DesignCon, Santa Clara, CA, 2005.
8. D. Smolyansky, "Advances in gigabit channel measurement-based characterization and simulation," presented at the IEC DesignCon, Santa Clara, CA, 2007.
9. F. Rao, C. Morgan, S. Gupta, and V. Borich, "The need for impulse response models and an accurate method for impulse generation from band-limited S-parameters," presented at the IEC DesignCon, Santa Clara, CA, 2008.
10. W. T. Beyene and C. Yuan, "An accurate transient analysis of high-speed package interconnects using convolution technique," *International Journal of Analog Integrated Circuits and Signal Processing*, vol. 35, no. 2-3, pp. 107–120, 2003.
11. P. Triverio and S. Grivet-Talocia, "Stability, causality, and passivity in electrical interconnect models," *IEEE Transactions on Advanced Packaging*, vol. 30, no. 4, pp. 795–808, Nov. 2007.
12. P. Triverio and S. Grivet-Talocia, "Causality-constrained interpolation of tabulated frequency responses," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 181–184.
13. H. Shi, "A refine procedure for S-parameter DC extrapolation based on sampling theorem and causality," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2007, pp. 43–46.

14. E. Gad, C. Chen, M. Nakhla, and R. Achar, "Passivity verification in delay-based macromodels of electrical interconnects," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 10, pp. 2173–2187, Oct. 2005.
15. S. Grivet-Talocia and A. Ubolli, "A comparative study of passivity enforcement schemes for linear lumped macromodels," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 4, pp. 673–683, Nov. 2008.
16. M. R. Wohlers, *Lumped and Distributed Passive Networks*, Academic, 1969.
17. D. C. Youla, L. J. Castriota, and H. J. Carlin, "Bounded real scattering matrices and the foundations of linear passive network theory," *IRE Transactions on Circuit Theory*, vol. CT-6, pp. 102–124, Mar. 1959.
18. B. Young and A. S. Bhandal, "Causality checking and enhancement of 3D electromagnetic simulation data," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2010, pp. 81–84.
19. P. Triverio and S. Grivet-Talocia, "Robust causality characterization via generalized dispersion relations," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 3, pp. 579–593, Aug. 2008.
20. S. Asgari, S. N. Lalgudi, and M. Tsuk, "Analytical integration-based causality checking of tabulated S-parameters," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2010, pp. 189–192.
21. W. Rudin, *Principles of Mathematical Analysis*, 3rd ed., McGraw Hill, 1976.

CHAPTER 5

Transmission Lines

Dan Oh and Joong-Ho Kim

Chapter 4, "Network Parameters," discusses general network parameters that can represent arbitrary passive structures. These network parameters provide accurate channel models for frequency domain analysis, but they can lead to numerically unstable or erroneous time-domain models. For uniform interconnect structures, a transmission line model provides a more physically intuitive and accurate circuit model than general network models. A thorough understanding of transmission line models helps SI engineers develop accurate channel models and leads to better physical channel designs. The first quarter of this chapter covers the basic theory of the telegrapher's equations and transmission line parameters. It also presents the formula used to convert transmission line parameters to other network parameters.

The second quarter of this chapter reviews a popular time-domain simulation technique, based on the recursive convolution for a lossy transmission line model. This method is based on the frequency-domain rational function approximation. It provides a fast and accurate time-domain model when compared to the traditional direct convolution method. However, the accuracy of the recursive convolution strongly depends on the rational function approximation, which is not always stable. The chapter explores the potential numerical instabilities associated with the recursive convolution approach.

The third quarter of this chapter discusses the modeling of transmission lines, based on measurement data. Transmission line models are commonly generated using electromagnetic field solvers. The accuracy of field solver models is limited by the accuracy of the input material parameters. However, material properties in high-frequency regions, such as conductor conductance and dielectric loss tangent, are hard to measure or characterize. Consequently, the direct measurement-based transmission line model can be quite useful, and even the effective material property can be calculated from transmission line models.

The last quarter of this chapter examines an on-chip interconnect model. A high-speed I/O interface uses a high-frequency clock, which is routed over a relatively long distance, because the I/O interface block often takes a large portion of the die peripheral. A simple RC network may not be sufficient to model clock nets for I/O interfaces, but a transmission line model can provide more accuracy. However, on-chip wires are quite different in nature than off-chip interconnects, because they often do not have clear return paths. The summary section provides a comparison of the various RLGC parameters of motherboard traces, package traces, and on-chip interconnects, and points out the key differences.

5.1 Transmission Line Theory

Maxwell's equations for the transverse electromagnetic (TEM) waves on multi-conductor transmission lines are reduced to the telegrapher's equations. The general form of the telegrapher's equations, in the frequency domain, is given by:

$$-\frac{\partial \vec{v}(z,\omega)}{\partial z} = [\mathbf{R}(\omega) + j\omega\mathbf{L}(\omega)]\vec{i}(z,\omega) = \mathbf{Z}_l(\omega)\vec{i}(z,\omega) \tag{5.1a}$$

$$-\frac{\partial \vec{i}(z,\omega)}{\partial z} = [\mathbf{G}(\omega) + j\omega\mathbf{C}(\omega)]\vec{v}(z,\omega) = \mathbf{Y}_l(\omega)\vec{v}(z,\omega) \tag{5.1b}$$

where $\vec{v}(z)$ is the voltage vector across the lines, and $\vec{i}(z)$ is the current vector along the lines. R, L, G, and C are the resistance (Ω/m), inductance (H/m), conductance (S/m), and capacitance (F/m) matrices per unit length; generally, they are functions of frequency.

The general solution to the preceding coupled differential equations is as follows [1]:

$$\vec{v}(\omega) = e^{-\boldsymbol{\Psi}(\omega)z}\mathbf{A} + e^{\boldsymbol{\Psi}(\omega)z}\mathbf{B} \tag{5.2a}$$

$$\mathbf{Z}_c(\omega)\vec{i}(\omega) = e^{-\boldsymbol{\Psi}(\omega)z}\mathbf{A} - e^{\boldsymbol{\Psi}(\omega)z}(\omega)\mathbf{B} \tag{5.2b}$$

where

$$\mathbf{Z}_c(\omega) = \mathbf{Y}_c^{-1}(\omega) = \boldsymbol{\Psi}(\omega)\big(\mathbf{G}(\omega) + j\omega\mathbf{C}(\omega)\big)^{-1} = \boldsymbol{\Psi}(\omega)\mathbf{Y}_l(\omega)^{-1} \tag{5.3a}$$

$$\boldsymbol{\Psi}(\omega) = \big[\,\big(\mathbf{R}(\omega) + j\omega\mathbf{L}(\omega)\big)\big(\mathbf{G}(\omega) + j\omega\mathbf{C}(\omega)\big)\,\big]^{1/2} = [\mathbf{Z}_l(\omega)\mathbf{Y}_l(\omega)]^{1/2}. \tag{5.3b}$$

$\mathbf{Y}_c$ and $\mathbf{Z}_c$ are the characteristic impedance and admittance matrices, respectively. $\boldsymbol{\Psi}$ is the propagation constant matrix. Equations (5.2a) and (5.2b) assume that $\mathbf{Z}_l$ and $\mathbf{Y}_l$ are symmetric, and $\mathbf{Z}_l\mathbf{Y}_l$ are diagonalizable, as follows:

$$\mathbf{Z}_l\mathbf{Y}_l = \mathbf{M}\hat{\mathbf{D}}\mathbf{M}^{-1} = \mathbf{M}\hat{\boldsymbol{\Psi}}^2\mathbf{M}^{-1} = \mathbf{M}\begin{bmatrix}\psi_1^2 & & \\ & \ddots & \\ & & \psi_N^2\end{bmatrix}\mathbf{M}^{-1} \tag{5.4}$$

where $\hat{\boldsymbol{\Psi}}$ is the modal propagation matrix, and ψ_i is the modal propagation constant. Note that $\mathbf{Y}_l\mathbf{Z}_l$ and $\mathbf{Z}_l\mathbf{Y}_l$ have the same eigenvalues, ensuring that the current waves propagate in the same way as the voltage waves, even though the eigenvectors are different. The modal propagation constant plays an important role in exploring and understanding various phenomena in transmission line analysis. One such example is the impact of crosstalk, discussed in Section 5.2.

The conversion from the transmission parameters ($\mathbf{Z}_c$ and $\boldsymbol{\Psi}$) to other network parameters is shown in Table 5.1 [1] [2]. When we apply functions to matrix $\mathbf{A}$, the following definition is used:

$$\text{fn}(\mathbf{A}) \equiv \mathbf{M}\text{fn}(\hat{\mathbf{A}})\mathbf{M}^{-1} = \mathbf{M}\begin{bmatrix}fn(a_1) & & 0 \\ & \ddots & \\ 0 & & fn(a_N)\end{bmatrix}\mathbf{M}^{-1} \tag{5.5}$$

where a_i is the element of the diagonalized matrix $\hat{\mathbf{A}}$.

After $\mathbf{Z}_c$ and $\boldsymbol{\Psi}$ are obtained, RLGC matrices are obtained using the following formulae:

$$\mathbf{R}(\omega) = \text{Re}[\boldsymbol{\Psi}(\omega)\mathbf{Z}_c(\omega)], \quad \mathbf{L}(\omega) = \frac{1}{\omega}\text{Im}[\boldsymbol{\Psi}(\omega)\mathbf{Z}_c(\omega)] \tag{5.6a}$$

$$\mathbf{G}(\omega) = \text{Re}[\mathbf{Z}_c^{-1}(\omega)\boldsymbol{\Psi}(\omega)], \quad \mathbf{C}(\omega) = \frac{1}{\omega}\text{Im}[\mathbf{Z}_c^{-1}(\omega)\boldsymbol{\Psi}(\omega)]. \tag{5.6b}$$

Table 5.1 Conversion Formulae Between the Transmission Line Parameter Matrices and **Z** and **ABCD** Matrix Parameters

	$\mathbf{Z}_c$, $\boldsymbol{\Psi}$	**Z**	**ABCD**
$\mathbf{Z}_c$, $\boldsymbol{\Psi}$	$\mathbf{Z}_c(\omega) = \boldsymbol{\Psi}(\omega)\mathbf{Y}_l^{-1}(\omega)$ $\boldsymbol{\Psi}(\omega) = [\mathbf{Z}_l(\omega)\mathbf{Y}_l(\omega)]^{1/2}$ $\mathbf{Z}_c(\omega) = \mathbf{Y}_c^{-1}(\omega)$ $\mathbf{Y}_l(\omega) = \mathbf{G}(\omega) + j\omega\mathbf{C}(\omega)$ $\mathbf{Z}_l(\omega) = \mathbf{R}(\omega) + j\omega\mathbf{L}(\omega)$	$\mathbf{Z}_{11} = \mathbf{Z}_c\coth(\boldsymbol{\Psi}l)$ $\mathbf{Z}_{12} = \mathbf{Z}_c\,\text{csch}(\boldsymbol{\Psi}l)$ $\mathbf{Z}_{11} = \mathbf{Z}_{22}, \mathbf{Z}_{12} = \mathbf{Z}_{21}$	$\mathbf{A} = \mathbf{Z}_c\cosh(\boldsymbol{\Psi}l)\mathbf{Z}_c^{-1}$ $\mathbf{B} = \mathbf{Z}_c\sinh(\boldsymbol{\Psi}l)$ $\mathbf{C} = \sinh(\boldsymbol{\Psi}l)\mathbf{Z}_c^{-1}$ $\mathbf{D} = \cosh(\boldsymbol{\Psi}l)$
Z	$\cosh(\boldsymbol{\Psi}l) = \mathbf{Z}_{12}^{-1}\mathbf{Z}_{11}$ $\mathbf{Z}_c = \mathbf{Z}_{12}\sinh(\boldsymbol{\Psi}l)$	$\begin{bmatrix}\vec{v}_1\\ \vec{v}_2\end{bmatrix} = \begin{bmatrix}\mathbf{Z}_{11} & \mathbf{Z}_{12}\\ \mathbf{Z}_{21} & \mathbf{Z}_{22}\end{bmatrix}\begin{bmatrix}\vec{i}_1\\ \vec{i}_2\end{bmatrix}$	See Table 4.1
ABCD	$\cosh(\boldsymbol{\Psi}l) = \mathbf{D}$ $\mathbf{Z}_c = \mathbf{B}\,\text{csch}(\boldsymbol{\Psi}l)$	See Table 4.1	$\begin{bmatrix}\vec{v}_1\\ \vec{i}_1\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\vec{v}_2\\ -\vec{i}_2\end{bmatrix}$

5.1.1 Quasi-Static Approximation

One of the underlying assumptions in the previous transmission line equations is that the propagation wave is a TEM mode. In general, transmission lines in an inhomogeneous dielectric medium, or one containing conductor loss, do not support TEM modes. Full-wave analysis (Maxwell equations) is required to accurately characterize hybrid modes in transmission lines. However, when the transverse components of the electric and magnetic fields are predominant over the longitudinal components, the fundamental hybrid mode becomes a quasi-TEM mode, in which TEM properties dominate the hybrid modes. The valid range of the quasi-TEM mode can be determined using dimensional analysis on the Maxwell equations [3–5]. In essence, a quasi-TEM model is valid when the transverse dimensions are much smaller than the wavelength. For integrated circuit applications, this means that the quasi-TEM model is sufficient, assuming that the conductor loss can be neglected.

The conductor loss due to the skin effect or surface roughness causes an electric field in the direction of the current flow, which is ignored in a quasi-TEM mode. Fortunately, study shows that the quasi-TEM approximation is also valid at the frequency range where the skin effect is predominant [6]. On the other hand, dielectric loss does not contribute any additional electric field in the current flow direction, and does not impact the quasi-TEM approximation as much as the conductor loss. Finally, because the electric and magnetic field distributions of a TEM mode are close to those of a static case, one can use static analysis to characterize quasi-TEM lines, instead of full-wave solvers. This justifies why many transmission line solvers are quasi-static tools.

5.1.2 Properties of RLGC Matrices

Typical transmission-line models use RLGC matrices as an input to specify the transmission line parameters. RLGC models have evolved from constant matrices into fully frequency-tabularized matrices. When digital I/O speed is slow, the frequency-varying nature of skin effect and dielectric losses can be ignored; constant RLGC matrices are sufficient for transmission line simulation. As I/O speed increases, the dispersion due the frequency-dependent losses can no longer be ignored, so R and G values are expressed using equations to model frequency dependency. The following expressions are widely used for modeling the skin effect and dielectric loss:

$$\mathbf{R}(\omega) = \mathbf{R}_{dc} + \mathrm{j}\sqrt{f}\mathbf{R}_s \tag{5.7}$$

$$\mathbf{G}(\omega) = \mathbf{G}_{dc} + \mathrm{j}f\mathbf{G}_d. \tag{5.8}$$

The preceding expressions work well when the loss is still relatively small. However, when frequency dependent loss is large, these simple expressions do not accurately capture transmission line behavior. Furthermore, these equations, in fact, violate the causality and cause additional accuracy issues. HSPICE's W-element model supports a tabular format that allows arbitrary frequency dependency. With this tabular format, S-parameter data (from either a full-wave solver or a measurement) can be used to accurately model transmission lines.

Section 5.4 describes the detailed methodology for extracting the transmission line model from measurement.

The rest of this section is devoted to the review of the physical properties of **L** and **C** matrices. Instead of simply listing the properties, this section presents brief derivations based on static assumption. These properties are also discussed in the context of telegrapher's equations. First consider the capacitance matrix. The capacitance matrix (**C**) relates the total free charges on the conductors to the voltages in the following manner:

$$\mathbf{C}\vec{v} = \vec{q}. \tag{5.9}$$

The capacitance matrix is calculated by obtaining the charges for N independent voltage excitations. The following matrix form describes the solution:

$$\mathbf{C} = \mathbf{Q}\mathbf{V}^{-1}. \tag{5.10}$$

By alternately exciting a unit voltage on one conductor with respect to the other conductors and the ground conductor, **V** becomes an identity matrix and **C** is simply equal to **Q**. Now, **Q** represents the physical charges on the conductors and the following properties can be deduced:

- Because **Q** is symmetric, due to reciprocity, **C** is also symmetric:

$$C_{i,j} = C_{j,i} \tag{5.11a}$$

- Because the diagonal elements of **Q** are the charges on the excited conductors, and the off-diagonal elements of **Q** are the induced charges on the resting conductors, the diagonal elements of **Q** are positive, and all the off-diagonal elements are negative:

$$C_{i,i} > 0 \text{ and } C_{i,j} < 0 \text{ for j} \neq \text{i} \tag{5.11b}$$

- Furthermore, the magnitude of the sum of the induced charges must be smaller than the charges on the excited conductor, because some charges are induced on the ground conductor. This makes **C** diagonally dominant:

$$\left|C_{i,i}\right| > \left|\sum_{i \neq j} C_{i,j}\right| \tag{5.11c}$$

- Because C is real, symmetric, and diagonally dominant, it is positive definite.

The preceding capacitance matrix is called the Maxwellian capacitance. The physical self- and mutual-capacitances (C_i^s and $C_{i,j}^m$) are calculated using the following relationships:

$$C_{i,j} = -C^m_{i,j} \tag{5.12a}$$

$$C_{i,i} = C^s_i + \sum_{i \neq j} C^m_{i,j}. \tag{5.12b}$$

In contrast, the inductance matrix (**L**) relates to the magnetic flux difference between the signal conductors and the reference conductor due to the currents on the signal conductors in the following manner:

$$\mathbf{L}\vec{I} = \vec{\psi}. \tag{5.13}$$

This static inductance is referred to as external inductance. It does not include any internal inductance due to skin effect. All the elements of **L** are positive. By exploring isomorphism between electrostatic and magnetostatic formulations, the inductance matrix is calculated using the following expression [7]:

$$\mathbf{L} = \frac{1}{c^2}\mathbf{C}^{-1}_{free} \tag{5.14}$$

where c is the speed of light, and $\mathbf{C}_{free}$ is the capacitance matrix for a free-space case, where all dielectric layers are replaced with free space.

L is symmetric due to reciprocity. The inverse of a positive definite matrix is also positive definite, so **L** is positive definite based on (5.14). This Maxwellian inductance matrix **L** is related to the physical inductance as follows:

$$L_{i,j} = L^m_{i,j} \tag{5.15a}$$

$$L_{i,i} = L^s_i. \tag{5.15b}$$

In the previous section, the solutions (5.2a) and (5.2b) to the telegrapher's equations (5.1a) and (5.1b) assume that the product **ZY** is diagonalizable (5.4). For lossless transmission lines, this condition is satisfied, because it is a given that both **L** and **C** are symmetric and positive definite, and so **LC** and **CL** can be diagonalizable with generally n distinct eigenvalues for inhomogeneous media [8]. For homogeneous media, both the **L** and **C** matrices are related to the following equation, based on (5.14):

$$\mathbf{LC} = \frac{1}{v^2}. \tag{5.16}$$

Therefore, all eigenvalues of the **LC** matrix for homogeneous media are identical. This is an important property, because it is closely related to the physics behind transmission line crosstalk, which is the topic of the next section.

5.2 Forward and Backward Crosstalk

Crosstalk in transmission lines has been described in a variety of ways. In this section, *forward crosstalk* is defined as the crosstalk occurring during the wave propagation. Hence, forward crosstalk does not depend on the termination conditions. On the other hand, *backward crosstalk* is any coupling due to reflection, making it a strong function of line impedance and termination conditions. The terms *near-end crosstalk* and *far-end crosstalk* represent the sum of forward and backward crosstalk at the source and load ends of a transmission line, respectively. For crosstalk analysis, assume that the transmission line is uniform and lossless. The coupling mechanism is not a strong function of losses (except for ground conductor loss).

Traditionally, the crosstalk phenomenon is explained with capacitive and inductive coupling behaviors. This approach leads to a mathematically simple expression that describes the coupling effects for two or even three coupled transmission lines. For instance, Dally and Poulton [9] provide the amplitude of the far-end crosstalk (V_F) and the near-end crosstalk (V_N) of the symmetric two-line transmission line shown in Figure 5.1:

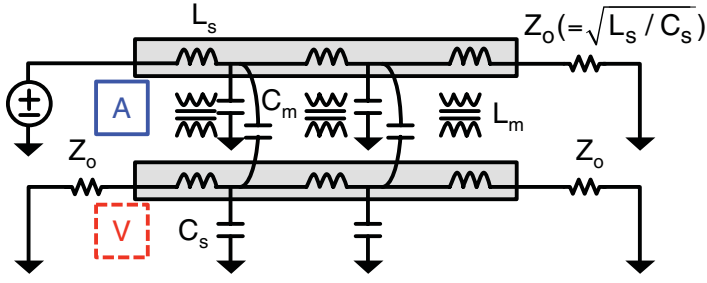


Figure 5.1 Symmetric Two-Line Transmission Line System

$$V_F = \frac{l\sqrt{L_s C_s}}{2}\left(\frac{C_m}{(C_s + C_m)} - \frac{L_m}{L_s}\right)\frac{V_s}{t_r} = \frac{t_d}{2}\left(\frac{C_m}{(C_s + C_m)} - \frac{L_m}{L_s}\right)\frac{V_s}{t_r} \tag{5.17a}$$

$$V_N = \frac{1}{4}\left(\frac{C_m}{(C_s + C_m)} + \frac{L_m}{L_s}\right)V_S \tag{5.17b}$$

where l is the length of the transmission line, and V_S is the aggressor voltage swing. Figure 5.2 shows the near-end and far-end crosstalk responses due to the rising step for a microstrip case. Note that the expression for the far-end crosstalk is only valid when the skew, between the even and odd modes, is smaller than the risetime (t_r).

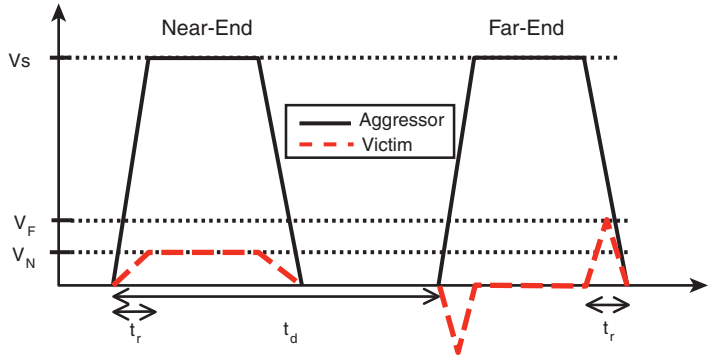


Figure 5.2 Near-End and Far-End Crosstalk Based on Inductive and Capacitive Coupling Analysis

However, the coupling phenomenon is fundamentally due to mode interaction in the electromagnetic sense. An explanation based on simple capacitance and inductance cannot explain all the physics behind the crosstalk mechanism, and it can sometimes result in misleading crosstalk effects, even for seasoned signal integrity engineers. For instance, a common misbelief is that one can eliminate coupling by sending signals in modal space. It is also widely believed that forward crosstalk is related to the derivative of the aggressor signal. The following section covers the basics of forward and backward crosstalk based on a modal analysis instead of capacitance and inductance couplings. The modal analysis will demonstrate that forward crosstalk is not really a function of the derivative of the aggressor response, and that transmitting signal in modal space does not solve crosstalk problems.

5.2.1 Mode Discontinuities and Reflection (Backward Crosstalk)

In waveguide theory, different media supports different propagation modal spaces. When two different media are connected, it creates discontinuity at the interface. The propagating modes are converted from one modal space to another, potentially causing mode coupling and reflection. Consider a case where the source is single-ended terminated and the transmission line is loosely coupled. In this case, the mode configuration (eigenvector) of the source and transmission line is all single-ended, so the modal spaces are identical and no conversion occurs. Only signal reflection occurs, depending on the impedance mismatch, but no coupling occurs. On the other hand, if the same system has a strongly coupled transmission line, the propagating signal is converted to coupled mode from single-ended mode, reflecting some of the modes due to the modal impedance mismatch at the interface. When these coupled modes have different amounts of reflection, crosstalk occurs, which is referred to as backward crosstalk. Note that the reflection of modes occurs at the interface, so backward crosstalk occurs only at the interface. After the signal crosses the interface, no additional crosstalk occurs, with the exception of potential forward crosstalk that occurs only for special media, and which the next section describes.

Mathematically, the mode reflection from the following expressions of the transmission and reflection coefficient matrices can be calculated using the following matrix equation [18]:

$$\mathbf{T} = (\mathbf{I} + \mathbf{Z}_t\mathbf{Y}_c)^{-1} \tag{5.18a}$$

$$\mathbf{\Gamma} = (\mathbf{Z}_t\mathbf{Y}_c + \mathbf{I})^{-1}(\mathbf{Z}_t\mathbf{Y}_c - \mathbf{I}) \tag{5.18b}$$

where $\mathbf{Z}_t$ is the termination matrix, and $\mathbf{Y}_c$ is the characteristic impedance of the transmission line. To avoid crosstalk, both the $\mathbf{T}$ and $\mathbf{\Gamma}$ matrices must be diagonal; otherwise, the coupling terms will result in crosstalk. Based on (5.18), $\mathbf{T}$ and $\mathbf{\Gamma}$ are diagonal, if and only if $\mathbf{Z}_t\mathbf{Y}_c$ is diagonal; that is, $\mathbf{Z}_t$ must be a simple linear multiplication of $\mathbf{Z}_c$. Consequently, the termination network also requires cross terminations between signal nodes to avoid crosstalk.

Such termination can cause significant power consumption and can require significant silicon area if the termination is on chip. On-chip termination is necessary for high-speed links, in order to avoid signal reflection. As shown from the earlier expression, the backward crosstalk is related to the mode conversion, which is associated with eigenvectors, and it is not directly related to eigenvalues. In other words, backward crosstalk is not directly related to any wave propagation characteristics. Specifically, it is independent of signal-edge rate and transmission-line length. The waveform shape of the coupled noise is identical to the source (aggressor) waveform, except for the polarity. This assumes that the eigenvectors are not a function of frequency.

In practical printed circuit board (PCB) applications, backward crosstalk occurs not only at the source or load ends, but also near trace escape areas, wirebond, via transitions, connector transitions, and package- and board-trace impedance mismatches (strictly speaking, the mode mismatch). Consequently, properly matching the trace impedance, in order to avoid coupling, is almost impossible (or at least impractical). A more effective way to handle crosstalk is to cancel it using equalization schemes [10–14]. These crosstalk cancellation techniques have been studied for SerDes applications that use differential signaling. However, so far, no commercial product has adopted any crosstalk cancellation method. The method can be more effective for memory interfaces, because they often use single-ended signaling, which is very sensitive to crosstalk. Unfortunately, high implementation costs and design complexity prohibit the use of such techniques in real applications.

Now an exact expression for the near-end crosstalk (V_N) of the symmetric two-line case shown in Figure 5.1 is derived based on the modal analysis. The resulting expression is compared with Dally and Poulton's expression (5.17b). Then, HSPICE simulation is performed to compare the accuracy of these two expressions. Based on Figure 5.1, odd and even capacitance and inductance are written as follows:

$$L_{odd} = L_s - L_m = L_{11} - L_{12} \tag{5.19a}$$

$$L_{even} = L_s + L_m = L_{11} + L_{12} \tag{5.19b}$$

$$C_{odd} = C_s + 2C_m = C_{11} - C_{12} \tag{5.19c}$$

$$C_{even} = C_s = C_{11} + C_{12} \tag{5.19d}$$

where L_s, L_m, C_s, and C_m are physical inductance and capacitance, and L_{11}, L_{12}, C_{11}, and C_{12} are Maxwellian quantities where C_{12} is negative. The characteristic impedance matrix is obtained using the following expression:

$$\mathbf{Z}_c = \begin{bmatrix} Z_s & Z_m \\ Z_m & Z_s \end{bmatrix} \tag{5.20a}$$

$$Z_s = \frac{1}{2}(Z_{odd} + Z_{even}) = \frac{1}{2}\left(\sqrt{L_{odd}/C_{odd}} + \sqrt{L_{even}/C_{even}}\right) \tag{5.20b}$$

$$Z_m = \frac{1}{2}(Z_{even} - Z_{odd}) = \frac{1}{2}\left(\sqrt{L_{even}/C_{even}} - \sqrt{L_{odd}/C_{odd}}\right). \tag{5.20c}$$

Applying this equation along with the termination, the transmitted term of the coupled voltage is

$$\mathbf{T}(2,1) = \frac{Z_m Z_o}{Z_s^2 - Z_m^2 + Z_o Z_s}. \tag{5.21}$$

The preceding expression does not reduce to (5.17b). To test the accuracy of these two equations, an HSPICE simulation is performed using the following transmission line parameters:

$$\mathbf{L} = \begin{bmatrix} 2.9350\text{E-}07 & 5.3183\text{E-}08 \\ 5.3183\text{E-}08 & 2.9350\text{E-}07 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1.1632\text{E-}10 & -1.0459\text{E-}11 \\ -1.0459\text{E-}11 & 1.1632\text{E-}10 \end{bmatrix}. \tag{5.22}$$

The HSPICE simulated near-end crosstalk was 70.06mV. Equation (5.21) results in 70.05mV, which exactly matches the HSPICE simulation. Equation (5.17b) results in 67.78mV, which clearly demonstrates that (5.17b) is only an approximation.

5.2.2 Mode Propagation (Forward Crosstalk)

The claim that there is no crosstalk during signal propagation may surprise many people. However, this is true from the modal analysis point of view. Transmission lines are just one of the special types of waveguides that support TEM (or quasi-TEM) wave propagation modes. As long as the transmission lines are uniform, these modes are independent and do not couple each other. Any propagation signal can be decomposed into a sum of transmission-line propagation modes, and then recomposed back to the original signal. Consequently, no mode conversion occurs during the propagation.

Now, the question becomes: What causes forward crosstalk? The answer is the modal velocity difference between propagation modes. In general, all N independent modes of transmission lines can propagate at different mode velocities. As the line length increases, the flight time of each mode can be significantly different. First, the original signal is decomposed into various

transmission modes, before it propagates through the transmission lines. Each mode arrives at the end of the lines at a different speed, and by the time it is composed back to the original signal, each mode has accumulated significant skew. This skew is seen as coupling noise at the far end. Therefore, forward crosstalk linearly increases as the transmission line length increases due to the accumulation of skew.

Figure 5.3 demonstrates this forward crosstalk mechanism. For the sake of simplicity, this example uses a balanced two microstrip-line system. In this case, there are odd and even modes, (0.5, 0.5) and (0.5, -0.5), respectively. Any voltage excitation is decomposed into the combination of these two modes before propagation. Figure 5.3 shows the transmission line excited with 1V at the aggressor line, and 0V at the victim line. The three plots in Figure 5.3 illustrate the voltage waveforms at three different locations, and show both the line and modal voltages. At the source location A, there is no delay difference, due to even and odd modes, and the net line voltage at the victim is zero as shown in the first plot. As the odd and even modes travel along the line, the skew starts to accumulate, and the waveform at location B shows the typical forward crosstalk waveform. The second plot in Figure 5.3 demonstrates how the skew has distorted both the aggressor and victim waveforms. At the aggressor, the skew has created a porch that causes timing jitter. In a realistic simulation, this porch is seen as edge slow down. On the other hand, at the victim line, the skew has generated a pulse. When the length of the line is extremely long, the skew between the odd and even mode waveforms can be more than a bit time, causing the complete decoupling of the odd and even waveforms, as shown in the third plot in Figure 5.3. In practice, the coupled region of traces is typically short, and this degenerate case seldom occurs.

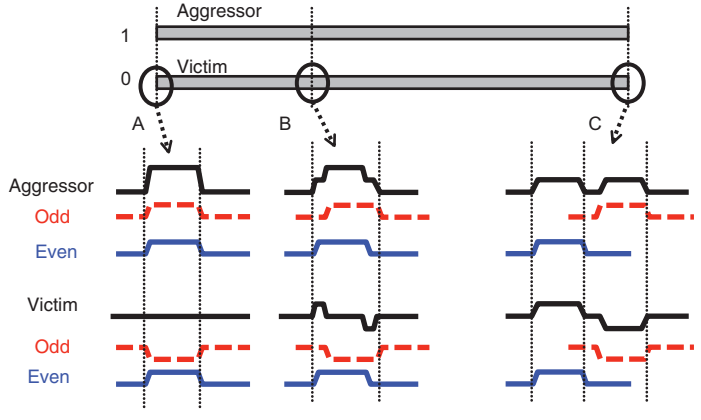


Figure 5.3 Example of Forward Crosstalk Based on a Coupled Transmission Line

Figure 5.4 demonstrates the odd and even mode velocity difference, using the transmission line described in (5.22). The length of the line is 40 cm. All the source and load ends are termi-

nated with Z_o. The victim line is excited with a PRBS data pattern. Examine three different aggressor data patterns: quiet, even, and odd. As expected, the odd-mode data pattern arrives first, followed by the single-ended mode (quiet data pattern), and the even-mode data pattern. It is interesting to note that only the single-ended mode has a porch during the rising or falling edge, because it consists of both even and odd modes, as predicted in Figure 5.3. On the other hand, the pure even or odd modes propagate at one speed and do not have any porch.

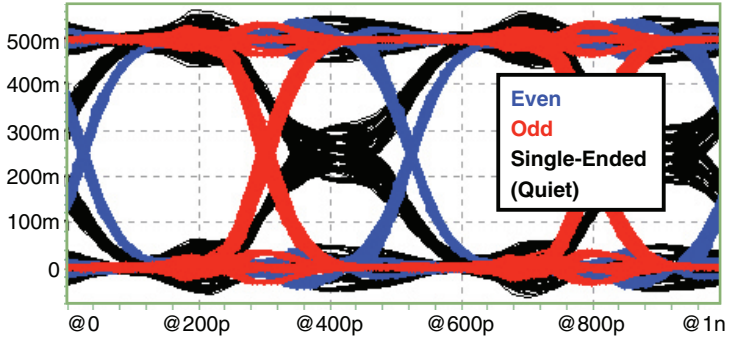


Figure 5.4 Quiet-, Even-, and Odd-Mode Data Patterns for Microstrip

Because forward crosstalk is due to the modal velocity difference, there is no forward crosstalk for traces in homogeneous media, because they all have the same modal velocity—see (5.16). Consequently, a homogeneous strip-line case does not have forward crosstalk. However, it can have far-end crosstalk due to various mode conversions and reflections (backward crosstalk). A buried microstrip has less forward crosstalk, because the difference in mode velocity is relatively small when compared to a conventional microstrip.

For high-speed digital cables, using a homogenous medium, such as coaxial or shielded twisted cables, is highly desirable to avoid forward crosstalk. One can also significantly reduce forward crosstalk by twisting cables, because it averages out the crosstalk impact. With differential lines, the twisting orientation of the two differential pairs should be in opposite directions from each other to further average out any accumulation of crosstalk from the two twisted pairs.

To conclude this section, derive the forward crosstalk expression for the far-end crosstalk (V_F) of the symmetric two-line case shown in Figure 5.1 based on the modal analysis. A more general expression that is valid for any skew between even and odd modes is derived. The skew, between the even and odd modes, is calculated as:

$$t_{skew} = t_{odd} - t_{even} = l\sqrt{(L_{11} - L_{12})(C_{11} - C_{12})} - l\sqrt{(L_{11} + L_{12})(C_{11} + C_{12})}$$

$$\cong l\sqrt{L_{11}C_{11}}\left(-\frac{C_{12}}{C_{11}} - \frac{L_{12}}{L_{11}}\right) = l\sqrt{L_s\,(C_s + C_m)}\left(\frac{C_m}{(C_s + C_m)} - \frac{L_m}{L_s}\right). \tag{5.23}$$

Here, the approximation of $\sqrt{1 - x} \cong 1 - x/2$ is applied to simplify the expression. Now, the following expressions for forward crosstalk are derived by overlapping the even- and odd-mode pulses with different skew amounts:

$$v_F = (t_{odd} - t_{even})\frac{V_s}{t_r}$$

$$\cong \frac{l\sqrt{L_s\,(C_s + C_m)}}{2}\left(\frac{C_m}{(C_s + C_m)} - \frac{L_m}{L_s}\right)\frac{V_s}{t_r} \tag{5.24a}$$

for $|t_{odd} - t_{even}| < t_r$ and for $|t_{odd} - t_{even}| > t_r$, we have

$$v_F = \frac{V_s}{2}. \tag{5.24b}$$

Further approximating $\sqrt{L_s\,(C_s + C_m)}$ to $\sqrt{L_sC_s}$ results in (5.17a). Figure 5.2 can be generalized based on equation (5.24a) to include different mode skew amounts, as shown in Figure 5.5.

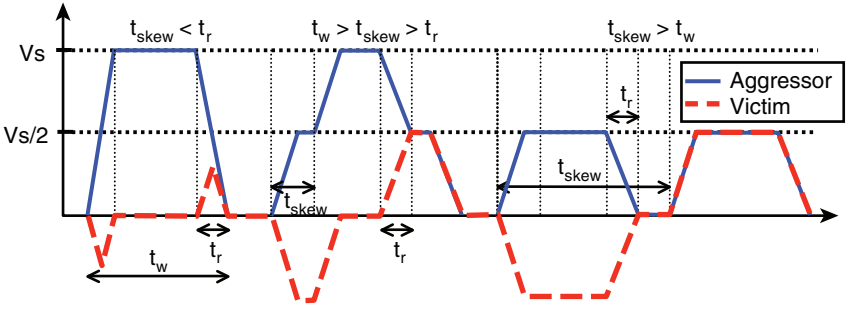


Figure 5.5 Far-End Crosstalk Based on Modal Analysis

5.3 Time-Domain Simulation of Transmission Lines

Transient simulation of lossy transmission lines has been a hot topic for the last two decades. The fundamental difficulty encountered in the transient simulation of a transmission line is that its frequency characteristics are transcendental, and that its terminations can be non-linear devices. The

non-linear device limits the analysis to be performed only in the time domain. A conventional or direct approach for handling the transcendental nature of transmission lines is numerical convolution with the Fourier or Laplace transformation [15–17].

The W-element in HSPICE [18] has gained wide adoption due to its modeling efficient and accuracy. Although the W-element, or similar algorithms, has been used for more than a decade, there are still on-going discussions concerning the accuracy of transient simulation [19]. The W-element uses a recursive convolution in conjunction with matrix delay extraction to simulate transmission lines. This section describes the basics of a transmission-line simulation technique and presents some of the challenges and potential issues. Transient simulation of lossy transmission lines can be unstable for severely lossy lines. This section provides a few tips to avoid this instability. In addition to the topic of recursive convolution, the section also reviews a traditional direct convolution technique.

5.3.1 Transmission Line Model Based on Method of Characteristics

Figure 5.6 shows an equivalent circuit representation of transmission lines that is the most suitable for modified nodal analysis (MNA). Based on the method of characteristics [20], the detail expressions can be derived as follows:

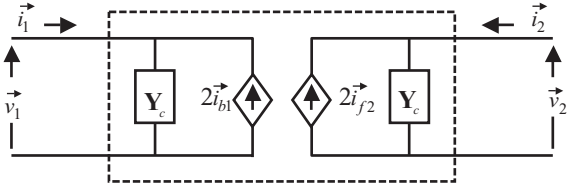


Figure 5.6 Transmission Line Circuit Model Representation

$$\vec{i}_1(\omega) = \mathbf{Y}_c(\omega)\vec{v}_1(\omega) - 2\vec{i}_{b1}(\omega) \tag{5.25a}$$

$$\vec{i}_2(\omega) = \mathbf{Y}_c(\omega)\vec{v}_2(\omega) - 2\vec{i}_{b2}(\omega) \tag{5.25b}$$

where

$$\vec{i}_{b1}(\omega) = e^{-\boldsymbol{\Psi}(\omega)l}\left(\vec{i}_2(\omega) + \vec{i}_{f2}(\omega)\right) \tag{5.25c}$$

$$\vec{i}_{f2}(\omega) = e^{-\boldsymbol{\Psi}(\omega)l}\left(\vec{i}_1(\omega) + \vec{i}_{b1}(\omega)\right). \tag{5.25d}$$

The exponential propagation matrix ($e^{-\boldsymbol{\Psi}(\omega)l}$) contains delay and exponential decay. Long transmission lines reveal severe exponential attenuation. A rational function, which is used to fit the exponential propagation matrix in the recursive convolution approach, does not really model the exponential decay well, due to its slow varying nature. This deficiency becomes more severe

as the line length increases. One way to improve the accuracy of long transmission lines is to segment them into smaller lines to reduce the exponential attenuation of each segment. A few segments are typically enough. However, using too many segments can lead to inaccurate results.

The delay of $e^{-\boldsymbol{\Psi}(\omega)l}$ is also an issue for rational function approximation, as it generates highly oscillatory behavior. Consequently, extracting the delay before applying rational function approximation is preferable. In the early days, the transient simulation techniques, based on direct convolution, did not extract the delay, and the resulting time-domain waveform showed a non-causal response. Extracting the delay helps both the recursive and direct convolution methods in terms of accuracy and simulation time. Because the most dominant delays in transmission lines are due to the inductance and capacitance, the following formula can be used to extract the delay [18]:

$$e^{-\tilde{\boldsymbol{\Psi}}(\omega)l} = e^{-\boldsymbol{\Psi}(\omega)l}\, e^{j\omega\boldsymbol{\tau}_d} \tag{5.26}$$

where $e^{-\boldsymbol{\Psi}(\omega)l}$ is the delay-less exponential propagation function, and the delay matrix $\boldsymbol{\tau}_d$ is represented by:

$$\boldsymbol{\tau}_d = [\mathbf{C}(\infty)\mathbf{L}(\infty)]^{1/2} l. \tag{5.27}$$

Note that the asymptotic values of $\mathbf{C}(\omega)$ and $\mathbf{L}(\omega)$ are used in the preceding equation. Any dynamic delay terms are still in (5.26). To model the extracted delay in the time domain, the following decomposition can be used:

$$e^{-j\omega\boldsymbol{\tau}_d} = \mathbf{M}_\tau e^{-j\omega\hat{\boldsymbol{\tau}}_{dm}} \mathbf{M}_\tau^{-1}. \tag{5.28}$$

Here, $\hat{\boldsymbol{\tau}}_{dm}$ and $\mathbf{M}_\tau$ are the eigenvalue and eigenvector matrices of $\mathbf{C}(\infty)\mathbf{L}(\infty)$. All the $\boldsymbol{\tau}_d$, $\hat{\boldsymbol{\tau}}_{dm}$, and $\mathbf{M}_\tau$ matrices are real and frequency-independent. The modal exponential delay matrix corresponds to a shift of the modal input signals in the time domain and is easily implemented. This time-domain delay element requires a sampling time step that is smaller than the delay itself. Consequently, the largest simulation time steps in transmission-line systems are limited by the smallest delay of any transmission line model. A short transmission line can be more effectively modeled using a few RLGC lumped elements. The dispersion effects caused by skin effect and dielectric loss are negligible for short lines, and a constant lumped RLGC model is sufficient. The coupling resistance often requires complex voltage-dependent models, but these can be omitted for short lines, as long as it does not contribute to major attenuation or coupling.

5.3.2 Companion Model for Time-Domain Simulation

The transmission-line model shown in Figure 5.6 can be easily converted to the companion model as shown in Figure 5.7. Then, this companion model can be applied to modified nodal

analysis (MNA) [21]. This section describes the companion model for the direct convolution approach. The subsequent section describes the companion model for the recursive convolution approach.

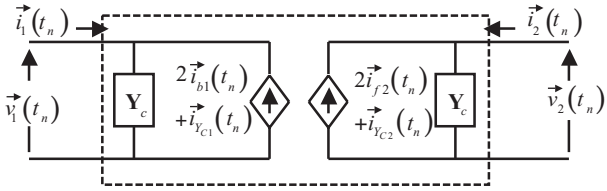


Figure 5.7 Companion Transmission-Line Model

The direct convolution method has been known to be numerically inefficient and inaccurate. Early implementations of a direct convolution approach were applied without extracting delay; as a result, they produced non-causal responses in addition to the computation time overhead [15–17]. For transmission lines, the delay extraction can be done analytically, as shown in the previous section. With a proper delay extraction, the direct convolution method does not add significant simulation time when compared to the recursive convolution approach, and it provides a robust and stable way to simulate transmission lines.

The stability issue becomes more important when simulating general S-parameter models. Recent studies have been conducted on the delay extraction of the S-parameter based on a numerical algorithm [22] [23]. With short transmission lines, or strong coupling terms, the direct convolution of the S-parameter can result in an inaccurate response due to aliasing (refer to Section 5.2.2). However, the direct convolution of transmission line models does not suffer from this issue, because the convolution of transmission line parameters can be performed in a band-limited way, as described in the following paragraphs.

Consider a companion model for the characteristic impedance ($\mathbf{Y}_c$) in (5.18). To fill the admittance matrix for MNA, the asymptotic value of $\mathbf{Y}_c$ is extracted, as follows:

$$\mathbf{Y}_c(\omega) = \mathbf{Y}_\infty + \widehat{\mathbf{Y}}_c(\omega) \tag{5.29}$$

where $\mathbf{Y}_\infty$ is the asymptotic value of $\mathbf{Y}_c$, which can be easily calculated numerically, because it saturates rather quickly. Then, the convolution is applied to only $\widehat{\mathbf{Y}}_c(\omega)$, which has a limited bandwidth. Now, the equation for the companion model, at time t_n, can be written as:

$$\vec{i}_i(t_n) = \hat{\mathbf{Y}}_c \vec{v}_i(t_n) - [\,2\vec{i}_{bi}(t_n) + \vec{i}_{Yci}(t_n)\,] \tag{5.30}$$

where

$$\vec{i}_{Yc1}(t_n) = \hat{\mathbf{Y}}_c(t) * v_1(t_n) \tag{5.31a}$$

$$\vec{i}_{Yc2}(t_n) = \hat{\mathbf{Y}}_c(t) * v_2(t_n) \tag{5.31b}$$

and

$$\begin{aligned}\vec{i}_{b1}(t_n) &= e^{-\boldsymbol{\Psi}(\omega)l} * \left(\vec{i}_2(t_n) + \vec{i}_{f2}(t_n) \right) \\ &= e^{-\tilde{\boldsymbol{\Psi}}(\omega)l} * \left[e^{-j\omega\tau_d} \left(\vec{i}_2(t_n) + \vec{i}_{f2}(t_n) \right) \right] \end{aligned} \tag{5.32a}$$

$$\begin{aligned}\vec{i}_{f2}(t_n) &= e^{-\boldsymbol{\Psi}(\omega)l} * \left(\vec{i}_1(t_n) + \vec{i}_{b1}(t_n) \right) \\ &= e^{-\tilde{\boldsymbol{\Psi}}(\omega)l} * \left[e^{-j\omega\tau_d} \left(\vec{i}_1(t_n) + \vec{i}_{b1}(t_n) \right) \right]. \end{aligned} \tag{5.32b}$$

The convolution of $e^{-\boldsymbol{\Psi}(\omega)l}$ and the current vectors $\vec{i}_1(t_n)$, $\vec{i}_2(t_n)$, $\vec{i}_{f2}(t_n)$, and $\vec{i}_{b1}(t_n)$, are somewhat complicated. First, the delayed versions of the current vectors $\vec{i}_1(t_n)$, $\vec{i}_2(t_n)$, $\vec{i}_{f2}(t_n)$, and $\vec{i}_{b1}(t_n)$ are calculated in the modal space. The resulting currents are convolved with the delay-less exponential matrix function $e^{-\tilde{\boldsymbol{\Psi}}(\omega)l}$ (which has no asymptotic value), so the direct convolution can be performed more efficiently.

5.3.3 Time-Domain Model with Recursive Convolution

A recursive convolution was first introduced by Semlyen and Dabuleanu [24]; it has been applied to various applications, such as time-domain solvers and device models [27–30]. It reduces the simulation complexity of $O(n^2)$ in direct convolution, to $O(n)$. The recursive convolution relies on the rational function approximation that is, in principle, only valid for lumped systems. Again, in principle, the rational function cannot represent either distributed systems, or any systems with a transcendental behavior. For long lossy transmission lines, the rational function approximation could lead to significantly inaccurate results.

After extracting the dominant delay in the transmission lines, the modeling of the exponential decaying function with a rational function is a key to accurate simulations. Both skin effect and dielectric loss accelerate exponential attenuation and impose a challenge to rational function approximation [19]. Even with a recent advance in the approximation method [25] [26], a rational function cannot adequately model exponential decay over wide frequency ranges, as loss severely increases. As previously mentioned in Section 5.3.1, one can divide a long line into segments.

Once again, first look at a companion model for the characteristic impedance matrix $\mathbf{Y}_c(s)$. In the Laplace domain, $\mathbf{Y}_c(s)$ can be approximated as:

$$\mathbf{Y}_c(s) \cong \frac{\mathbf{P}_M s^M + \cdots + \mathbf{P}_1 s^1 + \mathbf{P}_0}{q_M s^M + \cdots + q_1 s^1 + 1} = \mathbf{Y}_\infty + \sum_{m=1}^{M} \frac{\mathbf{A}_m}{1 + s/\omega_{pm}} \tag{5.33}$$

where q_m and the elements of $\mathbf{Y}_\infty$ ($\mathbf{P}_m$, and $\mathbf{A}_m$) are all real, and ω_{pm} is positive real. Note that a common pole is used for all matrix elements. Alternatively, each matrix element can be fitted using its own poles. The corresponding time-domain expression using a series of decaying exponentials is written as:

$$\mathbf{Y}_c(t) \cong \mathbf{Y}_\infty + \sum_{m=1}^{M} \mathbf{A}_m e^{-\omega_{pm} t}. \tag{5.34}$$

Now, assuming the input $\vec{v}_i(t)$ is piecewise-linearly continuous, $\mathbf{Y}_c(t)*\vec{v}_i(t)$ is expressed as the following convolution integral:

$$\begin{aligned} \vec{i}_{b1}(t_n) &= e^{-\boldsymbol{\Psi}(\omega)l} * \left(\vec{i}_2(t_n) + \vec{i}_{f2}(t_n) \right) \\ &= e^{-\tilde{\boldsymbol{\Psi}}(\omega)l} * \left[e^{-j\omega\boldsymbol{\tau}_d} \left(\vec{i}_2(t_n) + \vec{i}_{f2}(t_n) \right) \right]. \end{aligned} \tag{5.35}$$

By integrating the preceding integral and rearranging terms, the following recursive formula is obtained [18] [27]:

$$\mathbf{Y}_c(t_n)*\vec{v}_i(t_n) = \left(\mathbf{Y}_\infty + \sum_{m=1}^{M} \mathbf{A}_m - \sum_{m=1}^{M} \mathbf{D}_m(T) \right) \vec{v}_i(t_n) - \sum_{m=1}^{M} \mathbf{z}_m(t_n) \tag{5.36a}$$

where

$$\mathbf{z}_m(t_n) = \left(\mathbf{D}_m(T_{n-1}) e^{-\omega_{pm} T_n} - \mathbf{D}_m(T_n) \right) \vec{v}_i(t_{n-1}) + e^{-\omega_{pm} T_n} \mathbf{z}_m(t_{n-1}) \tag{5.36b}$$

$$\mathbf{D}_m(T_n) = \frac{\mathbf{A}_m}{\omega_{pm} T_n} \left(1 - e^{-\omega_{pm} T_n} \right). \tag{5.36c}$$

The companion model is written as:

$$\vec{i}_i(t_n) = \dot{\mathbf{Y}}_c \vec{v}_i(t_n) - \left[2\vec{i}_{bi}(t_n) + \vec{i}_{Yci}(t_n) \right] \tag{5.37a}$$

where

$$\dot{\mathbf{Y}}_c = \mathbf{Y}_\infty + \sum_{m=1}^{M} \mathbf{A}_m - \sum_{m=1}^{M} \mathbf{D}_m(T) \tag{5.37b}$$

$$\vec{i}_{Yci}(t_n) = \sum_{m=1}^{M} \mathbf{z}_m(t_n) \tag{5.37c}$$

and the expressions for $\mathbf{z}_m(t_n)$ and $\mathbf{D}_m$ were given by (5.36b) and (5.36c), respectively.

The recursive convolution of the delay-less exponential propagation function $e^{-\tilde{\Gamma}(\omega)l}$ is calculated in a straightforward manner by approximating with a rational function, as follows:

$$e^{-\tilde{\Gamma}(s)l} \cong \sum_{m=1}^{M} \frac{\mathbf{A}_m}{1 + s/\omega_{pm}}. \tag{5.38}$$

Note that there is no infinity term for the exponential propagation function. A recursive convolution can be applied to the delayed versions of $\vec{i}_1(t_n)$, $\vec{i}_2(t_n)$, $\vec{i}_{f2}(t_n)$, and $\vec{i}_{b1}(t_n)$, similar to (5.36a).

In the previous derivation, the assumption was that the input signal is piecewise linear. A similar expression can be obtained by assuming a piecewise step approximation, but the computational complex remains the same yet it reduces simulation accuracy. On the other hand, higher-order approximations could improve simulation accuracy, but they significantly increase computation time. Note that, generally, the rational approximation may result in complex conjugate poles. Then, $\mathbf{A}_m$ and ω_{pm} could be complex conjugate pairs with positive real values. One can still derive the recursive convolution formulae for this case, but it is a rather long complex expression. One can simplify the complexity of expression, based on piecewise step assumption, but the resulting equation is still quite complicated, and replacing the complex poles with real poles may be more efficient. The complex conjugate pairs represent oscillatory behavior in the frequency domain, and it is not necessary when modeling transmission lines, which have a smooth monotonic behavior.

5.4 Modeling Transmission Line from Measurements

This section discusses preparing accurate transmission line models, and describes the details of how to take S-parameter measurements, de-embed port discontinuities, and convert to transmission line parameters, based on the method described by J. Kim, D. H. Han, W. Kim, D. Oh, and C. Yuan [31–33]. This section also discusses two potential accuracy issues associated with measurement-based models. First, a measurement error near the resonant frequency is examined. The characteristic impedance measurement is very sensitive to the reflection caused by the port discontinuities; as a result, characterizing the characteristic impedance over a wide frequency range based on measurements is difficult. To address the port discontinuity issues with the characteristic impedance measurement, the section presents a de-embedding technique that mitigates the port discontinuity issue. Second, a time-domain simulation error due to inaccurate DC values is identified. A hybrid approach, which uses a combination of both time and frequency measurement data, is proposed to mitigate the DC accuracy issue. Several measurement examples, such as MCM-L coplanar lines and package microstrip lines, are considered to validate the accuracy of the proposed method.

5.4.1 Converting S-Parameter to Transmission Line Parameters

Frequency-domain measurement is considered to be the preferred method for high-frequency measurements. This is particularly true for characterizing coupling effects [41]. This section

covers a method of obtaining transmission line models from S-parameter data. First, the S-parameter is converted to ABCD parameters, using the conversion formula shown in Table 4.3 in Chapter 4. Then, the conversion between ABCD and transmission line parameters is done using the formula in Table 5.1.

The **D** matrix of the ABCD parameter is diagonalized to find the propagation constant matrix, using (5.4):

$$\mathbf{D} = \cosh(\boldsymbol{\Psi} l) \equiv \mathbf{M}\cosh(\hat{\boldsymbol{\Psi}} l)\mathbf{M}^{-1} = \mathbf{M}\begin{bmatrix} \cosh(\psi_1 l) & & 0 \\ & \ddots & \\ 0 & & \cosh(\psi_N l) \end{bmatrix}\mathbf{M}^{-1} \tag{5.39a}$$

where $\hat{\boldsymbol{\Psi}}$ is the modal propagation constant matrix. Applying the same diagonalization to the measured **D**, we have:

$$\mathbf{D} = \mathbf{M}\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}\mathbf{M}^{-1}. \tag{5.39b}$$

Finally, the propagation constant matrix is calculated as:

$$\boldsymbol{\Psi} = \mathbf{M}\hat{\boldsymbol{\Psi}}\mathbf{M}^{-1} = \mathbf{M}\left(\frac{1}{l}\begin{bmatrix} \cosh^{-1}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \cosh^{-1}(\lambda_N) \end{bmatrix}\right)\mathbf{M}^{-1} \tag{5.40}$$

where $\mathrm{Re}\{\cosh^{-1}(\lambda_i)\} \geq 0$. To satisfy the causality, the positive attenuation factor must be chosen. The imaginary part of the propagation constant needs special attention: the cyclically mapped phase output of the S-parameters should be unwrapped to the true radian phase. Note that the propagation constant matrix $\boldsymbol{\Psi}$ is generally not symmetric for multi-conductor transmission lines.

The characteristic impedance matrix ($\mathbf{Z}_c$) can be easily computed using the measured **C** matrix, as follows:

$$\mathbf{Z}_c = \mathbf{C}^{-1}\sinh(\boldsymbol{\Psi} l) \tag{5.41a}$$

where

$$\sinh(\boldsymbol{\Psi} l) = \frac{1}{2}(e^{\boldsymbol{\Psi} l} - e^{-\boldsymbol{\Psi} l}) \tag{5.41b}$$

and

$$e^{\boldsymbol{\Psi} l} = \mathbf{M}e^{\hat{\boldsymbol{\Psi}} l}\mathbf{M}^{-1}. \tag{5.41c}$$

It is interesting to note that, unlike the propagation constant matrix, the characteristic impedance matrix is always symmetric, even for the asymmetrical multi-conductor transmission lines. After the propagation constant and characteristic impedance matrices are determined, the frequency-dependent RLGC matrices can be calculated using (5.6).

5.4.2 De-embedding Measurement Parasitic

The high-frequency measurement data is very sensitive to discontinuities caused by two ports at the end of transmission lines. At resonance frequencies, the reflection coefficient (or S11 measurement) is less accurate, resulting in inaccurate characteristic impedance values. On the other hand, the propagation constant measurement does not suffer from this resonance, and it can be extracted accurately, using methods such as the multi-line method, which uses two measurements of two different line lengths [31–33], [42–45]. Figure 5.8 is an example of the geometry for the multi-line method. In this example, the two- or four-port measurements were made using SOLT (Short, Open, Load, and Through) calibrations. All the measured S-parameters are smoothed using a 10-point average FIR filter technique and the number of measurement points is increased using spline interpolation for accurate time domain analysis.

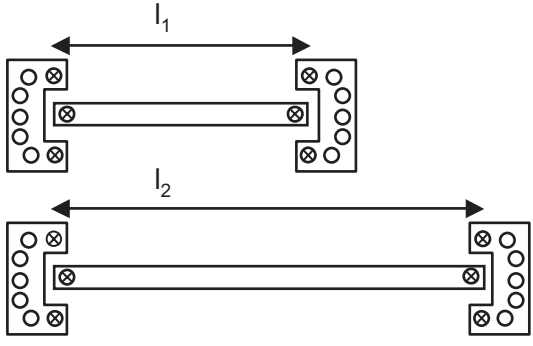


Figure 5.8 Schematic Top Metal Layer for Two Sets of a Single Transmission Line with Two Different Lengths

Using Table 5.1, the measured transmission matrix for two different lengths of the same transmission lines (l_1 and l_2) can be represented using ABCD matrices of the parasitic probe pads and the transmission line, as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_c \cosh(\boldsymbol{\Psi} l_1)\mathbf{Z}_c^{-1} & \mathbf{Z}_c \sinh(\boldsymbol{\Psi} l_1) \\ \sinh(\boldsymbol{\Psi} l_1)\mathbf{Z}_c^{-1} & \cosh(\boldsymbol{\Psi} l_1) \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{Y} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{m_1} & \mathbf{B}_{m_1} \\ \mathbf{C}_{m_1} & \mathbf{D}_{m_1} \end{bmatrix} \tag{5.42a}$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_c \cosh(\boldsymbol{\Psi} l_2)\mathbf{Z}_c^{-1} & \mathbf{Z}_c \sinh(\boldsymbol{\Psi} l_2) \\ \sinh(\boldsymbol{\Psi} l_2)\mathbf{Z}_c^{-1} & \cosh(\boldsymbol{\Psi} l_2) \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{Y} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{m_2} & \mathbf{B}_{m_2} \\ \mathbf{C}_{m_2} & \mathbf{D}_{m_2} \end{bmatrix} \tag{5.42b}$$

where **X** and **Y** are the admittance matrices of the parasitic probe pads at two ends. A lumped representation is used to represent pad discontinuities. It is important to note that **X** and **Y** are full matrices that take the coupling between probes into consideration. Using the inversion of the matrix of the shorter transmission lines and eliminating the admittance matrix **Y** from the equation, the following equation can be derived:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_c \cosh(\boldsymbol{\Psi}(l_2 - l_1))\mathbf{Z}_c^{-1} & \mathbf{Z}_c \sinh(\boldsymbol{\Psi}(l_2 - l_1)) \\ \sinh(\boldsymbol{\Psi}(l_2 - l_1))\mathbf{Z}_c^{-1} & \cosh(\boldsymbol{\Psi}(l_2 - l_1)) \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{X} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \tag{5.43a}$$

where

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{m_2} & \mathbf{B}_{m_2} \\ \mathbf{C}_{m_2} & \mathbf{D}_{m_2} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{m_1} & \mathbf{B}_{m_1} \\ \mathbf{C}_{m_1} & \mathbf{D}_{m_1} \end{bmatrix}^{-1}. \tag{5.43b}$$

By multiplying (5.43) using the inverse of the port discontinuity matrix, the ABCD matrix of the transmission line parameter is calculated as:

$$\begin{bmatrix} \mathbf{Z}_c \cosh(\boldsymbol{\Psi}(l_2 - l_1))\mathbf{Z}_c^{-1} & \mathbf{Z}_c \sinh(\boldsymbol{\Psi}(l_2 - l_1)) \\ \sinh(\boldsymbol{\Psi}(l_2 - l_1))\mathbf{Z}_c^{-1} & \cosh(\boldsymbol{\Psi}(l_2 - l_1)) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{X} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X} & \mathbf{I} \end{bmatrix}. \tag{5.44a}$$

Matrix **A** is the transpose of matrix **D** for uniform transmission lines (see Table 4.1 in Chapter 4). Consequently, $\mathbf{Z}_c \cosh(\boldsymbol{\Psi}(l_2 - l_1))\mathbf{Z}_c^{-1}$ is equal to the transpose of $\cosh(\boldsymbol{\Psi}(l_2 - l_1))$; we can then algebraically compute the parasitic probe pads by multiplying out the right side and comparing the diagonal terms:

$$\mathbf{X} = (\mathbf{M}_{22} - \mathbf{M}_{11}^T)(\mathbf{M}_{12} + \mathbf{M}_{12}^T)^{-1}. \tag{5.44b}$$

After the parasitic probe pads are de-embedded, the propagation constant and characteristic impedance matrices can be calculated using the procedure described in Section 5.4.1. Then, frequency-dependent RLGC matrices for HSPICE simulation are obtained using equations (5.6a) and (5.6b).

5.4.3 Examples

Two microstrip lines are built as a test structure. The lines are 6.558 mm and 13.337 mm long, and 76.6μm wide. The package substrate material has a thickness of 35.5μm. The signal and ground metal layers are copper, with thicknesses of 35.5μm and 16μm, respectively. It is important to note that transmission lines should be designed to be as short as possible (to extract the characteristic impedance in the S-parameter measurements), but not so short that the measurements are affected by the proximity effects between the two probes.

Figure 5.9 shows the propagation constant and characteristic impedance. Data is measured from 45MHz to 26.5GHz, with 801 linear points. The measured characteristic impedance, shown as a solid line in Figure 5.9(b), has discontinuities at $\lambda/2$ due to resonance [43]. Figure 5.10 shows the admittance of the parasitic probe pads.

To extend the modeling frequency range, the characteristic impedance is obtained in the 7-GHz to 26.5-GHz range by fitting data from 45MHz to 7GHz ($\lambda/4$), which is sufficiently far from $\lambda/2$. The high-frequency nature of the characteristic impedance can be extrapolated with acceptable accuracy, because at 7GHz, the characteristic impedance reaches a steady region, which is close to the infinite-frequency characteristic impedance. This is because the field is concentrated near the surfaces of the signal and ground conductors; as a result, internal inductance (due to skin effect) becomes negligible at high frequency, leaving only external inductance (steady term). To guarantee this steady behavior at $\lambda/4$, the trace length of the DUT must be designed accordingly.

Once again, a rational function is used to fit the characteristic impedance [33]. This approximation uses a 14th order rational function. Figure 5.11 shows the comparison between the measured and fitted data up to 7GHz. A good match exists for both the real and imaginary parts of the characteristic impedance. Figure 5.9(b) shows the overall comparison. To verify the accuracy of the extrapolated characteristic impedance, it is compared with a full-wave solver model, shown in Figure 5.9(b). The simulation was performed using HFSS, with the frequency-dependent dielectric constant and loss tangent. At $f = 26.5$GHz, the difference between the extended and simulated characteristic impedance is less than 1%. Now, the characteristic impedance and propagation constant functions are converted to RLGC parameters, as shown in Figure 5.12.

5.4.4 Impact of DC Values in Transmission Line Models

Although an accurate high-frequency model can be obtained from frequency-domain measurements, it still lacks DC values. DC, or low frequency, responses are important for digital signals, because they have a wide bandwidth (from DC to about the third harmonic frequency). Figure 5.13 shows the frequency spectrum of a 5-GHz digital signal with a 20ps risetime. Although the high peaks appear at 5GHz and 15GHz, the low-frequency region below ~1 GHz still has large amplitudes. The low frequency content of the digital signal can be understood more clearly by considering the spectral content of the step response shown in Figure 5.13, and the superposition principle shown in Figure 5.14.

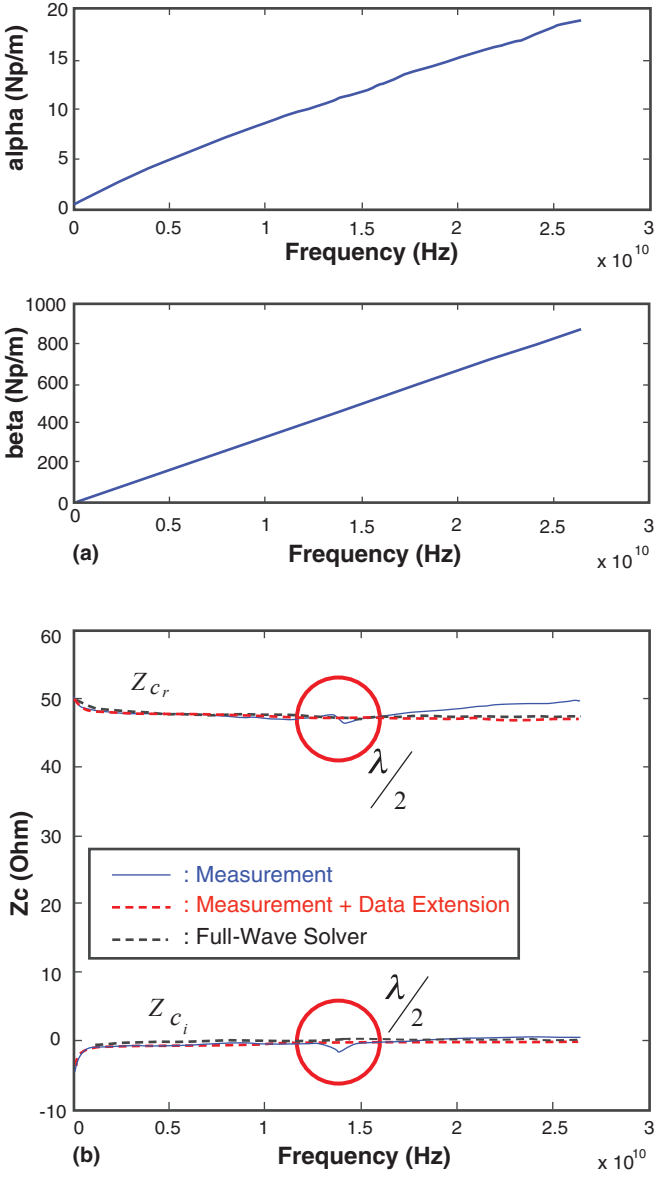


Figure 5.9 (a) Propagation Constant and (b) Characteristic Impedance [33] (© 2010 IEEE)

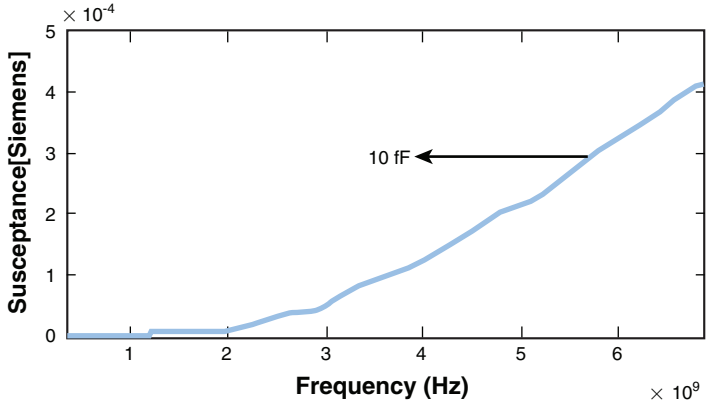


Figure 5.10 Imaginary Part of Admittance for Parasitic Probe Pads [33] (© 2010 IEEE)

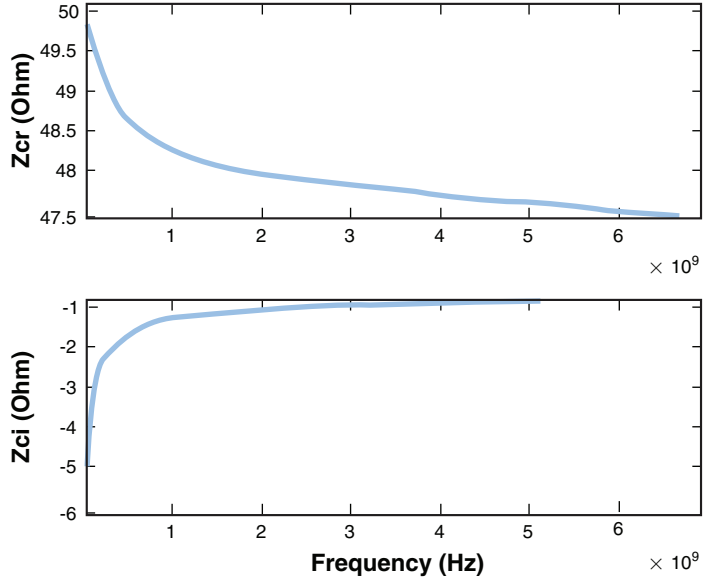


Figure 5.11 Characteristic Impedance of Measurement (Solid Line) and Data Fitting (Dashed Line) [33] (© 2010 IEEE)

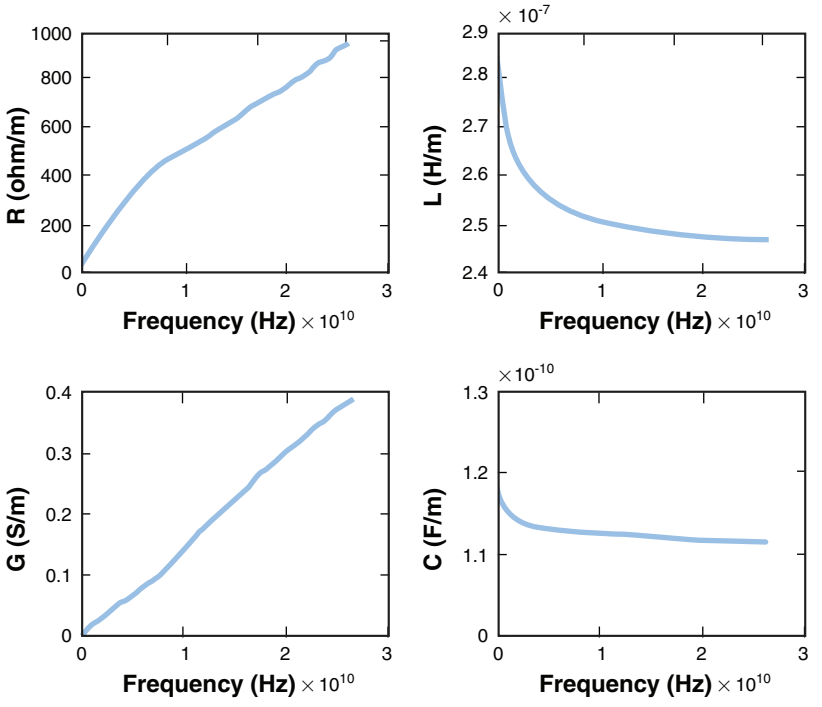


Figure 5.12 RLGC Parameters [33] (© 2010 IEEE)

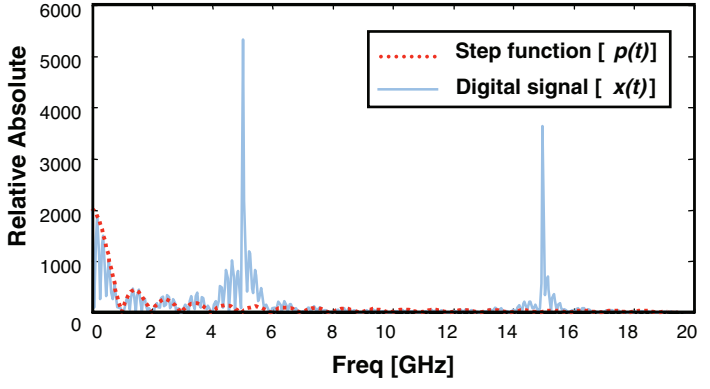


Figure 5.13 Spectrum of the Digital Signal and Step Function

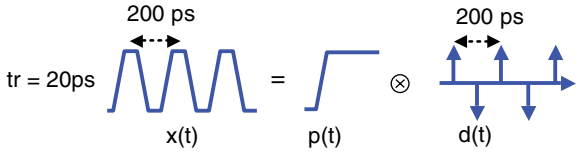


Figure 5.14 Digital Pulses Represented by the Sum of Steps and Impulses

Although DC, or low frequency, responses are important data points for a transmission line, accurately capturing the low-frequency response using either frequency-domain measurements or full-wave solvers is difficult. VNA equipment has a minimum supporting frequency of ~50MHz. As the data rate increases, VNA equipment continues to support higher frequency characterization. Unfortunately, this also raises the lower frequency bound. From the simulation point of view, full-wave Maxwell's equation solvers have difficulties in solving the low-frequency response [34] [35]. Because full-wave solvers are formulated to capture wave phenomena (E and H fields are coupled), they lose accuracy in the frequency range where the structure under consideration is seen as lumped elements (E and H fields are uncoupled). Recently, Zhu and Jiao showed an excellent improvement in this field [36].

As an example, we used VNA to measure the coplanar transmission line, shown in Figure 5.15. The dimensions are shown in the figure. The dielectric constant is 3.8 and the loss tangent is 0.02. The ground conductors of the coplanar line are not connected to the bottom-side metal plane. Two trace lengths, of 2.54mm and 5mm are used for VNA measurements, and a single 50-mm trace is used for TDR measurements.

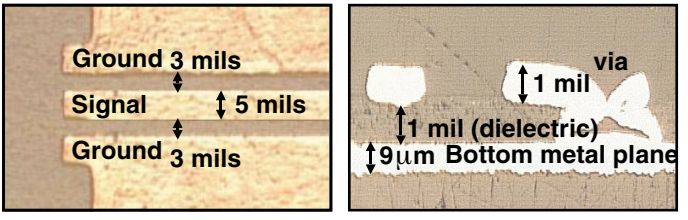


Figure 5.15 Coplanar Line on MCM-L Substrate: Top Layer (Left) and Cross Section (Right) [33] (© 2010 IEEE)

Using the method described in the previous section, one can extract the characteristic impedance and propagation constant, as shown in Figure 5.16. The characteristic impedance and propagation constant can be used to generate the RLGC model of the transmission line which can be simulated using the frequency-dependent tabular W-element model in HSPICE [37]. The

time-domain simulation waveform is compared to the TDR waveform in Figure 5.17. The Tektronix TDR equipment has a risetime of 30ps and amplitude of 250mV. A 5-cm transmission line is connected to ground in the far end.

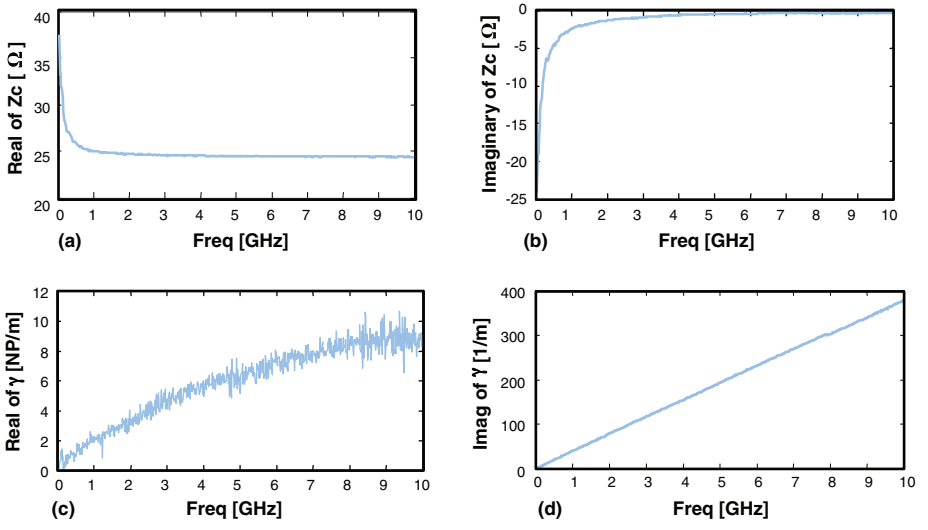


Figure 5.16 VNA Measured Characteristic Impedance and Propagation Constant

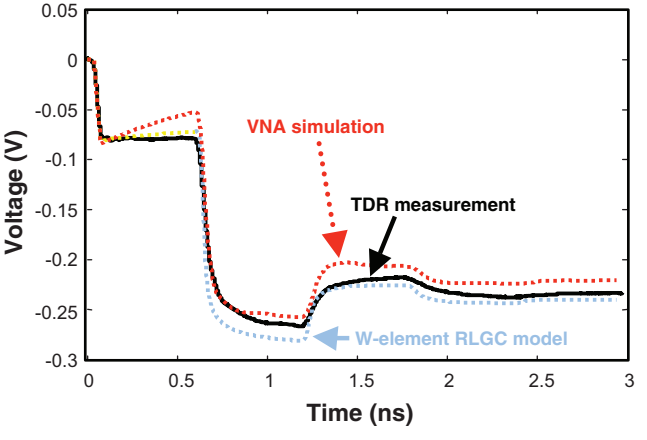


Figure 5.17 VNA-Measured Characteristic Impedance and Propagation Constant

In the simulation, we extrapolate the low-frequency data for the characteristic impedance and propagation constant in Figure 5.16 (from DC to 50 MHz) using the slopes at 50MHz. However, as shown in Figure 5.17, the simulated waveform using the VNA measurement does not agree well with the TDR measurement waveform. This is due to the inadequate DC response related to the low-frequency constraints of the transmission lines. The following paragraphs cover this issue in detail.

Next, the measurement data is also compared with a 2D field-solver model, based on quasi-static analysis such as HSPICE, or Maxwell 2D. The extracted parameters, based on the analytical W-element RLGC model, are as follows:

$$\begin{aligned} R &= 5.468 + 2.1126 \times 10^{-3}\ \sqrt{f}\quad \Omega/\mathrm{m} \\ L &= 142.3\quad \mathrm{nH/m} \\ G &= 1.39453 \times 10^{-11} f\quad \mathrm{S/m} \\ C &= 233.6\quad \mathrm{pF/m}. \end{aligned} \tag{5.45}$$

Figure 5.16 illustrates that the HSPICE simulation, based on this analytical model, also shows a significant discrepancy from the measurement. This problem is again due to inadequate DC modeling. The following paragraphs demonstrate the sensitivity of DC modeling in transmission line models.

To study the sensitivity of transmission line models, consider slightly different RLGC parameters. Table 5.2 provides the detailed values. The four similar transmission lines differ only in their DC resistance values.

Table 5.2 RLGC Models with Different DC Resistances

	Case 1	Case 2	Case 3	Case 4
L_o (nH/m)	340.8			
C_o (pF/m)	112.3			
R_o (Ω/m)	5e-6	300e-6	5	25
R_s (mΩ/m)	1.435			
G_o (S/m)	0			
G_d (pS/m)	13.1			

The transmission line is 0.2 m long, with short termination. All the lines are identical, except for the small differences in DC resistance (R_o). The DC resistances for the four cases are $1\mu\Omega$, $60\mu\Omega$, 1Ω, and 5Ω, respectively. The step source has amplitude of 250mV, a risetime of

30ps, and an output resistance of 50Ω. The source is connected to a line to be measured through a lossless transmission line, whose characteristic impedance is the same as the output resistance in order to simulate the TDR waveforms. The simulations use the HSPICE 2008.09 version. All the transmission lines have a different response, as shown in Figure 5.18. The simulation data shows a strong dependency on DC resistance values, although skin effect and dielectric loss terms supposedly dominate in this example.

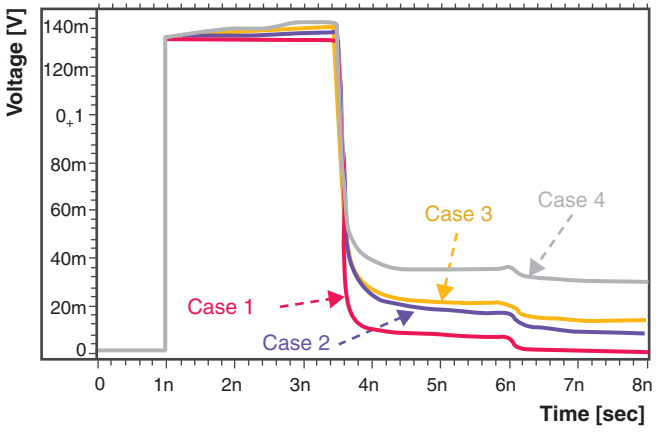


Figure 5.18 Pulse Responses Using HSPICE with RLGC Parameters in Table 5.2 [33] (© 2010 IEEE)

Observe that HSPICE produces an inaccurate response when the DC resistance sets are small ($R_o < 300\mu\Omega/\text{m}$). For example, although the DC resistance of both Case 1 ($1\mu\Omega$) and Case 2 ($60\mu\Omega$) is negligible, the response shows a large discrepancy (see Figure 5.18). This problem can be solved by entering more accurate DC information. The following section presents an accurate way to obtain DC values from the time-domain measurements.

5.4.5 DC Characterization of Transmission Line Parameters

Because frequency-domain measurements have a fundamental problem with low frequency data, a time-domain approach is considered to calculate a correct DC value. A transmission line can be characterized from the transient behavior of the line by using the short-pulse propagation technique [38] or TDR [39]. In this section, TDR is used to extract the transmission line parameters of the low-frequency region. The first reflection from the near end of the transmission line provides information about the characteristic impedance, whereas round-trip reflections from the far end provide information about the propagation constant. As an example, Figure 5.19 shows the TDR waveforms of the coplanar line from Figure 5.15, with both short and open terminations.

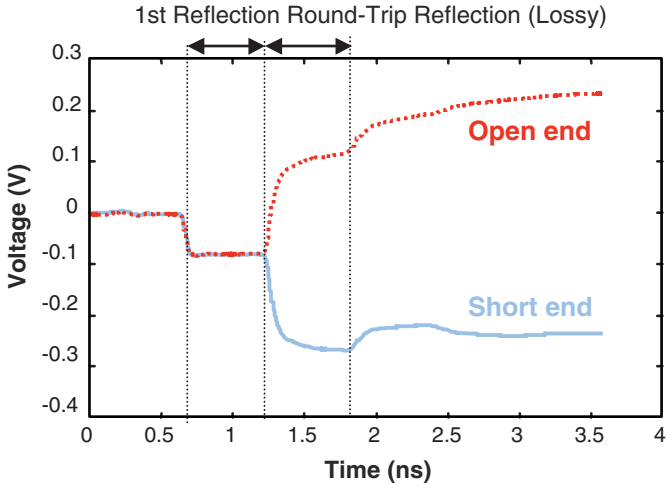


Figure 5.19 TDR Measurement of Transmission Lines with Short and Open Ends

Because the two lines have the same characteristic impedance, due to the identical cross section, the TDR waveforms in Figure 5.19 show the same first reflection. As shown in the figure, the first reflection and round-trip reflections is separated using time windowing, but only if the length of the transmission line is long enough when compared to the risetime and falltime of the signal. In general, the impedance can be calculated using the following equation from the TDR waveforms:

$$impedance = 50\frac{1+\Gamma}{1-\Gamma} \tag{5.46}$$

where 50Ω is the impedance of the TDR equipment and cables. In Figure 5.19, the amplitude of the input is 250mV. The impedance of the short-end waveform at the steady state (t ~3.5ns) is 1.2Ω, which is the DC resistance of the line. At t = 1.2ns, the voltage is –80.2mV, and the impedance is 25.7Ω.

To match simulation waveforms with the TDR measurements shown in Figure 5.19, the transmission line model must satisfy two low-frequency conditions. Here, these conditions are derived using the final-value theorem [40], and the static DC resistance. If the limits exist, the final value of the function $y(t)$ for time $t \to \infty$ is equal to the limit of the product of the corresponding Laplace transform $Y(s)$, and the variable s as $s \to 0$ [40]:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0}[sY(s)]. \tag{5.47}$$

This is the final-value theorem of the Laplace transform. The final-value theorem can be applied to the step response, as shown in Figure 5.19. In this figure, a step $x(t)$ is applied to a system $H(s)$, which is the Laplace transform of $h(t)$. The system response is $y(t) = x(t)*h(t)$. Because the Laplace transform of $x(t)$ is $1/s$, $Y(s)$ is $H(s)/s$. Now, we apply the final-value theorem for the ideal step response to $Y(s)$:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0}[sY(s)] = \lim_{s\to 0} H(s) \tag{5.48}$$

where the final-value of $y(t)$ is related to the low-frequency value of $H(s)$, with $s = j\omega = j2\pi f$. Although the final-value theorem is developed using the ideal step function, it is also valid for a step function with a finite risetime.

Applying this final-value theorem to the TDR waveforms in Figure 5.19, the first reflection is written as:

$$H(s) = \frac{Z_c(s) - 50}{Z_c(s) + 50} \tag{5.49}$$

where $Z_c(s)$ is the characteristic impedance of the transmission line, and 50Ω is the characteristic impedance of the signal source and cables. Then, according to the final value theorem in (5.48), the final value of the first-reflection $y(t)$ in Figure 5.19 has the following relationship:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} H(s) = \frac{Z_c(s\to 0) - 50}{Z_c(s\to 0) + 50}. \tag{5.50}$$

The first reflection can be measured using TDR measurements if the length of the transmission line is large when compared to the risetime of the step input (see Figure 5.19). The longer transmission line has the longer first reflection time window. Therefore, if the length of the transmission line is infinite, $y(t\to\infty)$ can be measured. However, the infinite-length transmission line is not required to measure $y(t\to\infty)$, because $y(t)$ quickly reaches a steady-state value that is equal to $y(t\to\infty)$. In Figure 5.19, the value at $t = 1$ns can be used for $y(t\to\infty)$. Then, based on (5.44), $Z_c(f\to 0)$ can be calculated from the first reflection in Figure 5.19. For the lines with DC losses, $Z_c(f\to 0)$ is given by:

$$Z_c(f\to 0) = \sqrt{\frac{R_{DC}}{G_{DC}}} \tag{5.51}$$

where R_{DC} is the per-unit-length DC resistance of the transmission line, and G_{DC} is the per-unit-length DC conductance. For the example shown in Figure 5.19, the amplitude of the input is 250mV, and $Z_c(f\to 0) = 25.7\Omega$.

The second low-frequency condition can be derived from the static resistance of the transmission line. The steady-state response of TDR waveforms must reach the static resistance value at the end, as shown in Figure 5.19. The static resistance for this case is 1.2Ω based on the short-terminated step response. This value matches the calculated value based on the cross section and conductivity information. We can express the steady-state input impedance, seen from the near end of the transmission line in Figure 5.19, as:

$$Z_{in} = Z_c \frac{R_L + Z_c \tanh(\gamma l)}{Z_c + R_L \tanh(\gamma l)} \tag{5.52}$$

where γ and l are the propagation constant and length of the transmission line, respectively, and R_L is the termination. With DC, the following simplifications are valid for transmission lines:

$$Z_c\,(f \to 0) = \sqrt{\frac{R_{DC}}{G_{DC}}},\ \alpha_{DC} = \sqrt{R_{DC}\,G_{DC}},$$
$$\tanh(\alpha_{DC} l) \approx \alpha_{DC} l \tag{5.53}$$

where α_{DC} is the DC attenuation constant. Then, the input impedance at DC, $Z_{in}\,(f \to 0)$, can be simplified as follows:

$$Z_{in}\,(f \to 0) = \frac{R_L + R_{DC} l}{1 + R_L G_{DC} l}. \tag{5.54}$$

Based on (5.51) and (5.54), from the low-frequency components, R_{DC} and G_{DC}, can be calculated using the following expressions:

$$G_{DC} = \frac{R_{DC}}{Z_c\,(f \to 0)^2} \quad [\mathrm{S/m}] \tag{5.55a}$$

$$R_{DC} = \frac{1}{l} \frac{Z_{in}\,(f \to 0) - R_L}{1 - \dfrac{R_L}{Z_c\,(f \to 0)^2} Z_{in}\,(f \to 0)} \quad [\Omega/\mathrm{m}]. \tag{5.55b}$$

A convenient way to measure R_{DC} is to use transmission lines with short terminations, where $R_{DC} = Z_{in}\,(f \to 0)/l$. Referring to Figure 5.19, since $Z_{in}\,(f \to 0) = 1.2\Omega$ for a 5-cm length, then $R_{DC} = 24.0\Omega/\mathrm{m}$. Also, since $Z_c\,(f \to 0) = 25.7\Omega$, then $G_{DC} = 36.3\mathrm{mS/m}$. Note that G_{DC} cannot be accurately measured with the impedance measurement using open termination.

When both G_{DC} and R_{DC} are zero, $Z_c\,(f \rightarrow 0) = \sqrt{L(f \rightarrow 0)/C(f \rightarrow 0)}$. When $G_{DC} = 0$, and $R_{DC} \neq 0$, then Z_c becomes infinity at DC. This zero G_{DC} value causes a DC convergence problem for time-domain simulations, and is often approximated to match $Z_c\,(f \rightarrow \infty)$, or assumed a small arbitrary value. A better approach is to approximate G_{DC} with small values using the TDR response. Recently, the S-parameter has been used to directly simulate transmission lines, without converting to a transmission line model. This S-parameter model at DC also suffers from the same inaccuracy problem, and the DC condition of the S-parameter must be set properly, as follows:

$$S_{11} = S_{22} = \frac{R_{DC}G_{DC} + R_{DC}/Z_o - G_{DC}Z_o}{2 + R_{DC}G_{DC} + R_{DC}/Z_o + G_{DC}Z_o}$$

$$S_{21} = S_{12} = \frac{2}{2 + R_{DC}G_{DC} + R_{DC}/Z_o + G_{DC}Z_o} \tag{5.56}$$

where Z_o is the reference characteristic impedance.

In the absence of timing-domain measurements, one can interpolate the DC points of the S-parameter data using the generalized dispersion relations described in Section 5.4.1. The corresponding transmission-line parameter can be obtained from this interpolated S-parameter using the equations shown in Section 5.4.1.

To demonstrate the proposed DC enhancement, the low-frequency constraint is applied to the MCM-L transmission line models in Figure 5.15. In Figure 5.17, the frequency-dependent W-element model (extracted from VNA without any DC constraints) showed a discrepancy in the TDR measurement waveform. Based on the TDR measurement in Figure 5.19, the DC resistance and conductance of the line are extracted as $R_{DC} = 24.0\Omega/\text{m}$, and $G_{DC} = 36.3\text{mS}/\text{m}$. Figure 5.20 shows the final comparison. As the figure illustrates, the enhanced model with DC correction has a good correlation to measurement.

5.5 On-Chip Wire Modeling

As I/O speed continues to increase, even on-chip routings start to exhibit transmission-line effects. This section discusses the transmission line modeling of on-chip interconnects. Traditionally, on-chip global nets are modeled using RC or RLC networks. Due to the highly lossy nature of on-chip wires, circuit engineers insert a buffer to account for loss and to minimize a delay due to the RC network. However, buffers are subject to power-supply noise, and minimizing the number of buffers is desirable. In a high-speed I/O interface, modeling the global clock net is more important than the core area, because the I/O clock frequency is often much higher than the core clock frequency. Modern I/O interfaces operate at multi-gigahertz data rates that will soon reach 10Gb/s in future designs. This will require routing the 5GHz clock signal net over the interface area, which

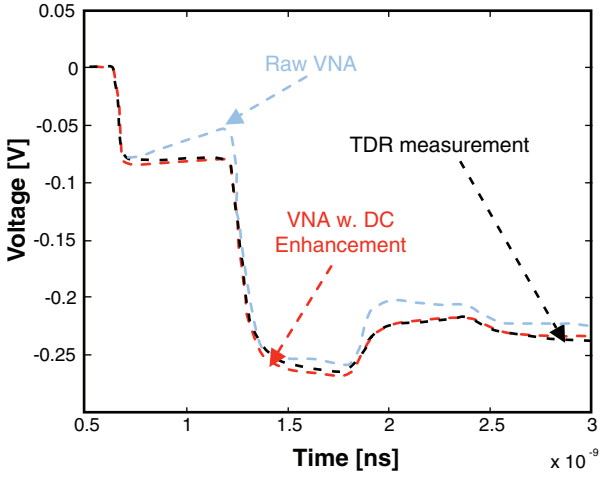


Figure 5.20 Low-Frequency Compensated Time-Domain Simulation Using VNA Measurement

often covers the entire die side of the processor. On the other hand, the core frequency remains relatively constant due to peak power consumption, and chip performance is increased by using multicores. With high frequency I/O clock nets, on-chip wires can no longer be modeled using simple lumped RC models; transmission line models or distributed RLGC models are preferable [46–48].

In addition to the high frequency nature of the I/O clock, each segment of clock wires for the I/O interface is relatively long when compared to core clock wires. This is because a smaller number of repeater buffers is used for I/O to minimize the buffer jitter induced by power-supply noise. For low-power applications, the LC resonance tank is considered as a clock distribution [49]. This resonant circuitry saves power consumption by eliminating the need for buffers. The importance of accurately modeling the clock wires is even more pronounced for this application, because the wire inductance becomes a part of the LC resonance tank. The on-chip modeling method presented by Qi, et al. [50] was originally developed to model clock wires for the LC resonance tank, but it can be applied to other general on-chip clock wires. This section reviews the on-chip wire modeling method described by Qi, et al. [50].

5.5.1 Challenges of On-Chip Wire Modeling

Off-chip PCB, or package, traces have well-defined return paths for signals. This is not true for on-chip wires, because they do not have a clear signal return path. The power or ground planes in the PCB or package are replaced by grids. Because the return path is loosely defined, it varies with the signal frequency contents. As a result, resistance and inductance of an on-chip wire is a strong function of frequency. The low-frequency contents of a signal return through a wide area of neighboring wires, whereas the high-frequency signal contents return through very close neighbor wires via the proximity effect. This can result in inaccurate low-frequency models due

to a limited modeling area. This frequency dependency of resistance and inductance is quite distinct from the conventional frequency variation caused by skin effect in PCB and package traces. In fact, the impact of skin effect is very small for on-chip wires, because the wire cross sections are typically smaller than the skin depth.

Figure 5.21 illustrates the proximity effect on the return path. At gigahertz frequencies, the proximity effect impacts on-chip wires in a profound way. Due to high frequencies, currents tend to return closer to the signal wires, resulting in a smaller loop inductance, as shown in Figure 5.21. This is because current always tries to find the path of least impedance. At DC or low frequencies, wire resistance dominates the wire loop impedance, instead of inductance. Therefore, currents spread themselves out over many parallel paths to minimize the resistance (see Figure 5.21). For the small-wire geometries used in current IC technologies, the proximity effect dominates, and the skin effect is not evident.

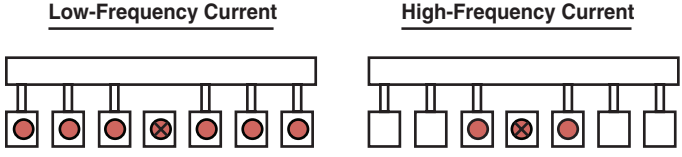


Figure 5.21 Return Current Paths for Different Signal Frequencies

5.5.2 Efficient Model Representation for On-Chip Wires

Although a full transmission line model can be used for on-chip wires, it can present a significant burden to simulation efficiency, because a short transmission line limits a simulation time step. For on-chip wires, distributed elements can be as accurate as transmission lines, without sacrificing the simulation time. PEEC π segments is often used to model the distributed effects of on-chip wires [51]. A standard RC extraction tool (such as QuickCap™) can be used to calculate wire resistance and capacitance for on-chip applications. As typical clock routing in an I/O interface is rather uniform, and on-chip clock nets can be also modeled using a simple 2D static solver. The inductance of wires can be computed using FastHenry [52]. To speed up circuit simulation, the forward couplings between two segments are ignored. Furthermore, wires are modeled using a single filament, because skin effect can be safely ignored and the current distribution is fairly uniform.

Figure 5.22 shows the equivalent RLC circuit model for a typical coplanar clock wire. Assume that the majority of the currents return from the nearest grounds at multi-gigahertz frequencies. The number of π segments can be determined by the clock frequency and the wire length to capture the distributed effects. Note that a full-wave solver is not necessary for more accurate on-chip wire modeling. A full-wave solver not only takes a significantly long simulation time, but it is less accurate when the return path is not well defined.

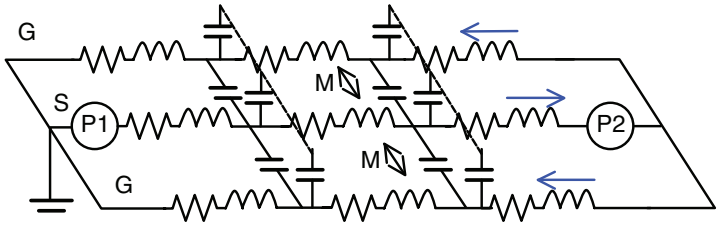


Figure 5.22 Segmented π Model for On-Chip Wires

5.5.3 Model Correlation with Measurements

The proposed on-chip wire model is validated using a test structure implemented in a 90nm process as shown in Figure 5.23. Both single-ended and differential clock traces are built and characterized. The S-parameters from the simulated circuit models are compared to the measured ones.

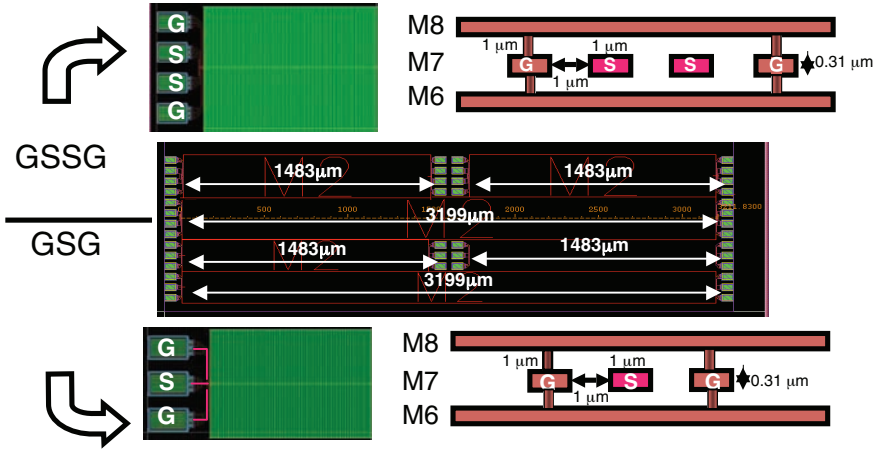


Figure 5.23 Single-Ended and Differential Clock-Test Structures

For on-chip measurements, the parasitic due to the probing pad can cause significant discontinuities. The de-embedding method, based on the two-line method described in Section 5.4.2, may not be adequate for on-chip measurements. On-chip probing requires probing pads with rather long feeding traces to the DUT wires. Consequently, the port discontinuities take a significant portion of the measured responses, and applying the two-line method is inadequate. In this section, a simple RLC model is generated using field solvers. Figure 5.24 shows the probe pad model that includes the wire feed from the pads to the DUT wires. The total feed structure adds ~7% to the total DUT inductance, and ~2% to the total DUT resistance at high frequencies.

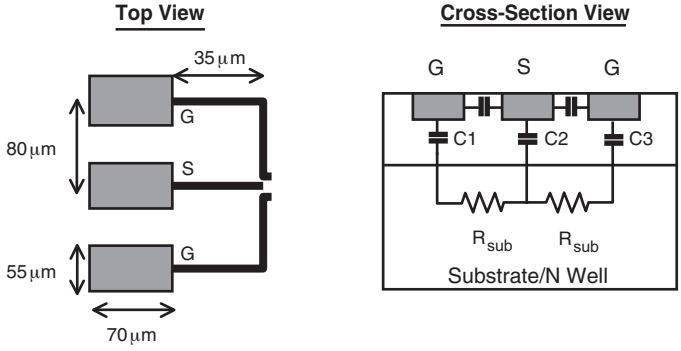


Figure 5.24 Compact Model for Probe Pads and Feeding Wires

Based on this simple RLC model, the S-parameter of the probe structure is calculated. Then, it is de-embedded from the total measured S-parameter, by converting to ABCD parameters using the conversion formulae presented in Table 4.3.

For the first correlation, a single-ended clock wire is considered. Figure 5.25 shows the S-parameter correlation. Excellent agreement is seen between the measurement and the simulation, up to 20GHz. RLC parameters can be obtained from measured S-parameters to compare to the simulated RLC results. Figure 5.26 plots the loop resistance of a single-ended clock tree. As expected, the resistance is flat in the low-frequency region and ramps up at higher frequencies due to the proximity effect. The mismatch at lower frequencies is largely due to the process variations, and its impact on current return loops, which is not modeled in this simulation. There are +/–28% wire resistance variations and +/–50% via resistance variations in this process. The inductance is compared in Figure 5.27. As predicted, inductance becomes smaller at high frequencies because return loops are smaller. Typical inductance variation due to skin effect is much smaller than the variation caused by the proximity effect. Capturing this large inductance frequency variation is extremely important to LC resonance clock distributions. Depending on the operating clock frequency, the inductance contribution from wires can vary significantly, which in turn, determines the peak resonance frequency in the clock distribution. Similar to the resistance case, some mismatch is observed in the low frequency region.

In high-speed I/O interfaces, differential clocks are used even with on-chip wire routing. Figure 5.28 illustrates mixed-mode differential S-parameters that have been measured and correlated. Section 5.1.4 describes the conversion between single-ended and mixed-mode S-parameters. A good match exists at most of the frequency points. The mismatch above 10GHz may be due to the frequency-independent pad model used in the simulation, as the previous pad model does not accurately model delay.

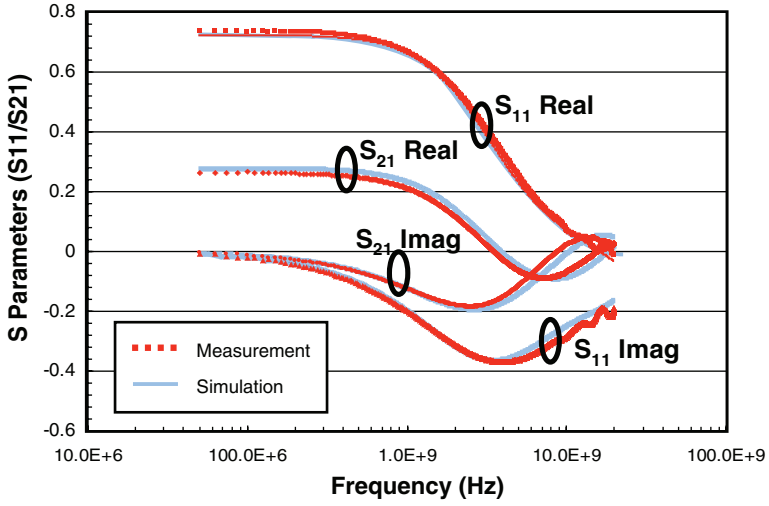


Figure 5.25 S-Parameter Correlation for Single-Ended Clock (Average Error Is 8% [e.g. S11 Imag])

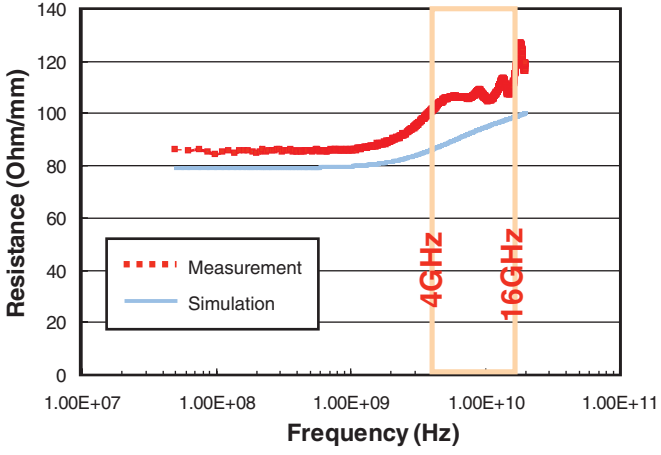


Figure 5.26 Loop Resistance Increases at High Frequencies Due to Proximity Effect (Average Error Is 13%)

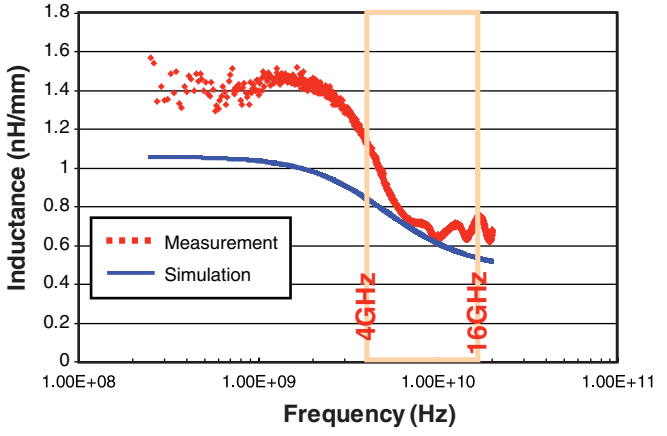


Figure 5.27 Loop Inductance Decreases (2x) at High Frequencies Due to Proximity Effect (Average Error Is 14% Above 500MHz)

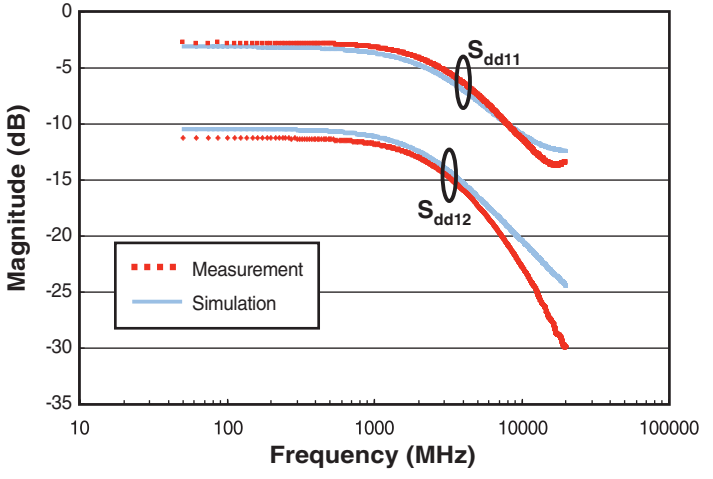


Figure 5.28 Mixed-Mode Differential-to-Differential S-Parameter Comparison for Differential Clock (Average Error Is 10% [e.g., Sdd12])

5.6 Comparison of On-Chip, Package, and PCB Traces

This section compares different transmission line types by their off-chip and on-chip levels: microstrip lines in a typical PCB motherboard, microstrip lines in a BT substrate package, and an on-chip wire. Figure 5.29 provides the geometry information for the motherboard and package

traces. The lengths of the motherboard traces are 8cm and 16cm, whereas the package traces lengths are 6.56mm and 13.34mm. The on-chip wire uses the structure in the previous section, as shown in Figure 5.23. Figure 5.30 shows the converted RLGC parameters. As expected, the resistance of an on-chip wire is significantly higher than the package and PCB traces, and it is constant over a wide frequency range. This indicates that the skin effect, or proximity effect, is not so severe, whereas the resistance of the package and PCB traces shows that the skin effect is frequency-dependent. As shown in Figure 5.30, the resistance of the package trace is also higher than that of the PCB trace. This is because the cross section of the package trace is smaller, resulting in a smaller area for current. In the on-chip wire case, both the inductance and capacitance vary, even at extremely higher frequencies, indicating that the transmission line model is not a good choice for modeling on-chip wires.

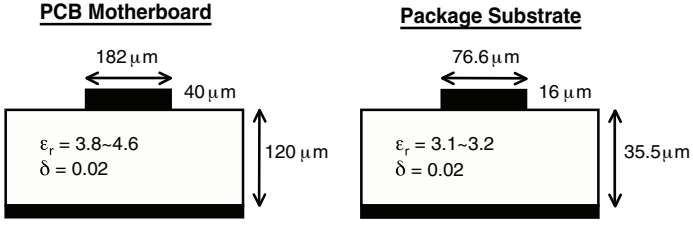


Figure 5.29 Microstrip Geometries: PCB Motherboard and Package Traces

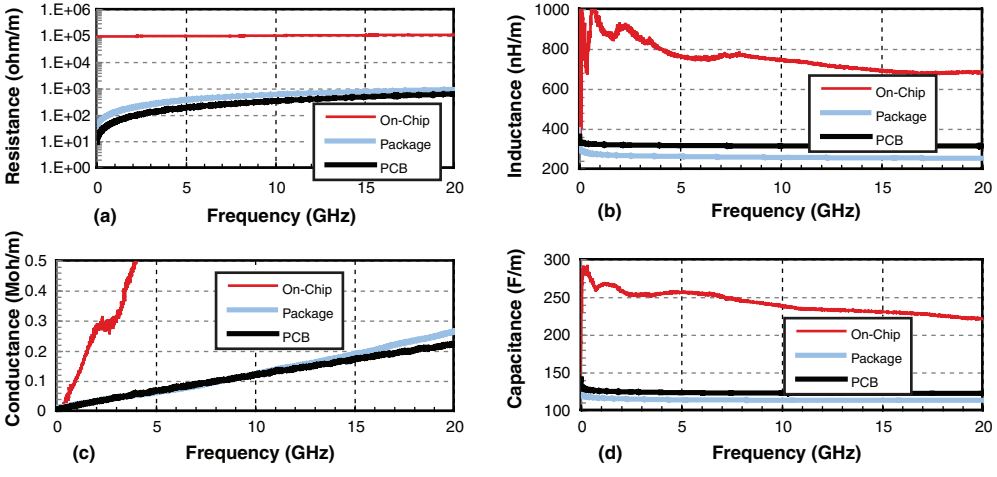


Figure 5.30 RLGC Parameters of On-Chip, Package, and PCB Traces

A more meaningful comparison can be achieved using the propagation constant function and characteristic impedance of the transmission parameters. Because we are interested in a qualitative comparison, the following simplified definitions for the propagation constant and characteristic impedance are used:

$$\gamma(\omega) = \sqrt{Z(\omega)Y(\omega)} \cong \alpha_c(\omega) + \alpha_d(\omega) + \beta(\omega) \tag{5.57a}$$

$$Z_c(\omega) = \sqrt{Z(\omega)/Y(\omega)} \cong \sqrt{L(\omega)/C(\omega))} \tag{5.57b}$$

where

$$\alpha_c(\omega) \cong R(\omega)\sqrt{C(\omega)/L(\omega)} \tag{5.57c}$$

$$\alpha_d(\omega) \cong G(\omega)\sqrt{L(\omega)/C(\omega)} \tag{5.57d}$$

$$\beta(\omega) = \omega\sqrt{L(\omega) \cdot C(\omega)}. \tag{5.57e}$$

$\alpha_c(\omega)$ and $\alpha_d(\omega)$ represent the attenuation due to conductor loss and dielectric loss, respectively. $\beta(\omega)$ represents phase delay (see Figure 5.31). $\alpha_c(\omega)$ is almost flat for the on-chip wire, because DC loss dominates. $\alpha_c(\omega)$ for the package trace is larger than the PCB trace, as noted previously. On the other hand, $\alpha_d(\omega)$ for the package and PCB is similar, because the dielectric loss is not a strong function of conductor geometry. The characteristic impedance for the package trace saturates to the steady value more slowly than for the PCB trace. This is because the internal inductance plays a more significant role in package traces, due to their smaller conductor geometry.

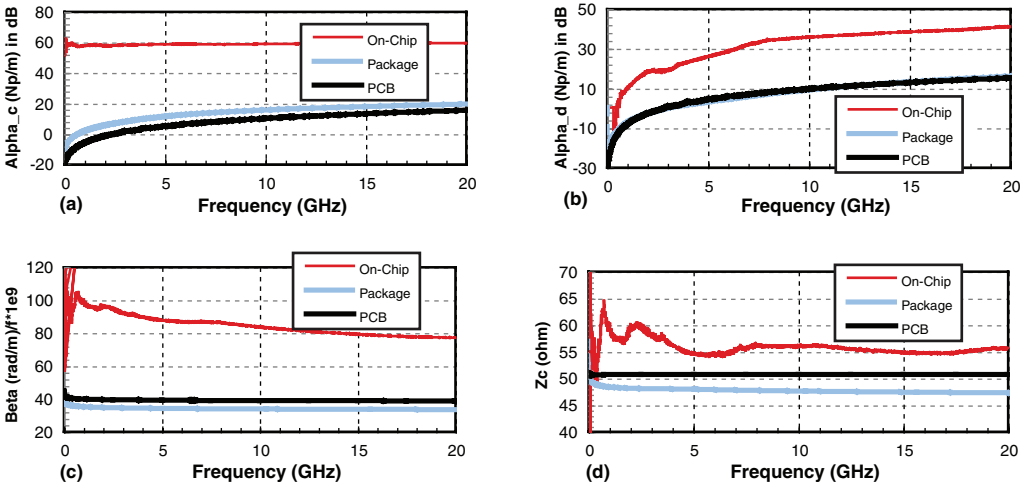


Figure 5.31 α_c, α_d, β, and Z_c of On-Chip, Package, and PCB Traces

5.7 Summary

This chapter provides a review of basic transmission-line theory. It discusses the fundamental physical properties of RLGC parameters and crosstalk phenomenon. It also discusses and compares popular transient simulation techniques for lossy dispersive lines (and also covers the pitfalls of these simulation methods). It presents a methodology with which to derive a frequency-dependent RLGC parameter, along with using S-parameter measurements. This chapter introduces a DC accuracy enhancement method, based on time-domain TDR response. On-chip wire modeling requires special attention and a segmented π model provides sufficient accuracy. Finally, the chapter covers the properties of different transmission lines: on-chip wires, package traces, and PCB traces.

References

1. J. Gruodis and C. S. Chang, "Coupled lossy transmission line characterization and simulation," *IBM Journal of Research and Development*, vol. 25, pp. 25–41, Jan. 1981.
2. W. Kim, J.-H. Kim, D. Oh, and C. Yuan, "Implementation of broadband transmission line models with accurate low-frequency response for high-speed system simulations," presented at the IEC DesignCon, Santa Clara, CA, Feb. 2006.
3. M. Horno, R. L. Mesa, F. Medina, and R. Marques, "Quasi-TEM analysis of multilayered multiconductor coplanar structures with dielectric and magnetic anisotropy including substrate losses," *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-38, no. 8, pp. 1059–1068, Aug. 1990.
4. F. L. Mesa, G. Cano, F. Medina, R. Margues, and M. Horno, "On the quasi-TEM and full-wave approaches applied to coplanar multistrip on lossy dielectric layered media," *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-40, no. 3, pp. 524–531, Mar. 1992.
5. R. K. Hoffmann, *Handbook of Microwave Integrated Circuits*, Artech House, 1987.
6. G. L. Matthaei, K. Kiziloglu, N. Dagli, and S. I. Long, "The nature of the charges, currents, and fields in and about conductors having cross-sectional dimensions of the order of a skin depth," *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-38, no. 8, pp. 1031–1036, Aug. 1990.
7. K. S. Oh, "Efficient modeling of interconnects and capacitive discontinuities in high-speed digital circuits," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1995.
8. K. D. Marx, "Propagation modes, equivalent circuits, and characteristic terminations for multiconductor transmission lines with inhomogeneous dielectrics," *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-21, no. 7, pp. 450–457, Jul. 1973.
9. W. Dally and J. Poulton, *Digital System Engineering*, Cambridge University Press, 1998.

10. J. Zerbe, P. Chau, C. Werner, W. Stonecypher, H. J. Liaw, G. J. Yeh; T. P. Thrush, S. Best, and K. Donnelly, "A 2 Gb/s/pin 4-PAM parallel bus interface with transmit crosstalk cancellation, equalization, and integrating receivers," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2001, pp. 66–67.
11. C. Pelard, E. Gebara, A. J. Kim, M. Vrazel, F. Bien, Y. Hur, M. Maeng, S. Chandramouli, C. Chun, S. Bajekal, S. Ralph, B. Schmukler, V. Hietala, and J. Laskar, "Realization of multigigabit channel equalization and crosstalk cancellation integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 10, pp. 1659–1669, Oct. 2004.
12. Y. Hur, M. Maeng; C. Chun, F. Bien, H. Kim, S. Chandramouli; E. Gebara, and J. Laskar, "Equalization and near-end crosstalk (NEXT) noise cancellation for 20 Gb/s 4-PAM backplane serial I/O interconnects," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 1, pp. 246–255, Jan. 2005.
13. J. Buckwalter and A Hajimiri, "Cancellation of crosstalk-induced jitter," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 621–632, Mar. 2006.
14. J. Wilson and D. Oh, "Active crosstalk cancellation for next-generation single-ended memory interfaces," in *Proceedings of Electronic Components and Technology Conference*, Lake Buena Vista, FL, Jun. 2011, pp. 202–208.
15. J. E. Schutt-Aine and R. Mittra, "Scattering parameter transient analysis of transmission lines loaded with nonlinear terminations," *IEEE Transactions on Microwave Theory and Techniques,* vol. 36, pp. 529–536, Mar. 1988.
16. A. R. Djordjevic, T. K. Sakar, and R. F. Harrington, "Analysis of time response of lossy multiconductor transmission line networks," *IEEE Transactions on Microwave Theory and Techniques,* vol. 35, pp. 898–908, Oct. 1987.
17. J. R. Griffith and M. S. Nakhla, "Time-domain analysis of lossy coupled transmission lines," *IEEE Transactions on Microwave Theory and Techniques,* vol. 38, pp. 1480–1487, Oct. 1990.
18. D. B. Kuznetsov and J. E. Schutt-Aine, "Optimal transient simulation of transmission lines," *IEEE Transactions on Circuits and Systems,* vol. 43, no. 2, pp. 110–121, Feb. 1996.
19. K. S. Oh, "Accurate transient simulation of transmission lines with the skin effect," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 19, pp. 389–396, Mar. 2000.
20. F. H. Branin Jr., "Transient analysis of lossless transmission lines," in *Proceedings of the IEEE*, vol. 55, 1967, pp. 2012–2013.
21. A. E. Ruehli, Ed, *Circuit Analysis, Simulation, and Design*, part 1, North-Holland, 1986.

22. S Grivet-Talocia, "Delay-based macromodels for long interconnects via time-frequency decompositions," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 199–202.
23. A. Chinea, P. Triverio, and S Grivet-Talocia, "Compact macromodeling of electrically long interconnects," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference,* Oct. 2008, pp. 199–202.
24. A. Semlyen and A. Dabuleanu, "Fast and accurate switching transient calculations on transmission lines with ground return using recursive convolutions," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-94, no. 3, pp. 561–569, Mar. 1975.
25. B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain response by vector fitting," *IEEE Transactions on Power Delivery,* vol. 14, no. 3, pp. 1052–1061, July 1999.
26. B. Gustavsen and C. Heitz, "Modal vector fitting: a tool for generating rational models of high accuracy with arbitrary terminal conditions," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 4, pp. 664–672, Nov. 2008.
27. K. S. Oh and J. E. Schutt-Aine, "An efficient implementation of surface impedance boundary conditions for the finite-difference time-domain method," *IEEE Transactions on Antennas and Propagation*, vol. 43, pp. 660–666, Jul. 1995.
28. M. La Scala, S. Sblendorio, and R. Sbrizzai, "Parallel-in-time implementation of transient stability simulations on a transputer network," *IEEE Transactions on Power Systems,* vol. 9, pp. 1117–1125, May 1994.
29. T. K. Tang, M. S. Nakhala, and R. Griffith, "Analysis of lossy multiconductor transmission lines using the Asymptotic Waveform Evaluation technique," *IEEE Transactions on Microwave Theory and Techniques,* vol. 39, pp. 2107–2116, Dec. 1991.
30. W. T. Beyene and J. E. Schutt-Aine, "Accurate diode forward recovery reverse model using asymptotic waveform evaluation techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 16, no. 12, pp. 1447–1453, Dec. 1997.
31. J. Kim and D. H. Han, "Hybrid method for frequency-dependent lossy coupled transmission line characterization and modeling," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2003, pp. 239–242.
32. W. Kim, J. Kim, D. Oh, and C. Yuan, "S-parameters based transmission line modeling with accurate low-frequency response," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 79–82.

33. J. Kim, D. Oh, and W. Kim, "Accurate characterization of broadband multiconductor transmission lines for high-speed digital systems," *IEEE Transactions on Advanced Packaging*, vol. 33, pp. 857–867, Nov. 2010.

34. D. Gope, A. Ruehli, and V. Jandhyala, "Solving low-frequency EM-CKT problems using the PEEC method," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2005, pp. 351–354.

35. J. Zhao and W. C. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 10, pp. 1635–1645, Oct. 2000.

36. J. Zhu and D. Jiao, "A rigorous method for fundamentally eliminating the low-frequency breakdown in full-wave finite-element-based analysis: combined dielectric-conductor case," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2011, pp. 69–72.

37. *HSPICE Signal Integrity Guide*, version 2008.09, Synopsis.

38. A. Deutsch, G. Arjavalingam, and G. V. Kopcsay, "Characterization of resistive transmission lines by short-pulse propagation," *IEEE Microwave and Guided Wave Letters*, vol. 2, pp. 25–27, Jan. 1992.

39. W. Kim and M. Swaminathan, "Simulation of lossy package transmission lines using extracted data from one-port TDR measurements," *IEEE Transactions on Advanced Packaging*, vol. 28, no. 4, pp. 736–744, Nov. 2004.

40. C. L. Liu and J. W. S. Liu, *Linear Systems Analysis*, McGraw-Hill, 1975.

41. R. Schaefer, "Discussing the limitations and accuracies of time and frequency domain analysis of physical layer devices," presented at the IEC DesignCon, Santa Clara, CA, 2005.

42. R. B. Marks, "A multiline method of network analyzer calibration," *IEEE Transactions on Microwave Theory and Techniques*, vol. 39, pp. 1205–1215, Jul. 1991.

43. D. F. Williams, and R. B. Marks, "Accurate transmission line characterization," *IEEE Microwave and Guided Wave Letters*, vol. 3, no. 8, pp. 247–249, Aug. 1993.

44. T. Winkel, L. S. Dutta, H. Grabinski, and E. Grotelǘshen, "Determination of the propagation constant of coupled lines on chips based on high frequency measurements," in *Proceedings of IEEE Multi-Chip Module Conference*, Feb. 1996, pp. 99–104.

45. T. Winkel, L. S. Dutta, and H. Grabinski, "An accurate determination of the characteristic impedance matrix of coupled symmetrical lines on chips based on high frequency S-parameter measurements," in *IEEE 49th ARFTG Conference Digest*, Jun. 1997, pp. 223–226.

46. A. Deutsch, H. H. Smith, G. V. Kopcsay, D. C. Edelstein, and P. W. Coteus, "On-chip wiring design challenges for GHz operation," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 1999, pp. 45–48.
47. P. J. Restle, "High speed interconnects: a designer's perspective," in *ICCAD Tutorial: Interconnect High Speed Design: Problems, Methodologies and Tools*, Nov. 1998.
48. X. Qi, A. Gyure, Y. Luo, S. Lo, M. Shahram, and K. Singhal, "Simulation of interconnect inductive impact in the presence of process variations in 90 nm and beyond," *IEEE Electron Device Letters*, vol. 27, no. 8, pp. 696–698, Aug. 2006.
49. J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz, "A 14-mW 6.25-Gb/s transceiver in 90-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2745–2757, Dec. 2007.
50. X. Qi, J.-H. Kim, L. Yang, R. Schmitt, and C. Yuan, "Compact on-chip wire models for the clock distribution of high-speed I/O interfaces," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2008, pp. 235–238.
51. A. E. Ruehli, "Inductance calculations in a complex integrated circuit environment," *IBM Journal of Research and Development*, pp. 470–481, Sep. 1972.
52. M. Kamon, M. J. Tsuk, and J. K. White, "FASTHENRY: a multipole-accelerated 3-D inductance extraction program," *IEEE Transactions on Microwave Theory and Techniques,* vol. 42, pp. 1750–1758, Sep. 1994.

This page intentionally left blank

Part II

Analyzing Link Performance

This page intentionally left blank

CHAPTER 6

Channel Voltage and Timing Budget

Dan Oh and Chuck Yuan

Previous chapters have discussed the accurate modeling of passive channel structures. The following chapters focus on the analysis and simulation aspects of the channel. The ultimate goal of signal integrity (SI) design is to determine whether the I/O channel under study can meet the target performance requirements under the worst-case operating conditions. To that end, one needs to perform analysis beyond the traditional modeling of devices and passive channel. A few key questions to answer are:

- What are the adequate voltage and timing budgets for a target system to ensure reliable link performance while avoiding system overdesign?
- How do we ensure that the high-volume production system has sufficient margin?
- How can we account for various device jitter and noise components, in addition to the traditional passive channel and output driver?
- How do we model different clocking topologies and their associated jitter?

Although these questions are important to modern high-speed I/O analysis, little literature is available. This chapter through Chapter 12 present materials that help answer these questions. The following is a brief overview.

Development of I/O systems starts with a test-chip design that typically uses an ideal system environment. For instance, printed circuit board (PCB) trace impedance is well controlled (to eliminate signal reflections), and sufficient numbers of bypassing capacitors are placed to minimize power-supply noise. However, one must consider various manufacturing variations to ensure success in high-volume production. Chapter 7, "Manufacturing Variation Modeling," describes a Design of Experiment (DoE) method, based on the Taguchi method, to account for manufacturing variations in channel simulation. The Taguchi design of experiment also enables

one to optimize the channel design. For instance, one can use Taguchi analysis to maximize the signal eye opening (voltage margin) by optimizing the driver's strength and termination.

Traditional SI analysis tends to focus on characterizing the impact of deterministic jitter due to passive channel attenuation and dispersion, and output driver slew rate and non-linearity. Typical deterministic jitter includes inter-symbol interference (ISI), crosstalk, and signal skew. However, a significant part of link jitter comes from device jitter, such as phase lock loop (PLL) jitter, clock buffer jitter, duty cycle distortion (DCD), and reference clock jitter. As a result, the link simulations must capture the link margin loss due to these jitter components to predict link voltage and timing margin. Chapter 8, "Link BER Modeling and Simulation," presents a statistical link simulation flow that simulates typical deterministic jitter sources, as well as active device jitter (including both deterministic and random jitter).

The statistical link simulator described in Chapter 8 is based on the assumption that the system or channel is linear and time invariant. Therefore, strictly speaking, it cannot be applied to non-linear drivers with time-varying on-die terminations (ODTs). It also assumes that the data pattern is random and uniform, so that it cannot handle data coding. Chapter 9, "Fast Time-Domain Channel Simulation Techniques," describes a fast time-domain channel simulation method, used in conjunction with the statistical simulator, to overcome the aforementioned limitations in the statistical method. The time-domain method is particularly useful for single-ended signaling interfaces, because it can handle simultaneously switching output (SSO) noise. Chapter 9 also reviews the basics of the AMI (Algorithm Model Interface) model proposed by the IBIS standard.

The clock signal contributes one of the most dominant jitter components. Chapter 10, "Clock Models in Link BER Analysis," describes the basic mechanism of clock jitter, the modeling methodology, and various types of clocking topologies, as well as clock jitter amplification and tracking. Chapter 10 also extends the statistical approach described in Chapters 8 and 9 to include jitter tracking. Chapter 11, "Overview of Power Integrity Engineering," discusses the power supply noise-modeling budget, as device jitter is often due to supply noise. Finally, Chapter 12, "SSN Modeling and Simulation," describes how the impact of intersymbol interference (ISI), or SSO, noise is mitigated using data coding.

This chapter discusses how channel voltage and timing budgets are defined. Balancing the channel voltage and timing (VT) budgets is one of the most critical tasks in designing high-speed I/O links. VT budgets must be appropriately assigned to individual channel components, such as the transmitter, receiver, and passive channel. An improperly balanced VT budget could cause a system failure or result in an overly stressed specification for a particular subcomponent, causing unnecessary yield loss and higher system costs.

However, very few engineers have the opportunity to define a channel's VT budget, unless they are involved with a standard-setting body, or working on a proprietary standard. Instead, much of practical signal-integrity analysis focuses on modeling and simulating a system to determine whether the designed system meets the specifications of a given standard. Different I/O interfaces have distinct electrical specifications, and they require different simulation setups and

approaches. Even without the benefit of setting the channel VT budget, a thorough understanding of how voltage and timing specifications are defined makes a system design trade-off possible. For example, one can trade channel ISI (that is, impedance variation) for crosstalk, if one knows how much channel timing error is allocated.

The following sections describe commonly used voltage and timing equations, explain the basic concepts behind them, and address the limitations associated with these equations. (Specifically, the equation-based timing budget cannot account for the interaction between different components, and the spectrum of various noise components.) Finally, a statistical link simulator is proposed to close a timing budget of a high-speed I/O interface.

6.1 Timing Budget Equation and Components

The simplest way to define the channel timing budget is to linearly sum up the individual jitter components from the transmitter, receiver, and passive channel. The channel timing budget is balanced if the sum is less than the symbol time or UI (unit interval). The underlying assumption in this simple approach is that all jitter components in the channel are linearly independent. Using the worst-case values for all the jitter components in the equation guarantees that the system will function in a worst-case scenario. Consequently, using such an equation to define the jitter specifications or budgets for each component of the channel is convenient. These budgets can, in turn, be translated into a device (transmitter and receiver) AC specification, and listed in a device datasheet. Because of its simplicity, this methodology was widely adopted for many years, even though such a methodology is known to be pessimistic.

Lower data rate systems, such as DDR and RDRAM memory systems, only consider deterministic jitter (DJ) components in their channel timing equations. When the data rate reaches multi-gigabits per second, such as PCI Express (PCIe), FlexIO, or XDR memory systems, the timing budget equation is extended to include random jitter (RJ). Specifically, the total peak-to-peak jitter is determined by linearly summing the DJs, and the root-sum-square of the RJs, as follows:

$$TJ \equiv TJ_{DJ} + TJ_{RJ} \equiv \sum DJ + 2Q_{BER}\sqrt{\sum \sigma^2}. \tag{6.1}$$

In (6.1), Q_{BER} is the value of the Q-function at the target bit error rates (BER), DJ represents the deterministic jitter components, and σ represents the random jitter (RMS) components. The $Q(x)$ function is the area under the normal distribution function, from x to infinity, and is related to the complementary error function, $erfc(x)$, as determined by

$$Q(x) = \frac{1}{2} erfc\left(\frac{x}{\sqrt{2}}\right). \tag{6.2}$$

A Q_{BER} of 7.03 at $BER = 10^{-12}$ is commonly used as a target value for links with error detection schemes. For memory systems without error detection, use a much lower bit error rate. For instance, a Q_{BER} of 9.26 guarantees a BER of 10^{-20}. As the data rate continues to increase, use lower BER target values to maintain a constant error rate for a fixed time. Table 6.1 lists the Q_{BER} for various BER targets.

Table 6.1 Target BER vs. Q_{BER}

BER	**10^{-1}**	**10^{-2}**	**10^{-3}**	**10^{-4}**	**10^{-5}**
Q_{BER}	1.28155	2.32635	3.09023	3.71902	4.26489
BER	**10^{-6}**	**10^{-7}**	**10^{-8}**	**10^{-9}**	**10^{-10}**
Q_{BER}	4.75342	5.19934	5.61200	5.99781	6.36134
BER	**10^{-11}**	**10^{-12}**	**10^{-13}**	**10^{-14}**	**10^{-15}**
Q_{BER}	6.70602	7.03448	7.34880	7.65063	7.94135
BER	**10^{-16}**	**10^{-17}**	**10^{-18}**	**10^{-19}**	**10^{-20}**
Q_{BER}	8.22208	8.49379	8.75729	9.01327	9.26234
BER	**10^{-21}**	**10^{-22}**	**10^{-23}**	**10^{-24}**	**10^{-25}**
Q_{BER}	9.50502	9.74179	9.97305	10.19916	10.42045
BER	**10^{-26}**	**10^{-27}**	**10^{-28}**	**10^{-29}**	**10^{-30}**
Q_{BER}	10.63722	10.84974	11.05823	11.26293	11.46402

A variety of channel and device effects can introduce deterministic and random jitter (DJ and RJ) into equation (6.1). (Chapter 2, "High-Speed Signaling Basics," provides a detailed description of various jitter characteristics.) For example, DJ includes duty-cycle-distortion (DCD), intersymbol interference (ISI), sinusoidal or periodic jitter (PJ), crosstalk, simultaneously switching outputs (SSO) jitter, and electromagnetic interference (EMI). DJ comes from all components of the link, including the transmitter, receiver, and passive channel. In contrast, random jitter (or RJ) comes only from the transmitter and receiver devices, and is primarily the result of thermal noise, shot noise, and flicker noise.

6.2 Fibre Channel Dual-Dirac Model

Equation (6.1) is equivalent to the double-delta or dual-Dirac model used in the Fibre Channel jitter specification [1]. The dual-Dirac model makes two assumptions: First, that the deterministic jitter is represented by two delta PDFs, separated by the total peak-to-peak DJ values. Second, the random jitter is modeled by the Gaussian distribution with the effective standard deviation σ, which is the root-mean square sum of the individual σs, as shown in (6.1). Figure 6.1 illustrates the dual-Dirac model.

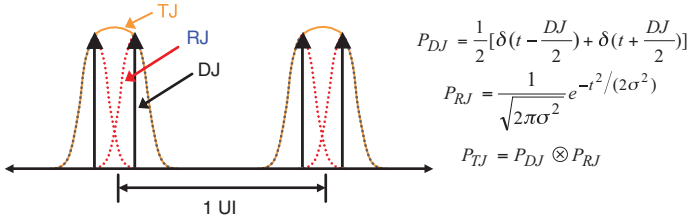


Figure 6.1 Fibre Channel Dual-Dirac Model

To demonstrate that (6.1) is equivalent to the double-delta model, consider two cases at a 10Gb/s data rate: One case with DJ = 52ps and σ = 2ps, and the other with DJ = 9.6ps and σ = 5ps. At BER = 10^{-12}, both cases result in 80ps, the total jitter based on (6.1). The bathtub curves for these two cases, using the log scale of BER and Q-space scale, are shown in Figure 6.2. The Q-function scaled plot shows a clear linear variation near the lower BER region. As shown in Figure 6.2, the dual-Dirac model predicts a 20% UI margin, as expected. Based on Figure 6.2, one might erroneously conclude that the case where DJ = 9.6ps and σ = 5ps is the better system, because it shows better margin for the BER range above 10^{-12}. However, in practice, this case is less desirable because it results in a smaller margin for the BER range below 10^{-12}. This example also demonstrates that an accurate DJ and RJ decomposition is important for system margin analysis.

Because the dual-Dirac model approximates the DJ distribution using simple delta functions, it ignores the detail distribution of DJ. This approximation results in pessimistic jitter estimation [2] [3]. To demonstrate this limitation, consider the four different probability density functions (PDFs) shown in Figure 6.3. The variation of TJ due to different DJ distributions is illustrated by Madden, Chuck, and Oh [3]. Figure 6.4 shows the total PDFs after convolving with 1ps RJ and 5ps DJ, and 1ps RJ and 20ps DJ, respectively. As shown in these figures, the double-delta model results in the worst-case deterministic jitter. This pessimism increases as the DJ distribution widens.

Figure 6.5 compares the resulting BER curves for the 1ps RJ and 20ps DJ case. The BER curve is calculated using the complementary error function for the double-delta (DD) case; numerical integration is used for the other cases. As expected, the double-delta case shows the most pessimistic curve. The discrepancy between the double-delta distribution and the other distributions increases as DJ increases. Note that, in general, the different DJ distributions have large differences in the high BER region. And while the difference is smaller in the low BER region (where RJ is dominant), a significant difference still exists. This is due to the fact that when RJ is convolved with DJ, the resulting function is no longer a conventional Gaussian normal distribution function, even at the tail area of the convolved function.

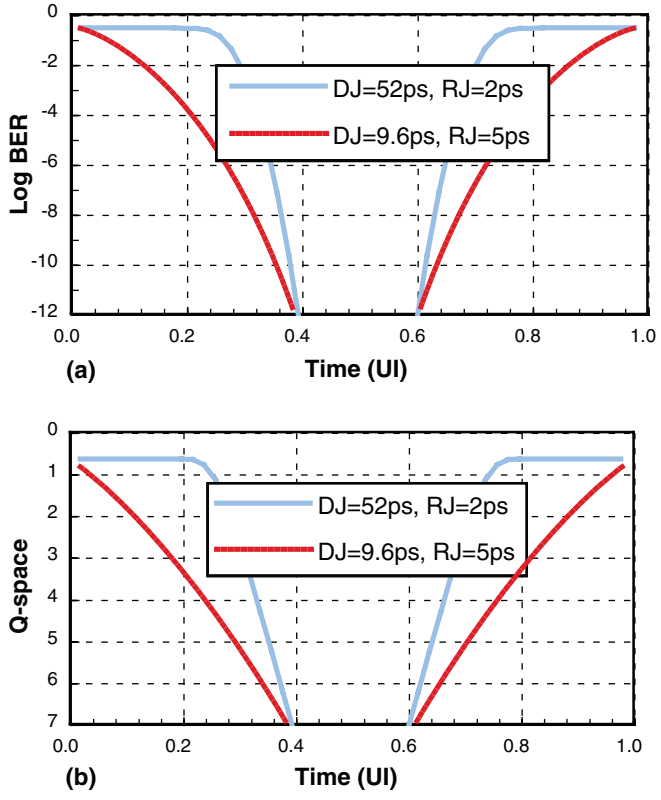


Figure 6.2 Timing Bathtubs for Two Different DJ and RJ Values Using Fibre Channel Dual-Dirac Model: Using Log(BER) Scale and Using Q-Space Scale

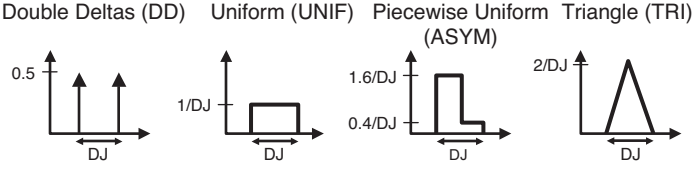


Figure 6.3 Four Test-Case DJ PDF Distributions: Double Deltas (DD), Uniform (UNIF), Piecewise Uniform (ASYM), and Triangle (TRI)

The popular tail fitting method [2] [3] based on the double-delta model always assumes the normal distribution, and produces somewhat different RJ numbers [3]. Table 6.2 lists the results of tail fitting the bathtub curves in Figure 6.5. For curve fitting data, BER values between 10^{-5} and 10^{-10} are used. Because the approximation is based on the pessimistic double-delta DJ representation, it typically results in an underestimation of DJ, as shown in the table. More interesting

is that it overestimates RJ. This discrepancy increases more for cases where DJ distribution is tight around the center.

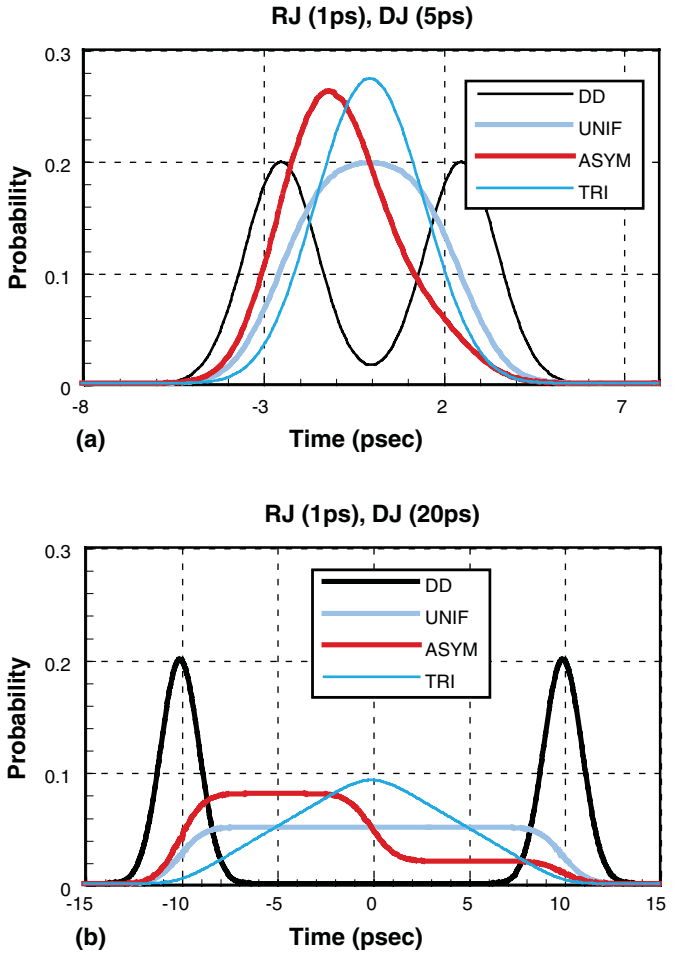


Figure 6.4 The Final PDFs for the Four Distributions Shown in Figure 6.3: Using 1ps RJ and 5ps DJ and Using 1ps RJ and 20ps DJ

Table 6.2 DJ and RJ Estimation of the Bathtub Curves in Figure 6.5 Based on the Tail-Fitting Method

	DD	UNIF	ASYM (left)	ASYM (right)	TRI
DJ	20ps	16.1ps	15.3ps	15.3ps	15.4ps
RJ	1ps	1.145ps	1.193ps	1.228ps	1.181ps

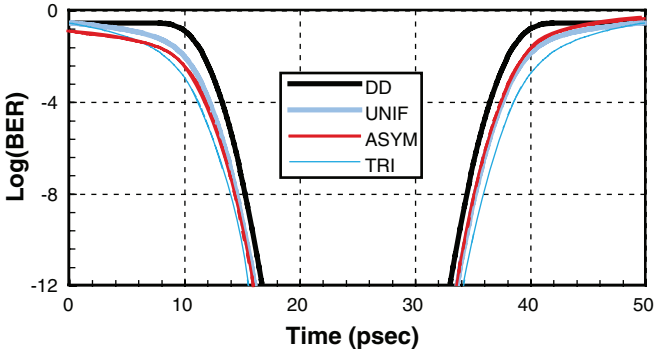


Figure 6.5 The Final Bathtubs for the Four Distributions Shown in Figure 6.3

The asymmetric DJ distribution results in an asymmetric bathtub curve, as shown in Figure 6.5. The tail-fitting method produces two different RJ values; one for the left side and one for the right side of the bathtub curve (see Table 6.2). However, only one RJ value is in the original distribution. In practice, slightly different RJ values are observed for the systems with the same RJ source but with the different DJ distributions. The main cause of this discrepancy could be the numerical difficulty associated with extracting RJ and DJ values from the total jitter. The tail-fitting method illustrates this issue in Table 6.2. Finally, it is also interesting to note that the asymmetric bathtub approaches the uniform case in the low BER region, because, for both cases, the normal distribution is integrated over the constant interval.

6.3 Component-Level Timing Budget

This section provides a specific example of a channel timing budget. As stated earlier, at low data rates, the linear sum approach is used for individual components, as follows:

$$TJ = TJ_{TX} + TJ_{RX} + TJ_{CH} + TJ_{CLK} = (DJ_{TX} + 2Q_{BER}\sigma_{TX}) + (DJ_{RX} + 2Q_{BER}\sigma_{RX}) + DJ_{CH} + (DJ_{RX} + 2Q_{BER}\sigma_{CLK}) \quad (6.3)$$

where the timing budget is broken into subcomponents, such as a passive channel (TJ_{CH}), transmitter (TJ_{TX}), receiver (TJ_{RX}), and clock (TJ_{CLK}). Notice that the passive channel does not have a random jitter contribution. Equation (6.3) results in a pessimistic timing budget. For high-speed applications, the root-sum-square of random jitter in (6.1) is preferred:

$$TJ = (DJ_{TX} + DJ_{RX} + DJ_{CH} + DJ_{CLK}) + 2Q_{BER}\sqrt{\sigma_{TX}^2 + \sigma_{RX}^2 + \sigma_{CLK}^2}. \quad (6.4)$$

The first-generation PCIe specification (2.5Gb/s) is based on (6.3), whereas the second-generation PCIe specification (5Gb/s) is based on (6.4). Table 6.3 compares these two

approaches, based on the PCIe2 specification [4] [5]. Equation (6.3) clearly predicts the pessimistic total jitter, as compared to (6.4). As will be shown in the following section, even (6.4) predicts somewhat pessimistic final jitter values. On the other hand, although (6.4) results in a better timing budget, it requires the extraction of the DJ and RJ components for each component, which can be quite an expensive process. In practice, this extraction is only done during the system characterization stage. The individual device timing testing for a production line is done using (6.3) with a limited data length (for instance, $BER = 10^{-6}$ or larger).

Table 6.3 Comparison of PCIe Gen2 Timing Budget for Data-Clock Architecture at 5Gb/s Using the Linear Sum and Root-Sum-Square Approaches (TJ is at 10^{-12})

(ps)	DJ	RJ	Linear Sum	Root-Sum-Square
Transmitter	30	1.4	50	
Receiver	48	1.4	68	
Channel	58	0	58	
Reference Clock	0	4	56	
Effective Sigma		4.46		
Total	136	63	232	199

So far, the timing specifications in (6.3) or (6.4) are based on the system-level jitter impact. To make these components meaningful for testing, the final device-level specification needs to be carefully defined, based on the system-level specifications. The component-level timing specification must consider the difference in the device impact on the system and on the testing environment. Due to this difference, the component-level specification is often larger than the system-level specification. Most of the device vendor specification is based on the component-level values. The system vendor needs to account for this difference when building the overall system. For standard I/O interfaces, the standards committee (such as PCIe) often provides a reference design guide, so system vendors do not need to worry about this difference.

6.4 Pitfalls of Timing Budget Equation

The timing budget equation in (6.1) is based on several assumptions. In this section, these assumptions, and their impacts, are discussed. As mentioned in the previous section, one of the assumptions is that the double-delta DJ approximation ignores the detailed DJ distribution, and predicts a pessimistic system margin. Another crucial assumption in (6.1) is that all jitter components are independent. This independent assumption results in

- Linear summation of deterministic jitter components
- Root-sum-of-square of random jitter components
- Linear sum of the above two sums

Let's take a closer look at the impact of these individual approximations. The linear sum of deterministic components results in pessimistic results. For instance, the crosstalk effect, jitter induced by SSO noise, and channel ISI are not truly independent. All of these effects are correlated with the channel or package resonance. Although the final system response, in terms of voltage, can be calculated by linearly adding individual response due to crosstalk, SSO noise, and victim ISI response, the jitter amount cannot be linearly added. Imagine a case where the victim line experiences crosstalk noise that has an opposite sign than the ISI reflection, resulting in reduction in the final jitter. Thus, adding peak-to-peak deterministic jitter values could be rather pessimistic. As discussed in the previous section, deterministic jitter is often specified for individual subcomponents of the link, such as the transmitter, receiver, and passive channel. The worst-case deterministic jitter, for each of these components, does not add linearly. For example, the worst-case data pattern for the transmitter jitter, in general, does not correspond to worst-case channel or receiver jitter.

Similarly, the root-sum-of-square of random jitter components ignores the complete spectrum of random jitter components and neglects the correlation between the various random jitter components. This approximation again results in pessimistic jitter values.

Random jitter in the clock signal requires special attention. Most systems use a common clock generator for both transmitter and receiver. This causes some of the low-spectrum jitter that appears at the transmitter to be tracked by the receiver. So, the spectrum content of the clock jitter needs to be considered carefully to avoid pessimistic budgets. Chapter 10 provides a detailed discussion of clock jitter.

In contrast to the previous two approximations, the last approximation of the linear sum of DJ and RJ results in optimistic results. Until recently, it was not well known that the channel correlates (colors) the component/device-level jitter distinctively, depending on the noise source. In particular, the transmitter device jitter is amplified by the ISI of the passive channel. Chapter 10 discusses this jitter amplification in more detail.

The net effect, due to aforementioned approximations, tends to produce a pessimistic timing budget. To demonstrate this, consider a PCIe link timing budget using a statistical link simulation tool. The jitter specification at 5Gb/s for a CDR-based PCIe Gen2 system was presented in Table 6.3. By adding the deterministic components specifications linearly, and the random components statistically, the budget table predicts almost zero link margin at 10^{-12} BER. A more accurate link margin can be determined using a statistical link simulator with all jitter component models at the maximum allowed value [6].

Figure 6.6 shows the test system. TX DJ is set equal to 0.15UI, where 0.1UI is assumed to be due to DCD. TX equalization is set to 6dB. Assuming that the CDR tracks low-frequency RJ, only high frequency is included in TX RJ (7mUI RMS) with a flat spectrum from 1.5MHz to

100MHz. Moreover, reference clock jitter with 20mUI RMS and a flat spectrum up to 2.5 GHz is used. The TX PLL transfer function is set to a 16-MHz bandwidth, with 3.0dB peaking (according to the PCIe Gen2 [4] specification). As shown in Figure 6.7, the channel alone generates 0.25UI DJ, versus the spec of 0.29UI. Rx has no linear EQ, where the CDR model is adjusted so that, when simulated with the TX and the channel, the CDR dithering range is roughly 0.24UI (which mimics the RX DJ specification). Additionally, the RX has 7mUI RJ. The sum of DJs is 0.64UI, which is 0.04UI better than the specification value in Table 6.3. As a result, one would expect that the link has a 0.04UI timing margin at BER $= 10^{-12}$.

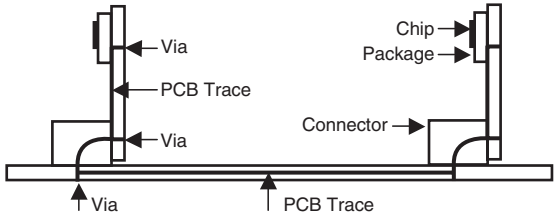


Figure 6.6 Typical PCI Express System for Long Channel Application

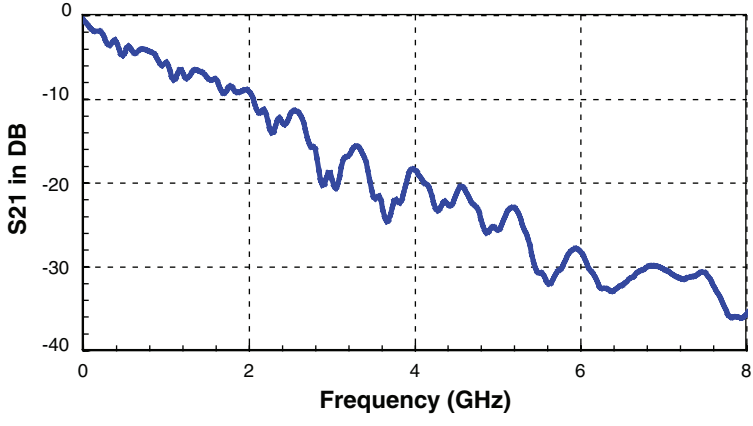


Figure 6.7 S-Parameter of the PCI Express Channel

The BER bathtub predicted by a statistical simulator is shown in Figure 6.8. The simulated timing BER bathtub is asymmetric due to the asymmetric eye shape. Rather than the expected 0.04UI timing margin at BER $= 10^{-12}$, the simulation shows an additional 0.05UI timing margin. This demonstrates that (6.4) gives pessimistic results. However, the bathtub curve is generated based on the dual-Dirac (Double Delta) model in Table 6.3 (with DJ = 0.68UI and RJ = 0.022UI), zero timing margin at BER $= 10^{-12}$ is obtained (see Figure 6.8). This shows the

close relationship between the dual-Dirac model and (6.1). Finally, if the simulated bathtub curve is fitted with the dual-Dirac model, as shown in Figure 6.9, the effective link DJ and RJ are 0.355UI and 0.039UI, respectively. Significantly smaller link DJ, and slightly larger RJ, are estimated. This is due to the complex interactions between the various jitter components in the link, further demonstrating the pitfalls of the current budgeting approach, based on a timing equation.

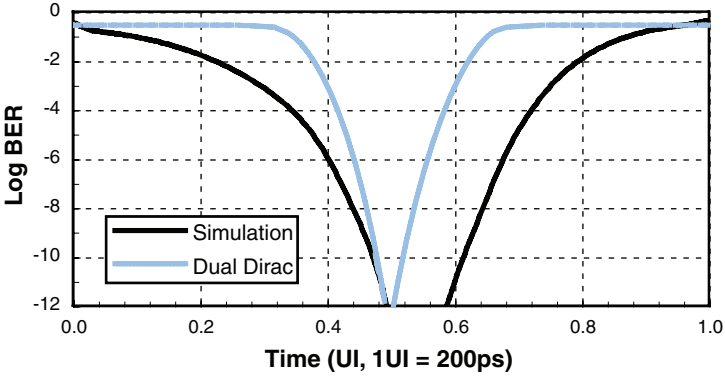


Figure 6.8 Bathtub Curves Using a Statistical Simulation vs. the Dual-Dirac Model

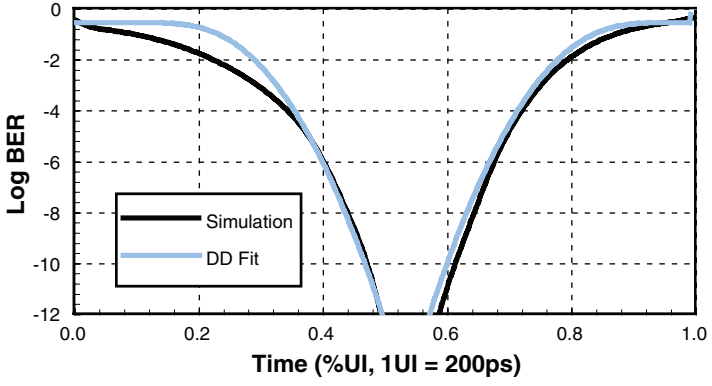


Figure 6.9 Dual-Dirac Approximation to Simulation Data

6.5 Voltage Budget Equations and Components

The voltage budget equation is much simpler than the timing equation, and is often expressed as

$$V_{RX,Deadband} + 2Q_{BER}V_{RX,RJ} \leq (V_{TX,Swing} - V_{ISI} - V_{XTK} - V_{SSO}) \tag{6.5}$$

where $V_{RX,Deadband}$ is the receiver deadband and $V_{RX,RJ}$ is the receiver random voltage noise. $V_{TX,Swing}$ is the transmitter voltage swing, V_{ISI} is the channel intersymbol interference, V_{XTK} is the channel crosstalk, and V_{SSO} is the output voltage noise due to simultaneous switching output noise. Equation (6.5) states that the voltage at the receiver pad must be larger than a certain amount (this is the receiver voltage input requirement). However, this requirement is not often directly used.

A more practical use for the voltage budget is to require a minimum transmitter voltage swing at the receiver input. This is often done using an eye mask, which specifies the eye opening at various timing locations. Figure 6.10 illustrates one common eye mask. Some I/O interfaces, such as the DDR memory system, use two different eye opening requirements: one for the set-up side, and one for the hold side [7].

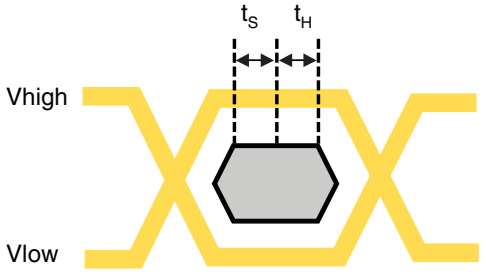


Figure 6.10 Typical Eye Mask Example

6.6 Summary

This chapter provides an overview of channel analysis and discusses why and how a channel budget is defined and used. At low data rates, the simple linear sum of deterministic jitter is used. At multi-gigabit data rates, random jitter is included, using a root sum of squares (RSS). This approach is based on the dual-Dirac model, and it contains significant pessimism. The better approach is to use a more rigorous statistical simulator. However, such simulation requires more sophisticated component models, which may not be available during the initial design stage. Although one often does not have the opportunity to define the channel budget directly, one can still quantify pessimism in the design by checking an existing budget against mass production parts, and allowing a tighter budget for next-generation interfaces.

References

1. "Fibre Channel - Methodologies for Jitter Specification," *National Committee for Information Technology Standardization (NCITS) Technical Report T11.2/Project 1230/Rev 10.0*, Jun. 1999.

2. M. P. Li, J. Wilstrup, R. Jesson, and D. Petrich, "A new method for jitter decomposition through its distribution tail fitting," in *Proceedings of International Test Conference*, 1999.
3. C. Madden, D. Oh, and C. Yuan, "System level deterministic and random jitter measurement and extraction for multi-gigahertz memory buses," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2004, pp. 7–10.
4. *PCI Express Card Base Specification*, Rev. 2.0, PCI-SIG, pp. 243–250, Dec. 20, 2006.
5. *PCI Express® Card Electromechanical Specification*, Rev. 2.0, PCI-SIG, pp. 43–46, April 11, 2007.
6. D. Oh, F. Lambrecht, J. Reng, S. Chang, B. Chia, C. Madden, and C. Yuan. "Prediction of system performance based on component jitter and noise budgets," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2007, pp. 33–36.
7. D. Oh, W. Kim, B. Stott, L. Yang, and C. Yuan, "Channel timing error analysis for DDR2 memory systems," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2005, pp. 3–6.

CHAPTER 7

Manufacturing Variation Modeling

Dan Oh

The voltage and timing budget analysis, discussed in Chapter 6, "Channel Voltage and Timing Budget," must guarantee reliable channel operations in the real world. This means that for high-volume manufacturing (HVM), the channel must be robust, even under worst-case operating conditions. To this end, variations in the channel (such as trace impedances and the material properties of the packages and printed circuit boards [PCBs]) and process variations in the transmitter and receiver must be considered in the channel analysis. Accurately characterizing these variations will result in a more relaxed and/or realistic specification of these parameters. Accurate specifications improve the component and system yield, and ultimately reduce the overall system cost.

As mentioned in previous chapters, performing the channel analysis based on worst-case conditions guarantees adequate channel performance. For example, the passive channel analysis is typically performed under a preselected set of worst- or corner-case conditions. The result is a worst-case value for channel timing jitter (t_{CH}). t_{CH} must satisfy the timing budget equation, based on the linear sum methodology discussed in Chapter 6. Engineers often derive the predefined corner cases from previous experience, engineering judgments, or educated guesses. These predefined cases do not necessarily represent the true worst-case conditions, but are close approximations. Worst-case–based analysis often results in pessimistic performance distributions that lead to overdesigned systems. Because the corner cases, and/or samples, are not statistically balanced, performing any qualitative analysis, such as sensitivity analysis or analytical system modeling, is difficult.

Alternatively, one can characterize the channel using statistical analysis. Statistical analysis is preferred over worst-case–based analysis, because it not only predicts the performance distribution, but also provides additional information (such as sensitivity and worst- and best-case conditions) based on well-defined mathematical models. The overall channel design can be

optimized using statistical analysis. For example, driver strength and termination values can be tuned to produce a better channel voltage margin.

One of the widely used statistical analysis methods is based on the Monte Carlo method [1]. Because the Monte Carlo approach randomly generates a large number of samples to characterize the performance distribution, it is more suitable when simulation times are short. However, it may not be ideal for channel analysis, where there are many parameters to vary, and each simulation could take a significant length of time. The other widely used statistical method is based on Design of Experiment (DoE) methods, such as orthogonal arrays and Central Composite Design (CCD).

A common experiment used by many engineers to study the sensitivity of each parameter is changing one parameter at a time while the rest of the parameters are set to nominal values. Although this approach seems to be intuitive and results in the smallest number of experiments, it does not produce an accurate statistical model. This is because the calculated outputs are strongly biased towards the nominal settings.

A better experiment design can be obtained using orthogonal arrays (OAs), based on the Taguchi method [5–7]. Genichi Taguchi developed this method to improve the quality of manufactured goods. The Taguchi method is widely used in many engineering fields, including, more recently, channel simulation [2–4].

This chapter reviews statistical analysis, based on orthogonal arrays. Realistic channel-simulation examples are employed throughout the entire process, and a linear regression model is used to model performance. Although a surface response model is commonly used to optimize the performance [8], it is beyond the scope of this book and it is omitted.

7.1 Introduction to the Taguchi Method

Correct usage of DoE provides a balanced experiment that exercises all the input factors in an unbiased manner. A brute-force way to perform such an experiment, referred to as a *full-factorial experiment*, is to cover all the possible combinations. Although this is a sure way to obtain a balanced experiment, it is not practical, because the number of experiments grows quickly with the number of factors. For instance, if each factor has three level settings (for example, low (L), nominal (N), and high (H)), it would take 3^N experiments for N factors. In a typical channel model, the number of variables N can easily exceed 10. The number of experiments would exceed 59,000.

The input parameters that are adjusted in an experiment are called *factors*. The different settings for these input parameters (such as L, N, and H) are called levels. In principle, different factors can have different levels. OAs for mixed-level experiments are available for limited cases. This chapter focuses on three-level factors, purely for practical reasons.

A variety of small-sized experiments, called *fractional-factorial experiments* (FFEs), maintains a statistical balance called orthogonality. The experiment matrix, used for these fractional-factorial experiments, is called an orthogonal array (OA). Obviously, fractional-factorial experiments do not provide all the potential information that one could obtain from a full-factorial experiment; however, they provide sufficient information for post-statistical

analysis. Taguchi developed a set of OAs that can be applied to various applications [5]. Table 7.1 demonstrates the effectiveness of OAs by comparing the number of experiments required for a full-factorial experiment to the number required for OAs.

Table 7.1 Comparison of the Total Number of Experiments for Different Input Factors Between Full-Factorial and OAs

# of Factors	# of Levels	Full Factorial	OA
2	2	4	4
3	2	8	4
4	2	16	8
7	2	128	8
15	2	32,768	16
31	2	2,147,483,648	32
4	3	81	7
13	3	1,594,323	27
17	3	129,140,163	81

7.1.1 Properties of Orthogonal Arrays

One of the most fundamental and important properties of OAs is that all the experiments are balanced. In other words, the impact of a particular factor can be evaluated independent of the other factors. To illustrate this point, consider the L9 OA, shown in Table 7.2, which supports four three-level factors with nine experiments. Note that each column contains an equal number of levels. For any given factor level, the settings of the other factor levels are balanced, allowing the impact of one factor change to be calculated independent of the other factors (assuming that all factors are independent). For instance, all the columns contain three 0s, three 1s, and three 2s.

Table 7.2 L9 Orthogonal Array

Trial #	F1	F2	F3	F4
E1	0	0	0	0
E2	0	1	1	1
E3	0	2	2	2
E4	1	0	1	2
E5	1	1	2	0

Table 7.2 L9 Orthogonal Array

Trial #	F1	F2	F3	F4
E6	1	2	0	1
E7	2	0	2	1
E8	2	1	0	2
E9	2	2	1	0

7.1.2 Interaction of Factors

As described in the previous section, when all the factors are independent, the impact of an individual factor can be calculated regardless of other factors, as long as all the levels of the other factors are balanced. However, in real life, some factors are often correlated or interrelated. For instance, assume that the impact of factor F1 may increase or decrease, depending on the setting of factor F2. In this case, the interaction between F1 and F2 has to also be accounted for. Note that the L9 OA can be used for either four factors, or three factors with one interaction.

The OA should also be balanced in terms of interactions. OAs, by default, provide a balanced experiment in terms of two-factor interactions, as illustrated in Table 7.3. As with the single-factor case, the level settings for two factors are balanced. However, this is no longer true for three-factor interactions, as shown in Table 7.4. L9 OA simply does not contain enough cases to cover all the combinations of three factors. However, the L54 OA supports 54 experiments and up to five input factors. It provides a balanced experiment, even for three factor interactions.

Table 7.3 Two-Factor Interaction Setting of L9 Orthogonal Array

Trial #	F1/F2	F1/F3	F1/F4	F2/F3	F2/F4	F3/F4
E1	0/0	0/0	0/0	0/0	0/0	0/0
E2	0/1	0/1	0/1	1/1	1/1	1/1
E3	0/2	0/2	0/2	2/2	2/2	2/2
E4	1/0	1/1	1/2	0/1	0/2	1/2
E5	1/1	1/2	1/0	1/2	1/0	2/0
E6	1/2	1/0	1/1	2/0	2/1	0/1
E7	2/0	2/2	2/1	0/2	0/1	2/1
E8	2/1	2/0	2/2	1/0	1/2	0/2
E9	2/2	2/1	2/0	2/1	2/0	1/0

Table 7.4 Three-Factor Interaction Setting of L9 Orthogonal Array

Trial #	F1/F2/F3	F1/F2/F4	F1/F3/F4	F2/F3/F4
E1	0/0/0	0/0/0	0/0/0	0/0/0
E2	0/1/1	0/1/1	0/1/1	1/1/1
E3	0/2/2	0/2/2	0/2/2	2/2/2
E4	1/0/1	1/0/2	1/1/2	0/1/2
E5	1/1/2	1/1/0	1/2/0	1/2/0
E6	2/0/2	1/2/1	1/0/1	2/0/1
E7	2/0/2	2/0/1	2/2/1	0/2/1
E8	2/1/0	2/1/2	2/0/2	1/0/2
E9	2/2/1	2/2/0	2/1/0	2/1/0

The interaction of factors is an important consideration when attempting to model systems accurately. However, accounting for these interactions in general experiments can be quite complicated. First, a Taguchi analysis is run without any interaction terms, to identify which factors are most likely related. Based on the results of the analysis, the interaction levels of all the possible combinations are estimated. From this estimate, a few dominant interaction cases can be selected for the next Taguchi analysis, which includes interaction columns. This process is quite complex, and although this is possible in some cases, it is not possible for general cases. Consequently, interaction modeling is omitted in this book.

7.1.3 OA Notations

OA naming notation describes the nature of a specific OA. The most general form of notation is described in this section. Although this form is not commonly used, all other notations are simplifications of this general form, and their meanings should be straightforward. The most general notation is

$$\mathrm{L}_{trials}(levels^{factors}, strength)$$

where *trials* represents the total number of experiments, *levels* represents the number of factor levels, and *factors* is the maximum number of input factors supported by the OA. For example, $\mathrm{L}_{27}(3^{13}, 2)$ can be used for any number of input factors below 14. In general, using a higher order of factors can improve the statistical quality of the experiment (and your confidence in the results), but it is not necessary. No restrictions exist on how one assigns OA columns to the input factors, but the results could vary, based on the column assignments. *strength* represents the highest number of interaction factors supported by the OA. For example, L9 OA supports a maximum of four factors and interaction between two of them; it is represented as $\mathrm{L}_9(3^4, 2)$. Common short notations for this OA are $\mathrm{L}_9(3^4)$ or L9.

7.1.4 Common Three-level OAs

Table 7.5 shows common OAs for three-level factors: $L_9(3^4, 2)$, $L_{18}(3^7, 2)$, and $L_{27}(3^{13}, 2)$. (Note that the high-order tables for $L_{36}(3^{13}, 2)$, $L_{54}(3^5, 3)$, and $L_{81}(3^{40}, 2)$ are listed later in Table 7.17 in Section 7.5.)

Table 7.5 $L_9(3^4, 2)$, $L_{18}(3^7, 2)$, and $L_{27}(3^{13}, 2)$ Orthogonal Arrays

$L_9(3^4, 2)$	$L_{18}(3^7, 2)$	$L_{27}(3^{13}, 2)$
0000	0000000	0000000000000
0112	1111110	1011122012100
0221	2222220	2022211021200
1011	0012120	0101112201210
1120	1120200	1112201210010
1202	2201010	2120020222110
2022	0102211	0202221102120
2101	1210021	1210010111220
2210	2021101	2221102120020
	0220111	0010111220121
	1001221	1021200202221
	2112001	2002022211021
	0121022	0111220121001
	1202102	1122012100101
	2010212	2100101112201
	0211202	0212002022211
	1022012	1220121001011
	2100122	2201210010111
		0020222110212
		1001011122012
		2012100101112
		0121001011122
		1102120020222
		2110212002022
		0222110212002
		1200202221102
		2211021200202

7.1.5 Sensitivity Analysis (Main/Average Effect)

After the design of experiment is performed using OAs, the output is processed using various standard statistical analysis tools. Here, "output" refers to the parameters extracted from the channel simulations (for example, transient SPICE simulations). Typical parameters of interest are the voltage margin (or data-eye opening at a given sampling point), and timing jitter (or data-eye fuzz at a given reference voltage for single-ended signaling, or zero crossing for differential signaling).

Carefully balanced experiments yield many useful statistical parameters. This section covers how to compute the sensitivity information derived from the OA results. Because an I/O channel depends on so many factors, identifying which of the input parameters has the greatest impact on the system is important. This impact is referred to as the *main effect*, or *average effect*, and it is derived from the OA.

The main effect for a given factor is calculated by averaging all the values at fixed-level settings. Consider a simple case, where the channel voltage margin (eye opening at the eye center) is the parameter of interest. The average value (or main effect) of the voltage margin, at three different settings of termination impedance and PCB length, is plotted in Figure 7.1 and Figure 7.2, respectively. In Figure 7.1, the system output varies linearly with the termination impedance. In Figure 7.2, there is a non-linear variation relative to the PCB length. Because a typical system exhibits some sort of non-linear variation, modeling this non-linearity accurately, from a performance standpoint, is important. Strictly speaking, the main effect is not sensitivity, because it is not normalized to unit variation. As a result, if one factor varies more than the others do, that factor appears to be a more sensitive parameter than the others are, though its actual sensitivity could be less. What really matters is the total variation over a range, and main effect provides this information.

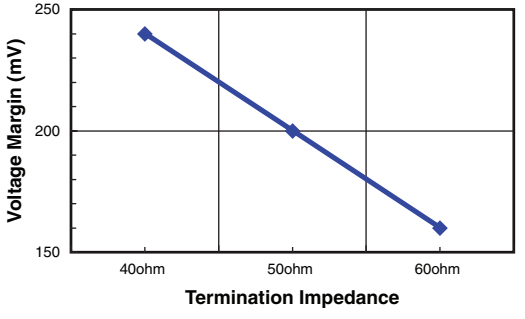


Figure 7.1 Main Effect with Linear Variation

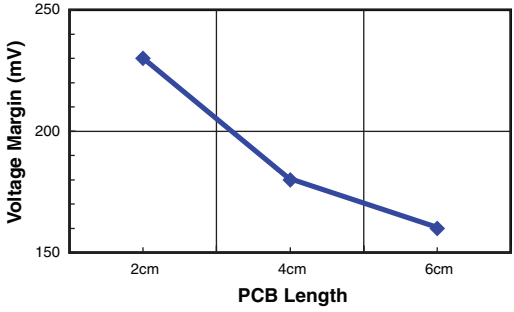


Figure 7.2 Main Effect with Nonlinear Variation

The main-effect plots can also show interactions between two factors by plotting the curve for a given factor against another factor. Figure 7.3(a) illustrates a case with small or no interaction; the curves are simply shifted from each other. Figure 7.3(b) shows some variations in the curves, but no drastic change, in terms of the direction of curve, has been observed. This case shows a minor interaction between two factors. Figure 7.3(c) shows a strong interaction. The level of interaction can be quantified by measuring the variation of the slopes in the main plots.

7.1.6 Worst and Best Cases Estimations

Predicting the best- and worst-case conditions is quite useful in many applications. Use this simple expression to estimate the output performance:

$$Estimation = \overline{T} + (F_1(L_i) - \overline{T}) + \cdots (F_N(L_i) - \overline{T}) \tag{7.1}$$

where $\overline{T}$ is the grand average (the average of all outcomes), and $F_1(L_i)$ is the average value of all outcomes (with the factor F_1 being at level L_i). The estimation at the given setting is calculated by summing individual F_i terms with the specified level. Although the expression seems elementary, it provides an accurate estimation. The best- and worst-case estimations and their settings are obtained by calculating all the possible combinations using (7.1). When the interaction terms are negligible, find the best condition by simply picking the maximum (or minimum, depending on the output performance type) setting for each factor in the main plots.

Note that (7.1) contains no interaction terms, so the estimation can only be accurate when interaction is small. When a minor interaction exists, the estimation could be slightly off, but the predicted corner-case setting can still be accurate. A more accurate value is obtained by simply performing additional experiments based on the predicted settings. Table 7.6 compares data using the estimation from (7.1) to actual simulation data. The Taguchi method is applied using various physical parameters (such as dielectric constant, height, trace width, and so on) to compute the PCB trace impedance Z_o. Note that a good match is found in this case.

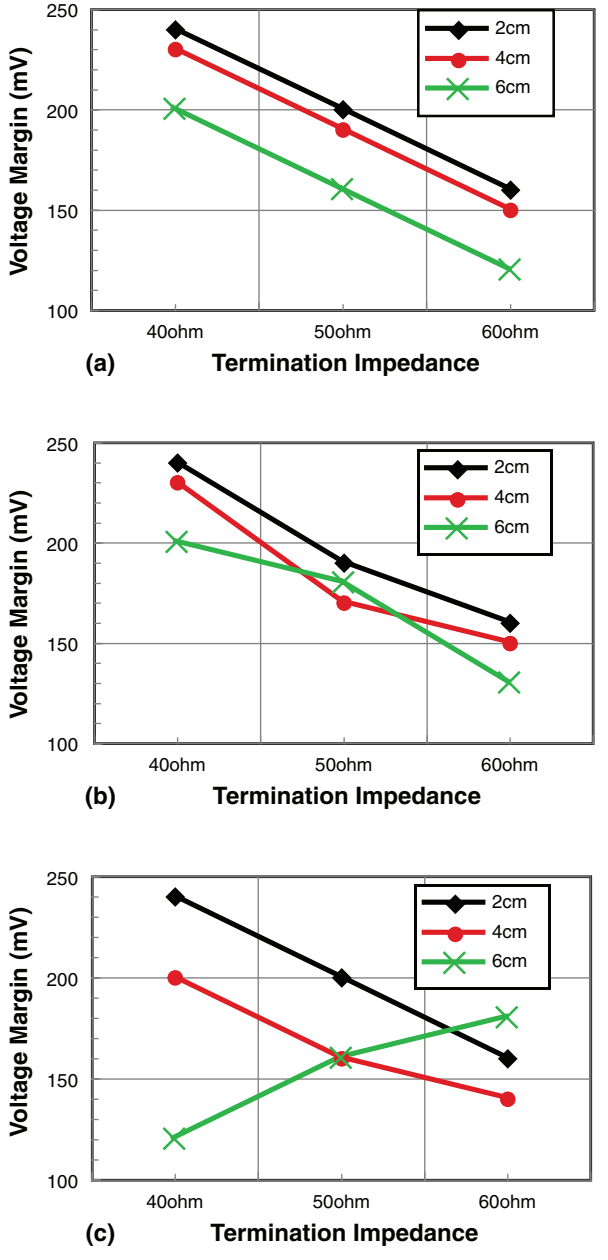


Figure 7.3 Main Plot Showing (a) No, (b) Minor, and (c) Strong Interactions

Table 7.6 Accuracy of the Estimated PCB Impedance Using (7.1)

F1	F2	F3	F4	F5	F6	F7	F8	F9	Estimation (Ω)	Actual (Ω)
2	2	0	2	1	2	2	0	0	41.6	42.0
2	2	0	2	1	2	2	1	0	42.9	43.2
2	2	0	2	1	2	2	0	1	43.2	43.5
2	2	0	2	1	2	2	2	0	44.0	44.0
2	2	0	2	1	1	1	0	2	44.3	44.5

7.1.7 Linear Regression Models

Linear regression is a widely used statistical technique to model system performance. The first-order linear regression model (LRM) is expressed as follows:

$$y = C_0 + C_0F_1 + \cdots + C_NF_N + \varepsilon. \tag{7.2}$$

The coefficient C_i s are typically calculated by performing the least square fit. By adding additional terms, the interaction can be modeled also:

$$y = C_0 + \sum_{i=1}^{N} C_iF_i + \sum_{i=1}^{N}\sum_{j>i}^{N} C_iF_iF_j + \varepsilon. \tag{7.3}$$

The previous expressions can be further extended to handle non-linear behavior by adding the second-order terms as follows:

$$y = C_0 + \sum_{i=1}^{N} C_iF_i + \sum_{i=1}^{N} B_iF_i^2 + \varepsilon \tag{7.4}$$

$$y = C_0 + \sum_{i=1}^{N} C_iF_i + \sum_{i=1}^{N} B_iF_i^2 + \sum_{i=1}^{N}\sum_{j>i}^{N} C_iF_iF_j + \varepsilon. \tag{7.5}$$

An analytical expression of output performance can be useful when performing additional analysis, such as surface response modeling [8]. However, a simple distribution and estimation of worst and best cases is often sufficient to post process the data. Generating an accurate regression model is not an easy process. It often involves an unstable numerical approximation that loses accuracy. The linear regression model is widely developed, and presented in many textbooks, so a detailed description is not covered in this book. Instead, the next section presents a new way to model output performance that is fast, accurate, and numerically more stable than linear regression.

7.1.8 Piecewise Linear Models

Section 7.1.6 discusses a simple, yet effective, way to predict the output performance of a given factor setting using (7.1). This equation can also be used for arbitrary settings by approximating them using a piecewise linear assumption. For instance, the impact of a factor F_1 at level j can be approximated using linear interpolation, based on the two values at levels m and n, where levels m and n bound the level j. (Any high-order interpolation can be used to improve interpolation accuracy.) This model provides a numerically stable and accurate way to model output performance, regardless of system non-linearity. Interaction can be also accounted for by extending interpolation to 2-D space.

Because the analytical model is no longer present, the distribution and variation cannot be calculated analytically. However, they can be easily obtained by performing a Monte Carlo simulation using the PWL system model. This approach is more general because it can handle any arbitrary distributions for input factors.

It is important to note that the Monte Carlo method can also be applied to channel simulation and obtain similar results. The difference is in the simulation time. Here, the Taguchi method is first applied to derive a piecewise linear (PWL) model for the system. Then the Monte Carlo simulation is performed using the PWL model. Because the channel simulation takes much longer than the evaluating PWL model, the simulation time is drastically improved.

7.1.9 Analysis of Variance (ANOVA)

Analysis of variance (ANOVA) is a collection of statistical models that calculate the impact of variation in a variety of input factors. These models are useful when estimating the contribution of individual factors to output performance. The details of ANOVA models are readily available from many statistics textbooks [8], so they are not discussed here.

Perhaps the most important parameter in ANOVA tables is the percent contributions from individual factors. These values estimate the impact of individual factors: the higher the percentage, the more dominant the factor. Table 7.7 is a typical ANOVA table. When the contributions of some of the factors are small, the accuracy of the table is improved by merging these small terms with an error term. This process is called *pooling*. Table 7.8 is an ANOVA table after pooling. Finally, note that ANOVA analysis assumes that factors are independent. The main plot, described in Section 7.1.5, shows similar information, but does not provide a numeric value.

Table 7.7 Sample ANOVA Table Before Pooling

Factors	Degrees of Freedom	Sum of Squares	Variance	Variance (F) Ratio	Pure Sum of Squares	% Contribution
F1	2	0.000193	9.66E-05	2.28	0.000109	0.0709
F2	2	0.000176	8.82E-05	2.08	9.16E-05	0.0598

F3	2	0.000248	0.000124	2.92	0.000163	0.106
F4	2	0.000258	0.000129	3.04	0.000173	0.113
F5	2	0.00058	0.00029	6.84	0.000495	0.323
F6	2	0.000397	0.000198	4.68	0.000312	0.204
F7	2	0.0226	0.0113	266	0.0225	14.7
F8	2	0.00672	0.00336	79.3	0.00664	4.34
F9	2	0.0652	0.0326	769	0.0651	42.5
F10	2	2.37E-05	1.19E-05	0.28	0	0
F11	2	0.0506	0.0253	597	0.0505	33
F12	2	0.00607	0.00303	71.6	0.00598	3.91
Error	2	8.48E-05	4.24E-05			0.68
Total	26	0.153				100

Table 7.8 Sample ANOVA Table After Pooling

Factors	Degrees of Freedom	Sum of Squares	Variance	Variance (F) Ratio	Pure Sum of Squares	% Contribution
F1	Pooled	0.000193				
F2	Pooled	0.000176				
F3	Pooled	0.000248				
F4	Pooled	0.000258				
F5	Pooled	0.00058				
F6	Pooled	0.000397				
F7	2	0.0226	0.0113	15.3	0.0211	13.8
F8	Pooled	0.00672				
F9	2	0.0652	0.0326	44.2	0.0637	41.6
F10	Pooled	2.37E-05				
F11	2	0.0506	0.0253	34.3	0.0492	32.1
F12	Pooled	0.0148				
Error	20	8.48E-05	4.24E-05			0.68
Total	26	0.153	0.153			100

7.2 DDR DRAM Command/Address Channel Example

DDR DRAM is widely used for consumer devices, such as HDTV, as well as in PCs. However, the consumer device market is much more cost sensitive than the PC market. Memory subsystems for consumer devices must be designed using low-cost system solutions. Consequently, the manufacturing specifications must be relaxed to reduce the system cost. Ideally, the specifications are relaxed enough that no testing is required, saving the significant cost associated with screening and testing.

Soldering DRAM devices to a board improves the signal quality for data signals, because data (DQ) signals are routed point-to-point and need no connector transition. The DDR channel, used for the consumer market, is of a much lower capacity than the channels used for PCs. However, because multiple devices must share the same address and command (RQ) signals, routing RQ signals when the consumer system's board has a limited number of routing layers can be quite difficult. Significant reflection would occur in cases like this, due to the long trace branches and the wide variation in channel impedance. In this section, a multi-drop DDR RQ channel is used to demonstrate the Taguchi method.

Figure 7.4 illustrates the RQ channel topology. The lead-in trace length is approximately 5 cm. It branches out to two traces and terminates on the board. Note that the termination impedance is slightly higher, in order to boost the receiver swing. Table 7.9 shows the input factor settings and descriptions. A total of 12 factors are presented, and $L_{27}(3^{13}, 2)$ OA is used. The channel setup- and hold-side eye openings and eye fuzz are obtained from HSPICE simulation. The bit duration is 3.012ns. The setup-side eye opening is measured at a 0.565ns shift from the eye center to account for clock jitter, transmitter jitter, and the set-up time requirement. The hold-side eye opening is measured at a 0.69ns shift from the eye center to account for the hold time requirement. Both cases report the worst side eye opening, above or below the reference voltage (0.9V). The center of the eye is calculated based on the fuzz center. Table 7.10 lists the results of the experiment.

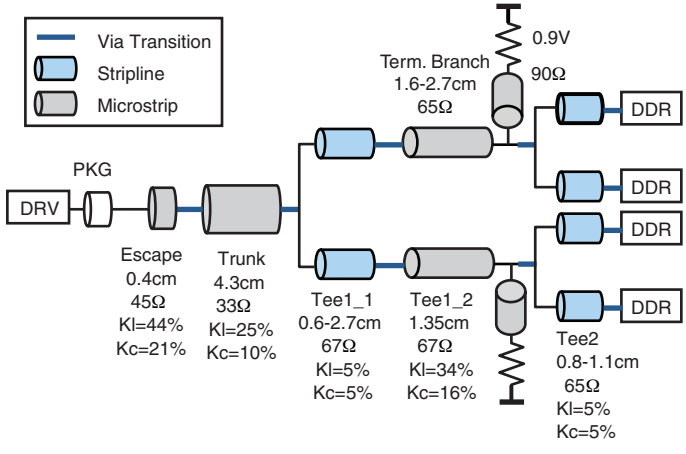


Figure 7.4 DDR RQ Channel Topology

Table 7.9 Input Factor Settings for DDR RQ Channel Simulation

	Factors	Description	Low	Nominal	High
1	dtr	Risetime of output driver	100ps	170ps	240ps
2	Ctx	Driver output capacitance	3pF	3.5pF	4pF
3	Ron_vddio	Driver pull up impedance	14Ω	18Ω	22Ω
4	Ron_gnd	Driver pull down impedance	14Ω	18Ω	22Ω
5	Cdram	DRAM receiver capacitance	1pF	1.5pF	2pF
6	Ldram	DRAM receiver leakage inductance	2nH	3nH	4nH
7	DelCdram	Pin-to-pin rx capacitance difference	-60fF	0	60pF
8	DelLdram	Pin-to-pin rx inductance difference	-1nH	0	1nH
9	Rt_vddio	Pull up termination impedance	235Ω	240Ω	245Ω
10	Rt_gnd	Pull down termination impedance	235Ω	240Ω	245Ω
11	mbscale	PCB impedance scaling factor	0.85	1.0	1.15
12	pkgscale	Package impedance scaling factor	0.85	1.0	1.15

Table 7.10 Output Results for DDR RQ Channel Simulation

	Factors	Setup Eye Margin (mV)	Hold Eye Margin (mV)	Fuzz (ps)
1	0:0:0:0:0:0:0:0:0:0:0:0	480	529	376
2	1:0:1:1:1:2:2:0:1:2:1:0	426	526	346
3	2:0:2:2:2:1:1:0:2:1:2:0	399	527	336
4	0:1:0:1:1:1:2:2:0:1:2:1	407	538	358
5	1:1:1:2:2:0:1:2:1:0:0:1	382	500	389
6	2:1:2:0:0:2:0:2:2:2:1:1	420	498	355
7	0:2:0:2:2:2:1:1:0:2:1:2	363	506	406
8	1:2:1:0:0:1:0:1:1:1:2:2	438	522	332
9	2:2:2:1:1:0:2:1:2:0:0:2	398	490	389
10	0:0:1:0:1:1:1:2:2:0:1:2	399	508	379

Table 7.10 Output Results for DDR RQ Channel Simulation

	Factors	Setup Eye Margin (mV)	Hold Eye Margin (mV)	Fuzz (ps)
11	1:0:2:1:2:0:0:2:0:2:2:2	388	524	363
12	2:0:0:2:0:2:2:2:1:1:0:2	408	501	374
13	0:1:1:1:2:2:0:1:2:1:0:0	399	511	381
14	1:1:2:2:0:1:2:1:0:0:1:0	433	500	295
15	2:1:0:0:1:0:1:1:1:2:2:0	436	546	370
16	0:2:1:2:0:0:2:0:2:2:2:1	433	515	326
17	1:2:2:0:1:2:1:0:0:1:0:1	383	481	410
18	2:2:0:1:2:1:0:0:1:0:1:1	400	535	417
19	0:0:2:0:2:2:2:1:1:0:2:1	353	513	389
20	1:0:0:1:0:1:1:1:2:2:0:1	452	518	380
21	2:0:1:2:1:0:0:1:0:1:1:1	406	507	351
22	0:1:2:1:0:0:1:0:1:1:1:2	422	496	324
23	1:1:0:2:1:2:0:0:2:0:2:2	400	521	361
24	2:1:1:0:2:1:2:0:0:2:0:2	386	514	444
25	0:2:2:2:1:1:0:2:1:2:0:0	411	493	335
26	1:2:0:0:2:0:2:2:2:1:1:0	409	523	432
27	2:2:1:1:0:2:1:2:0:0:2:0	449	531	301

Figures 7.5–7.7 illustrate the main plots for the DDR RQ channel. On the setup-side eye opening (see Figure 7.5), DRAM input capacitance plays the dominant role. On the hold-side eye opening (see Figure 7.6), the driver impedance and channel impedance have a greater impact than capacitance; this is because the eye has shrunk due to the reflection at the hold side. For the eye fuzz (see Figure 7.7), all the input capacitance, the driver impedance, and the channel impedance are important, because both ISI (caused by the DRAM input capacitance) and reflection (caused by mismatch) are important.

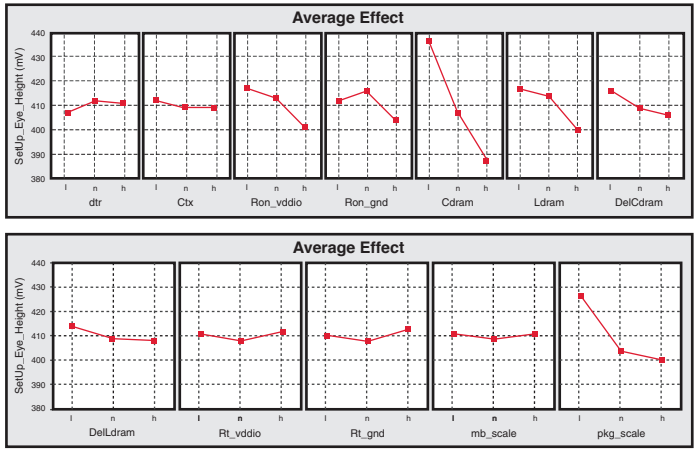


Figure 7.5 Main Plots for DDR RQ Channel Simulation for the Setup-Side Eye Opening

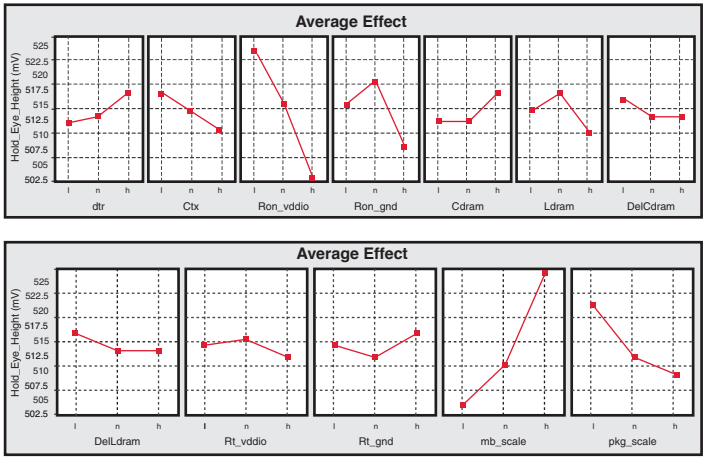


Figure 7.6 Main Plots for DDR RQ Channel Simulation for the Hold-Side Eye Opening

Table 7.11 is the ANOVA table for eye fuzz before applying pooling. Table 7.12 is the same fuzz simulation after pooling. The dominant impact is the DRAM input capacitance, as predicted by the main effect. Equation (7.1) is used to perform the Monte Carlo analysis, and Figure 7.8 shows the final distribution for the setup- and hold-side eye opening. Figure 7.9 illustrates the dis-

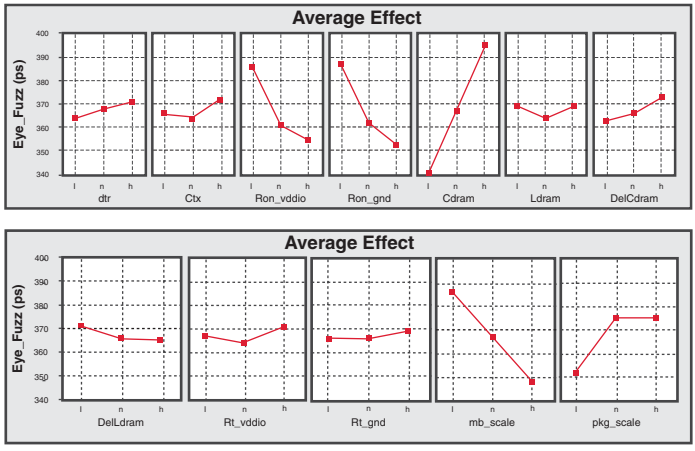


Figure 7.7 Main Plots for DDR RQ Channel Simulation for the Eye Fuzz

tribution for eye fuzz. For the setup-side eye distribution, the mean value is 410mV, with a standard deviation of 18.3mV. For the hold-side eye distribution, the mean value is 514mV, with a standard deviation of 10.4mV. For the eye fuzz distribution, the mean value is 367ps, with a standard deviation of 24.8ps.

Table 7.11 DDR RQ Fuzz Simulation ANOVA Table Before Pooling

Factors	Degrees of Freedom	Sum of Squares	Variance	Variance (F) Ratio	Pure Sum of Squares	% Contribution
dtr	2	221	110	3.53	158	0.448
Ctx	2	305	153	4.89	243	0.687
Ron_vddio	2	4.84e+03	2.42e+03	77.4	4.78e+03	13.5
Ron_gnd	2	5.85e+03	2.93e+03	93.6	5.79e+03	16.4
Cdram	2	1.36e+04	6.78e+03	217	1.35e+04	38.2
Ldram	2	154	76.9	2.46	91.3	0.258
DelCdram	2	395	197	6.32	332	0.94

Table 7.11 DDR RQ Fuzz Simulation ANOVA Table Before Pooling

Factors	Degrees of Freedom	Sum of Squares	Variance	Variance (F) Ratio	Pure Sum of Squares	% Contribution
DelLdram	2	192	95.8	3.07	129	0.365
Rt_vddio	2	221	111	3.54	159	0.449
Rt_gnd	2	58.3	29.1	0.932	0	0
mb_scale	2	6.5e+03	3.25e+03	104	6.44e+03	18.2
pkg_scale	2	3.01e+03	1.5e+03	48.1	2.95e+03	8.33
Error	2	62.5	31.3			2.29
Total	26	3.54e+04				100

Table 7.12 DDR RQ Fuzz Simulation ANOVA Table After Pooling

Factors	Degrees of Freedom	Sum of Squares	Variance	Variance (F) Ratio	Pure Sum of Squares	% Contribution
dtr	Pooled	221				
Ctx	Pooled	305				
Ron_vddio	2	4.84e+03	2.42e+03	24.1	4.64e+03	13.1
Ron_gnd	2	5.85e+03	2.93e+03	29.1	5.65e+03	16
Cdram	2	1.36e+04	6.78e+03	67.5	1.34e+04	37.8
Ldram	Pooled	154				
DelCdram	Pooled	395				
DelLdram	Pooled	192				
Rt_vddio	Pooled	221				
Rt_gnd	Pooled	58.3				
mb_scale	2	6.5e+03	3.25e+03	32.3	6.3e+03	17.8
pkg_scale	2	3.01e+03	1.5e+03	15	2.81e+03	7.93
Error	2	62.5	101			7.39
Total	26	3.54e+04				100

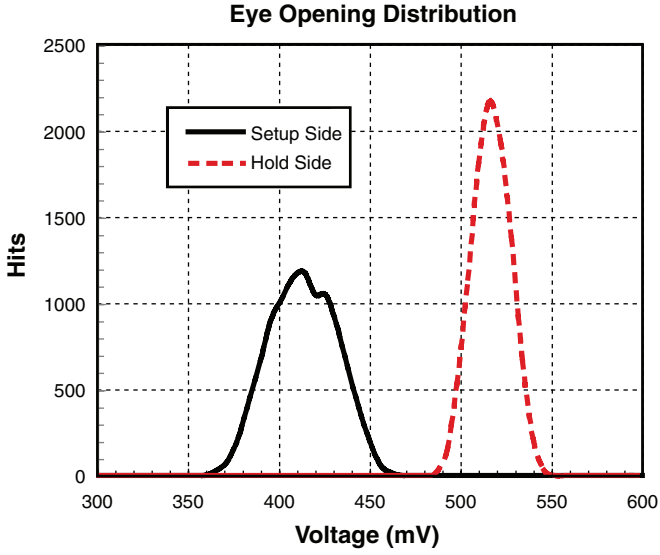


Figure 7.8 Setup- and Hold-Side Eye Opening Distributions

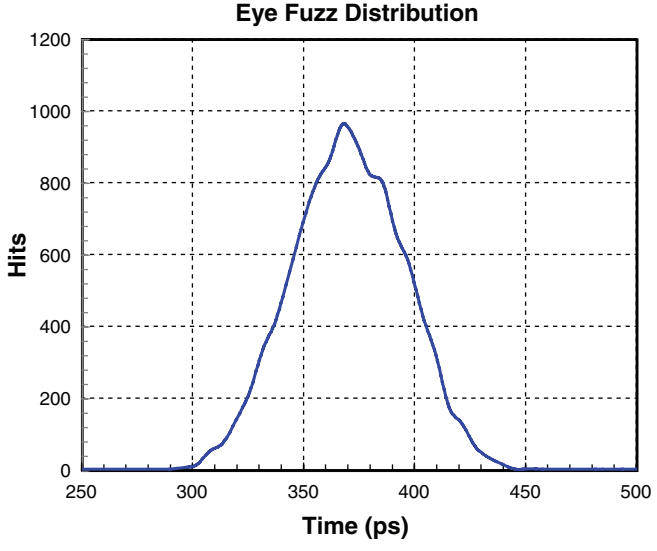


Figure 7.9 Eye Fuzz Distribution

7.3 Backplane Link Modeling Example

In this section, a backplane channel is used to demonstrate another example of Taguchi analysis. The example system comes from Kollipara, et al. [9], where Monte Carlo simulation demonstrated the impact of the statistical variations in SerDes link performance. Figure 7.10 shows the system setup. Figure 7.11 illustrates the physical PCB, or package substrate parameters. Note that the top plane is removed from the field solver model for microstrip traces. Table 7.13 provides the final list of input parameters that are adjusted; 41 parameters are considered in this case.

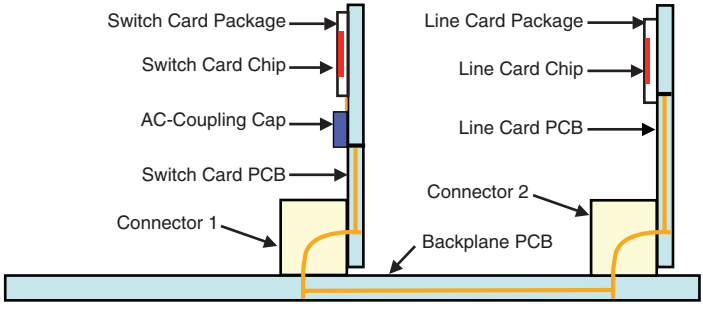


Figure 7.10 Different Components of a Backplane Channel

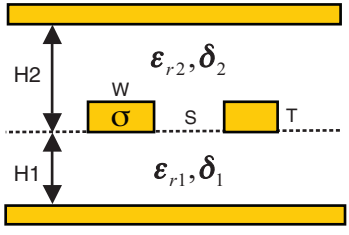


Figure 7.11 Physical Parameters for a PCB or Package Substrate

Table 7.13 Complete List of Input Parameters for Backplane Channel Variations

Component	Parameter	Description	Nominal	3 Sigma
Backplane PCB	ε_{r1}, ε_{r2}	FR4 relative dielectric constant	4.0	0.2
	δ_{r1}, δ_{r2}	FR4 loss tangent	0.02	0.002
	W	Trace width	7.5mil	1mil
	H_1, H_2	Dielectric heights	10mil	1.5mil
	T	Conductor thickness	0.6mil	0.08mil

Table 7.13 Complete List of Input Parameters for Backplane Channel Variations

Component	Parameter	Description	Nominal	3 Sigma
	Via_fac	Via impedance scaling factor	1	0.1
	Via_L	Via length scaling factor	1	0.1
	Cstub	Stub capacitance scaling factor	1	0.1
	Connector_fac	Connector impedance scaling factor	1	0.01
Line Card PCB	ε_{r1}, ε_{r2}	FR4 relative dielectric constant	4.0	0.2
	δ_{r1}, δ_{r2}	FR4 loss tangent	0.02	0.002
	W	Trace width	5mil	0.67mil
	H_1	Bottom dielectric height	5mil	0.75mil
	H_2	Top dielectric height	10mil	1.5mil
	T	Conductor thickness	0.6mil	0.08mil
Switch Card PCB	ε_{r1}, ε_{r2}	FR4 relative dielectric constant	4.0	0.2
	δ_{r1}, δ_{r2}	FR4 loss tangent	0.02	0.002
	W	Trace width	5mil	0.67mil
	H1	Bottom dielectric height	5mil	0.75mil
	H2	Top dielectric height	10mil	1.5mil
	T	Conductor thickness	0.6mil	0.08mil
Line Card Device Package	ε_r	FR4 relative dielectric constant	4.0	0.2
	δ_r	FR4 loss tangent	0.02	0.002
	W	Trace width	2mil	0.27mil
	H	Dielectric height	4.4mil	0.66mil
	T	Conductor thickness	1.2mil	0.16mil

Table 7.13 Complete List of Input Parameters for Backplane Channel Variations

Component	Parameter	Description	Nominal	3 Sigma
Switch Card Device Package	ε_r	FR4 relative dielectric constant	4.0	0.2
	δ_r	FR4 loss tangent	0.02	0.002
	W	Trace width	2mil	0.27mil
	H	Dielectric height	4.4mil	0.66mil
	T	Conductor thickness	1.2mil	0.16mil
Others	$C_{i,\,tx}$, $C_{i,\,rx}$	ESD and parasitic device capacitance	Typical	10%
	σ_{copper}	Copper conductivity	52MS/m	5.2MS/m

Instead of performing a single DoE on all of these parameters, use a two-stage approach. First, convert the physical parameters for each trace model to electrical parameters (for example, propagation constant and characteristic impedance). Next, capture the attenuation and impedance variations using DoE, reducing the total number of parameters to just two for each trace model. (The following section covers this reduction process.) Finally, combine the final trace models with the rest of the parameters to simulate and perform the statistical analysis described in Section 7.3.2. Table 7.14 lists the parameters, which have been reduced to 16. Figure 7.12 illustrates the overall process.

Table 7.14 Reduced Number of Input Parameters for Backplane Channel Variations

Component	Parameter	Description	Nominal	3 Sigma
Backplane PCB	α	Attenuation scaling factor		
	ζ	Impedance scaling factor		
	Via_fac	Via impedance scaling factor	1	0.1
	Via_L	Via length scaling factor	1	0.1
	Cstub	Stub capacitance scaling factor	1	0.1
	Connector_fac	Connector impedance scaling factor	1	0.01

Table 7.14 Reduced Number of Input Parameters for Backplane Channel Variations

Component	Parameter	Description	Nominal	3 Sigma
Line Card PCB	α	Attenuation scaling factor		
	ζ	Impedance scaling factor		
Switch Card PCB	α	Attenuation scaling factor		
	ζ	Impedance scaling factor		
Line Card Device Package	α	Attenuation scaling factor		
	ζ	Impedance scaling factor		
Switch Card Device Package	α	Attenuation scaling factor		
	ζ	Impedance scaling factor		
Others	$C_{i,\,tx}$, $C_{i,\,rx}$	ESD and parasitic device capacitance	Typical	10%

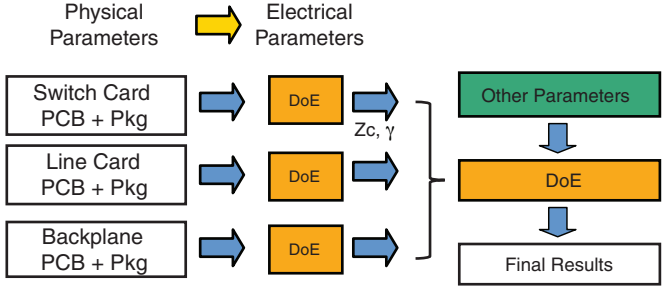


Figure 7.12 Overall Flow of the Backplane Channel Variation Modeling

7.3.1 PCB Trace Modeling

In this section, the DoE process is demonstrated using the backplane PCB trace model, shown in Figure 7.11. The goal of this DoE is to find the distribution of the characteristic impedance and propagation constant due to physical parameter variations. The nine physical parameters, ε_{r1}, ε_{r2},

δ_{r1}, δ_{r2}, H_1, H_2, W, T, and σ_{copper}, are used in this experiment. The three α limits from Table 7.13 are used for the OA table's high and low values. Table 7.15 lists the final factor settings.

Table 7.15 Input Factor Settings for PCB Trace-Modeling Experiments

	Factors	Low	Nominal	High
1	ε_{r1}	3.8	4.0	4.2
2	ε_{r2}	3.8	4.0	4.2
3	δ_{r1}	0.018	0.02	0.022
4	δ_{r2}	0.018	0.02	0.022
5	σ_{copper}	46.8M	52.0M	57.2M
6	W	6.5mil	7.5mil	8.5mil
7	T	0.52mil	0.6mil	0.68mil
8	H_1	8.5mil	10mil	11.5mil
9	H_2	8.5mil	10mil	11.5mil

Because there are nine input factors, an $L_{27}(3^{12}, 2)$ OA is used for this experiment. A quasi-static field solver is used to compute the impedance values. The mean and standard deviations of the simulated cases were 47.6Ω and 3.96Ω, respectively. Of course, these are not the statistical values of the final distribution. The final distribution mean and sigma, calculated using Monte Carlo simulation with a PWL model, are 49.5Ω and 1.38Ω. 10K samples were used for the Monte Carlo simulation. A Monte Carlo simulation, using an actual field solver for the sample number of cases is performed to validate the results. The mean and standard deviations are 49.7Ω and 1.39Ω. Figure 7.13 compares the overall distributions and shows a very good match between the two results.

7.3.2 Complete Backplane Link Modeling

By applying the previous modeling to different traces, the overall backplane channel is constructed using HSPICE W-element transmission lines. The impedance and attenuation variations are modeled by applying scaling factors to the RLGC matrices. Table 7.16 shows the OA factor settings for the overall channel. $L_{81}(3^{40}, 2)$ OA is used for this experiment. The best- and worst-case AC responses are predicted, based on the PWL model, and plotted with the nominal case shown in Figure 7.14. The amplitude at 3.125GHz (which is the Nyquist frequency for this SerDes link) is used to judge the performance. The overall distribution, once again, is compared with an exact field solver and PWL model using Monte Carlo simulation, and the amplitude distributions plotted (see Figure 7.15).

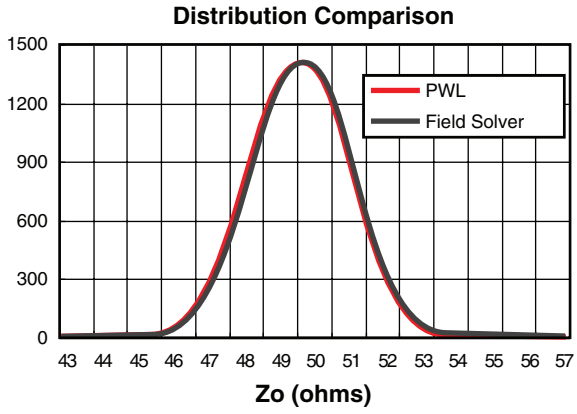


Figure 7.13 PCB Impedance Distribution Using Field Solver and PWL Model Based on OAs

Table 7.16 Input Factor Settings for a Complete Backplane Channel Modeling Experiment

	Factors	Low	Nominal	High
1	Main board PCB α	0.9036	1.0	1.0964
2	Main board PCB ζ	0.9226	1.0	1.0774
3	Via_fac	0.9	1.0	1.1
4	Via_L	0.9	1.0	1.1
5	Cstub	0.9	1.0	1.1
6	Connector_fac	0.9	1.0	1.1
7	Line card PCB α	0.9065	1.0	1.0935
8	Line card PCB ζ	0.9272	1.0	1.0728
9	Switch card PCB α	0.9065	1.0	1.0935
10	Switch card PCB ζ	0.9272	1.0	1.0728
11	Line card Package α	0.9284	1.0	1.0716
12	Line card Package ζ	0.9355	1.0	1.0645
13	Switch card Package α	0.9284	1.0	1.0716
14	Switch card Package ζ	0.9355	1.0	1.0645
15	$C_{i,TX}$	1.44pF	1.6pF	1.76pF
16	$C_{i,RX}$	1.62pF	1.8pF	1.98pF

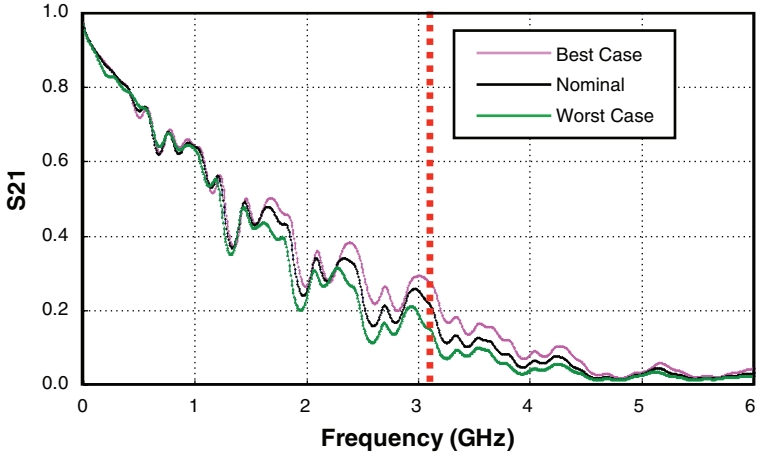


Figure 7.14 AC Responses of the Nominal, Worst-, and Best-Case Channels

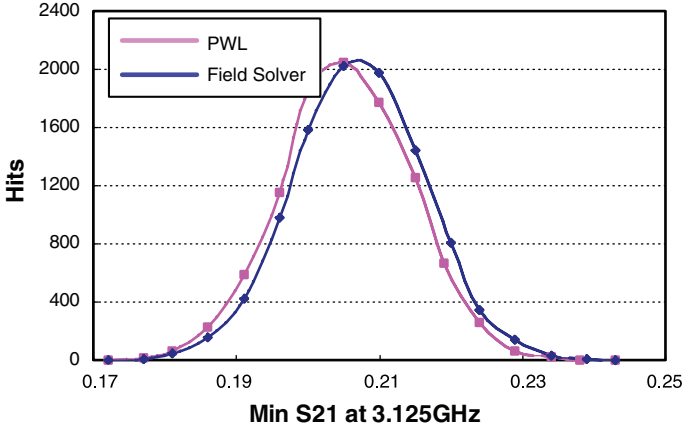


Figure 7.15 S21 Amplitude Distributions at 3.125GHz using PWL and Field Solver Monte Carlo Simulation

7.4 Summary

Modeling manufacturing variation is very important when analyzing channel performance under the worst-case operating conditions. Typical channel settings are not sufficient for HVM production systems. The Taguchi analysis, described in this chapter, is a very useful tool, as long as the interactions between factors are small.

The piecewise linear model introduced in this chapter provides a more robust and reliable way to generate the output distribution. Although rigorous linear regression models can provide statistical parameters in a closed form, they are very hard to apply to real-world examples. The Monte Carlo simulation, with a piecewise linear model, meets most channel-analysis needs.

7.5 Appendix

Table 7.17 $L_{36}(3^{13}, 2)$, $L_{54}(3^5, 3)$, and $L_{81}(3^{40}, 2)$ Orthogonal Arrays

$L_{36}(3^{13}, 2)$	$L_{54}(3^5, 3)$	$L_{81}(3^{40}, 2)$
0000000000000	11111	000000000000000000000000000000000000000000
1111111111110	11111	212110222111101012210201200112101100 1000
2222222222220	22222	121220111222202021120102100221202200 2000
0000111122220	22222	021211022211110101221020120011210110 0100
1111222200000	11000	200021211022211110101221020120011210 1100
2222000011110	10100	112101100100012122011122220202112010 2100
0012012201120	10010	012122011122220202112010210022120220 0200
1120120012200	10001	221202200200021211022211110101221020 1200
2201201120010	01100	100012122011122220202112010210022120 2200
0021021210210	01010	002121102221111010122102012001121011 0010
1102102021020	01001	211201021002212022002000212110222111 1010
2210210102100	00110	120011210110010001212201112222020211 2010
0120210221011	00101	020002121102221111010122102012001121 0110
1201021002121	00011	202112010210022120220020002121102221 1110
2012102110201	22000	111222202021120102100221202200200021 2110
0121002122101	20200	011210110010001212201112222020211201 0210
1202110200211	20020	220020002121102221111010122102012001 1210
2010221011021	20002	102100221202200200021211022211110101 2210
0102220110121	02200	001212201112222020211201021002212022 0020
1210001221201	02020	210022120220020002121102221111010122 1020
2021112002011	02002	122102012001121011001000121220111222 2020
0112201002211	00220	022120220020002121102221111010122102 0120

1220012110021	00202	20120011210110010001212201112222020211120
2001120221101	00022	11001000121220111222202021120102100221 20
0210122020112	01122	01000121220111222202021120102100221202 20
1021200101222	01212	22211110101221020120011210110010001212 20
2102011212002	01221	10122102012001121011001000121220111222 20
0211100212022	02211	00021211022211110101221020120011210110 01
1022211020102	02121	21202200200021211022211110101221020120 01
2100022101212	02112	12110222111101012210201200112101100100 01
0222121101002	10122	02112010210022120220020002121102221111 01
1000202212112	10212	20020002121102221111010122102012001121 01
2111010020222	10221	11201021002212022002000212110222111101 01
0201212011202	20211	01200112101100100012122011122220202112 01
1012020122012	20121	22111101012210201200112101100100012122 01
2120101200122	20112	10022120220020002121102221111010122102 01
	11022	00200021211022211110101221020120011210 11
	12012	21111010122102012001121011001000121220 11
	12021	12022002000212110222111101012210201200 11
	22011	02021120102100221202200200021211022211 11
	21021	20202112010210022120220020002121102221 11
	21012	11110101221020120011210110010001212201 11
	11202	01112222020211201021002212022002000212 11
	12102	22020211201021002212022002000212110222 11
	12201	10201200112101100100012122011122220202 11
	22101	00112101100100012122011122220202112010 21
	21201	21020120011210110010001212201112222020 21
	21102	12201112222020211201021002212022002000 21
	11220	02200200021211022211110101221020120011 21
	12120	20111222202021120102100221202200200021 21
	12210	11022211110101221020120011210110010001 21

	22110	0102100221202200200021211022211110101221
	21210	2220202112010210022120220020002121102221
	21120	1011001000121220111222202021120102100221
		0001212201112222020211201021002212022002
		2122011122220202112010210022120220020002
		1210110010001212201112222020211201021002
		0210022120220020002121102221111010122102
		2001121011001000121220111222202021120102
		1122220202112010210022120220020002121102
		0122102012001121011001000121220111222202
		2210201200112101100100012122011122220202
		1001000121220111222202021120102100221202
		0022120220020002121102221111010122102012
		2110222111101012210201200112101100100012
		1201021002212022002000212110222111101012
		0201200112101100100012122011122220202112
		2022002000212110222111101012210201200112
		1110101221020120011210110010001212201112
		0110010001212201112222020211201021002212
		2201112222020211201021002212022002000212
		1022211110101221020120011210110010001212
		0010001212201112222020211201021002212022
		2101100100012122011222202021120102100022
		1222202021120102100221202200200021211022
		0222111101012210201200112101100100012122
		2010210022120220020002121102221111010122
		1101012210201200112101100100012122011122
		0101221020120011210110010001212201112222
		2222020211201021002212022002000212110222
		1010122102012001121011001000121220111222

References

1. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Chapman & Hall, 1964.
2. E. Matoglu, N. Pham, D. N. D. Araujo, M. Cases, and M. Swaminathan, "Statistical signal integrity analysis and diagnosis methodology for high-speed systems," *IEEE Transactions on Advanced Packaging*, vol. 27, no. 4, pp. 611–629, Nov. 2004.
3. A. Norman, D. Shykind, M. Falconer, and K. Ruffer, "Application of design of experiments (DOE) method to high-speed interconnect validation," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2003, pp. 15–18.
4. W. Beyene, N. Cheng, J. Feng, and C. Yuan, "Statistical and sensitivity analysis of voltage and timing budgets of multi-gigabit interconnect systems," presented at the IEC DesignCon, Santa Clara, CA, 2004.
5. G. Taguchi, *Introduction to Quality Engineering*, Amer. Supplier Ins., 1986.
6. S. H. Park, *Robust Design and Analysis for Quality Engineering*, Champman & Hall, 1996.
7. P. J. Ross, *Taguchi Techniques for Quality Engineering*, 2nd ed., MacGraw-Hill, 1996.
8. D. C. Montgomery, *Design and Analysis of Experiments*, 5th ed., Wiley, 2000.
9. R. Kollipara, B. Chia, Q. Lin, and J. Zerbe, "Impact of manufacturing parametric variations on backplane system performance," presented at the IEC DesignCon, Santa Clara, CA, 2005.

CHAPTER 8

Link BER Modeling and Simulation

Dan Oh, Sam Chang, Jihong Ren, and Vladimir Stojanovic

The voltage and timing budget analysis, discussed in Chapter 6, "Channel Voltage and Timing Budget," must guarantee reliable channel operations in the real world. This means that for high-volume manufacturing (HVM), the channel must be robust, even under worst-case operating conditions. To this end, variations in the channel (such as trace impedances and the material properties of the packages and PCBs) and process variations in the transmitter and receiver must be considered in the channel analysis. Accurately characterizing these variations will result in a more relaxed and/or realistic specification of these parameters. This, in turn, improves the component and system yield, and reduces the overall system cost.

In the past, signal-integrity analysis focused on characterizing the impact of deterministic jitter sources, such as inter-symbol interference (ISI), crosstalk, driver skew, and receiver sampling offset. Uncorrelated and random noise sources, such as power supply noise, thermal noise, and reference clock jitter, are difficult to evaluate during the design phase. Typically, the device jitter caused by these noise sources is measured experimentally. However, if we are to achieve optimal performance in modern, high-speed I/O designs (with their ever-shrinking timing margins), considering the effects of both deterministic and random noise sources during the design phase is crucial.

Traditional SPICE-based simulation techniques accurately predict system-level performance, including various deterministic effects from passive channels and devices. Part I of this book, Chapters 3–5, focuses on passive channel modeling and simulation. Chapter 6 proved that as the I/O speed increases, the component-level jitter analysis cannot accurately predict the impact of jitter at the system level. To address this issue, significant effort is being devoted to simulating the impact of device jitter on the system-level performance, specifically, the system voltage and timing margin at a given bit error rate (BER). This chapter starts with the historical review of link BER simulation methods. It then focuses on the popular statistical approach used

in LinkLab [1–3] and StatEye [4]. Chapter 9, "Fast Time-Domain Channel Simulation Techniques," presents time-domain methods.

8.1 Historical Background and Chapter Organization

Perhaps the most critical part of link BER simulation is characterizing the intersymbol interference (ISI) of a passive channel. A straightforward approach to computing the probability error due to ISI is to consider all possible input data patterns, where the pattern length is as large as the channel response time. Many high-speed channel responses are more than 100-bits long. In such cases, 2^{100} combinations need to be considered for PAM2 signal! Helstrom provides a good summary of the early work done to improve this ISI calculation [5]. This early work falls into two categories: one category estimates bounds on the error probability; the other focuses on improved numerical algorithms with which to perform error calculation. Capsper and Ahmad present a highly efficient method based on the convolution of the pulse response [6] [7]. Using random data pattern and linear time invariant (LTI) assumptions, this convolution approach efficiently calculates the probability mass function (PMF) of the ISI effect, without any approximation. LinkLab and StatEye have adopted this method. Section 8.2 discusses the general simulation framework, based on this convolution. Section 8.3 describes the convolution method itself.

Alternative, fast time-domain simulation techniques have been proposed for non-linear time-invariant systems [8–12]. Using the time-domain simulation can alleviate the random data pattern assumption associated with the previous convolution approach. This allows the modeling of data coding, which is an important topic in high-speed data communication. In addition, by using an improved algorithm, driver non-linearity can be modeled in the time domain. Non-linearity modeling is important in single-ended signaling systems, which often reveal a strong non-linearity due to asymmetric channel pull-up and pull-down response. Although these techniques are still not sufficiently fast enough to predict low BER (10^{-12} to 10^{-18}), they are adequate to characterize the impact of passive channels. Chapter 9 describes the fast time-domain simulation method, along with an acceleration algorithm to calculate ISI PDF [13]. The IBIS *Advanced Technology Modeling* (ATM) standards committee defined an API interface called the Algorithm Model Interface (AMI) to support the system-level BER simulation [14]. This interface allows IP vendors to model proprietary circuit features or behavior, such as equalization and clock-data recovery (CDR) circuitry, in a common EDA environment. The model, however, has limited support for modeling various jitter types. Chapter 9 also briefly describes the AMI model.

One major limitation of most of the early convolution-based techniques is that they treat a passive channel response and jitter sources independently. The result is that the interaction between device jitter and passive channels is not accurately simulated. For instance, to model the system-level impact of jitter, one must consider both its probability density function (PDF) and its spectrum information to simulate jitter amplification and tracking. However, the majority of

the statistical approaches assume a white random jitter spectrum, and jitter is typically modeled as the receiver jitter, ignoring any transmit jitter coloring due to the ISI of passive channel [4] [6] [7]. The equivalent voltage noise (EVN) concept, described by Stojanovic and Horowitz [1], resolves the issues of colored random jitter by using autocorrelation and Gaussian distribution approximation. (Section 8.4 provides a detailed description of EVN.) This method is based on perturbation and is only valid when the jitter amount is small compared to the channel response. Oh, et al., extend this concept to model colored bounded jitter [3]. The transmit duty-cycle distortion (DCD) can also be modeled using the equivalent voltage noise concept. However, a more rigorous approach, without any approximation, based on short and long pulse ISI PDFs has also been presented [2] (see Section 8.3.2).

Modeling clocking architectures is crucial for jitter analysis, because the clocking architecture strongly affects the impact of jitter on the link margin. Clock and data recovery (CDR) is widely used in high-speed serial link designs, such as SerDes. For other types of on-board I/O interfaces, including PCI Express, HyperTransport, FlexIO, and Elastic Interface systems, forwarded or distributed clocking schemes are often used. These interfaces commonly use a single reference clock for transmitter and receiver; hence, modeling potential jitter tracking is important to avoid system overdesign. Chapter 10, "Clock Models in Link BER Analysis," covers jitter modeling of the various clocking architectures.

8.2 Statistical Link BER Modeling Framework

This section describes the general mathematical formulation and overall flow of statistical link simulation. The formulation description is presented at a high level to cover the theoretical basis of many variants of statistical link simulation methods. In particular, this section describes the details of ISI and random and deterministic jitter modeling, which is shared by most of the statistical link simulation methods.

8.2.1 Mathematical Formulation

With the linear time invariant (LTI) assumption, the output signal $y(t)$ of a channel at the receiver sampler without transmitter and receiver jitter can be determined by

$$y(t) = \sum_k b_k p(t - kT) + y_{-\infty} = \sum_k (b_k - b_{k-1}) s(t - kT) + y_{-\infty} \tag{8.1}$$

where $p(t)$ is the channel pulse, $s(t)$ is the step response, k is the input symbol index, T is the symbol time, and b_k is the transmit symbol. For the sake of simplicity, assume that $y_{-\infty}$ is zero. For ease of mathematical manipulation, $p(t)$ and $s(t)$ start from zero, and can be derived from SPICE simulation or the S-parameter of the passive channel. Because receiver samplers are only sensitive to voltage input (rather than power), the S-parameter based on voltage waves is used (as

described in Chapter 4, "Network Parameters"). With the transmitter jitter ε_k^{TX}, the output of the channel becomes

$$y(t) = \sum_k (b_k - b_{k-1}) s(t - (\varepsilon_k^{TX} + kT)). \tag{8.2}$$

After sampling at $t = mT + \varepsilon_m^{RX}$, where ε_m^{RX} is the receiver jitter, the sampled signal y_m is determined by

$$y_m = \sum_k (b_k - b_{k-1}) s(\varepsilon_m^{RX} - \varepsilon_k^{TX} + (m - k)T). \tag{8.3}$$

Note that ε^{RX} is not a function of the index k, because it does not alter the transmitted signal, whereas ε^{TX} is. Figure 8.1 illustrates this dependency. As shown in Figure 8.1, the transmitter jitter can be approximated as impulses at the rising and falling edges of a pulse. The receiver jitter can be also approximated as impulses as also shown in Figure 8.1. The difference between these two approximations is that for the transmitter jitter, the two edges at the rising and falling transitions are uncorrelated, whereas for the receiver jitter, these two edges are identical.

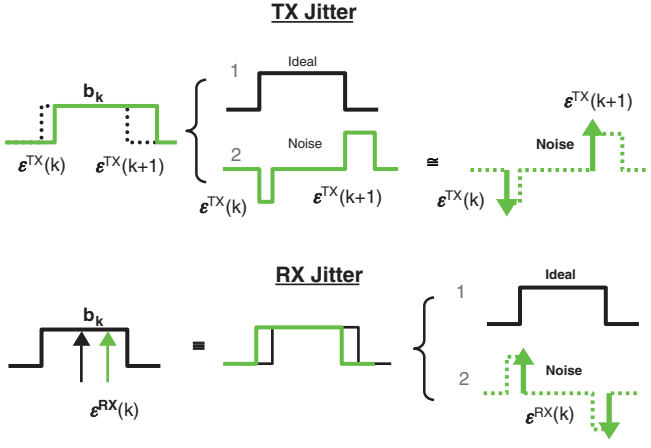


Figure 8.1 Impulse Approximation of Transmitter and Receiver Jitter

Now, the transmitter and receiver jitter is mapped into equivalent voltage noise at the receiver by applying the first-order Taylor series expansion

$$\begin{aligned} y_m &\cong \sum_k (b_k - b_{k-1}) s((m-k)T) - \sum_k (b_k - b_{k-1}) \varepsilon_k^{TX} h((m-k)T) \\ &\quad + \varepsilon_m^{RX} \sum_k (b_k - b_{k-1}) h((m-k)T) \\ &= \sum_k b_k p((m-k)T) + n^{TX} + n^{RX} \\ &= b_m p(0) + \sum_{k \neq m} b_k p((m-k)T) + n^{TX} + n^{RX} \\ &= y^M + y^{ISI} + n^{TX} + n^{RX} \end{aligned} \tag{8.4}$$

where $h(t)$ is the impulse response of the channel, y_m is the received signal without ISI, and y^{ISI} is the voltage noise caused by the ISI of a passive channel. n^{TX} and n^{RX} represent the equivalent voltage noise for transmitter and receiver timing jitter, respectively. Other noise sources, such as thermal noise and crosstalk, can be included in (8.4) as additional terms.

Based on (8.4), compute the BER by adding the error probabilities of the upper and lower eyes

$$\begin{aligned} BER(v_{REF}) &= P(y_m < v_{REF} | b_m = 1) P_1 + P(y_m > v_{REF} | b_m = 0) P_0 \\ &= P(y^{ISI} + n^{TX} + n^{RX} + y^M < v_{REF} | 1) P_1 \\ &\quad + P(y^{ISI} + n^{TX} + n^{RX} + y^M > v_{REF} | 0) P_0 \end{aligned} \tag{8.5}$$

where v_{REF} is the reference voltage (which is typically zero for differential signaling and non-zero for single-ended signaling), and P_1 and P_0 are the probabilities of the input bit being 1 and 0, respectively. In general, the random variables y^{ISI}, n^{TX}, and n^{RX} are correlated, because they are all functions of the symbol pattern and the channel impulse response. The exact calculation of BER must take the correlation between y^{ISI}, n^{TX}, and n^{RX} into account by averaging the error probability over all possible bit patterns. The number of data patterns is strongly related to the signal length required to capture channel response and jitter spectrum. For instance, a lossy backplane channel response can last easily more than 100-bit long for high-speed operation, leading to 2^{100} data patterns! Therefore, this approach is computationally prohibitive due to the large number of combinations. To simplify the computation, assume that the voltage noise terms are independent, unless specified otherwise. With this assumption, the individually calculated probability density functions (PDFs) of y^{ISI}, n^{TX}, and n^{RX} can be convolved to calculate the final PDF of the system error. Subsequent sections describe the modeling details of y^{ISI}, n^{TX}, and n^{RX}.

Generally, noise sources in high-speed links are either bounded (such as power supply noise) or Gaussian (such as thermal noise). n^{TX} and n^{RX} can be represented as the sum of two random variables: $\zeta^{Bounded}$ and $\zeta^{Gaussian}$. Then, (8.5) is written as

$$BER(v_{REF}) = P(\zeta^{Gaussian} + \zeta^{Bounded} + y^M < v_{REF}|1)P_1 + P(\zeta^{Gaussian} + \zeta^{Bounded} + y^M > v_{REF}|0)P_0. \quad (8.6)$$

With the probability mass function (PMF) of $\zeta^{Bounded}$, we can rewrite the preceding equation as follows:

$$BER(v_{REF}) = P_1 \sum_v P(\zeta^{Gaussian} < v_{REF} - y^M - v|1)P(\zeta^{Bounded} = v) + P_0 \sum_v P(\zeta^{Gaussian} > v_{REF} - y^M - v|0)P(\zeta^{Bounded} = v). \quad (8.7)$$

Using the Q-function for the cumulative distribution function of the Gaussian random variable, the final expression for the BER calculation is

$$BER(v_{REF}) = \sum_v \left(P_1 Q\left(\frac{v + y^M - v_{REF}}{\sigma} \right) + P_0 Q\left(\frac{-v - y^M + v_{REF}}{\sigma} \right) \right) P(\zeta^{Bounded} = v). \quad (8.8)$$

Here, σ is the variance of $\zeta^{Gaussian}$, and $Q(x)$ is the Q-function.

As shown in Equation (8.4), EVN does not model all the receiver-side jitter components. However, any remaining receiver-side jitter can be modeled afterwards. For instance, in a serial link application with a CDR, the CDR dither is modeled as a receiver statistical sampling distribution (as described in Chapter 10). Given the sampling distribution P^{RX}, the overall system BER is the sum of the conditional BERs at each phase

$$BER = \sum_\phi BER_\phi(v_{REF}|\phi)P^{RX}(\phi) \quad (8.9)$$

where ϕ is the phase index and BER_ϕ is the link BER at ϕ.

The BER eye diagram, shown in Figure 8.2, is calculated by sweeping Equation (8.8) over different sampling phases and reference voltage offsets. The timing bathtub is obtained by taking the horizontal slice at the target reference voltage, whereas the voltage bathtub is obtained from the vertical slice at the fixed phase. However, these bathtub curves from simple slices of a BER eye diagram assume an ideal receiver, which samples at a fixed location without jitter. In practice, a probability distribution similar to P^{RX} in (8.9) describes the receiver sampling distribution. Consequently, the bathtub curves derived from measurement generally look different from the bathtub curves directly extracted from a BER eye. To capture the impact of sampling uncertainty on bathtub curves, we need to use the conditional BER calculation described in Equation (8.9), but with different phase offset values (as will be described in Section 8.2.3). Figure 8.3 compares two different bathtubs: one from a single horizontal slice of the BER eye diagram, and the other

one after integrating receiver sampling distribution. As expected, the bathtub curve-timing margin is reduced, after accounting for the receiver sampling uncertainty.

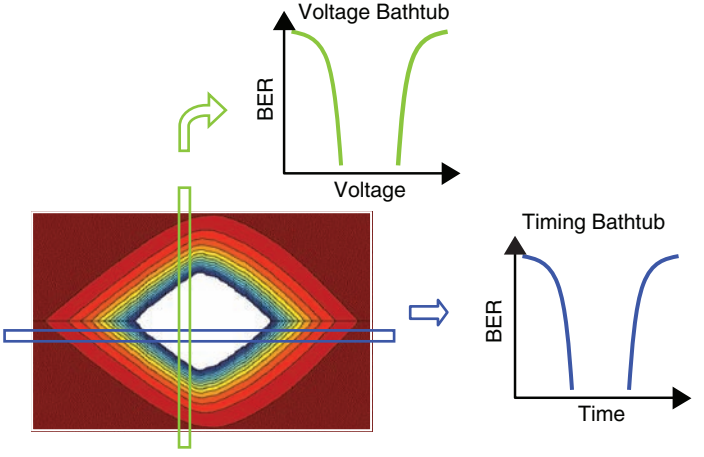


Figure 8.2 The BER Eye Diagram of a Typical Lossy Channel

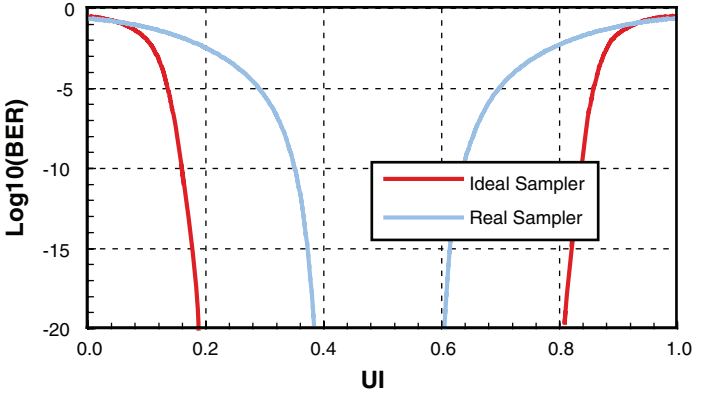


Figure 8.3 System-Level Timing Bathtub Curves with an Ideal vs. Realistic Sampler

8.2.2 Notes on the Equivalent Voltage Noise Model

The previous section described the general expression for calculating the system-level BER with random jitter. Before moving on to the next section that describes the statistical simulation framework, a few comments on jitter modeling choices and accuracy issues are given as follows. Two

choices exist for modeling the receiver-side timing jitter. The first choice is the equivalent voltage noise concept described in (8.4). The other choice is to model the jitter as the receiver sampling distribution using conditional probability, as shown in (8.9). The same approach can be applied to the transmitter jitter as long as the transmitter jitter has only low-frequency content (see Figure 8.1). When the transmitter jitter in Figure 8.1 is low frequency, the impulse at the rising edge is the same as the impulse at the falling edge, mimicking the receiver jitter impulses in Figure 8.1.

The main advantage of the equivalent voltage noise model is that it can accurately account for the channel coloring of the transmitter jitter. For instance, it can account for jitter amplification due to channel loss. However, because this model is based on a Taylor-series approximation, it loses accuracy when the jitter amplitude increases. For this reason, low-frequency jitter should be modeled as the receiver sampling distribution, rather than the equivalent voltage noise. When the transmitter jitter has both low- and high-frequency contents, modeling these components separately is more accurate, by treating the high-frequency component as the transmitter jitter and the low-frequency component as the receiver jitter. When transmitter jitter is affected by the channel, it is considered high frequency. The transmitter DCD is the highest jitter component.

Figure 8.4 compares the margin loss due to various amounts of Gaussian jitter, using EVN, the Rx sampling distribution method, and the time-domain simulation. As expected, the margin loss predicted by the time-domain simulation increases as more bits are simulated (10^5 vs. 10^8). To capture the impact of RJ at low BER (such as 10^{-15}), time-domain simulation requires too many bits. The Rx sampling distribution method is the most optimistic, because it ignores the jitter amplification of the passive channel. The EVN method efficiently captures both jitter amplification and the tail statistics of RJ, and so predicts the worst margin loss.

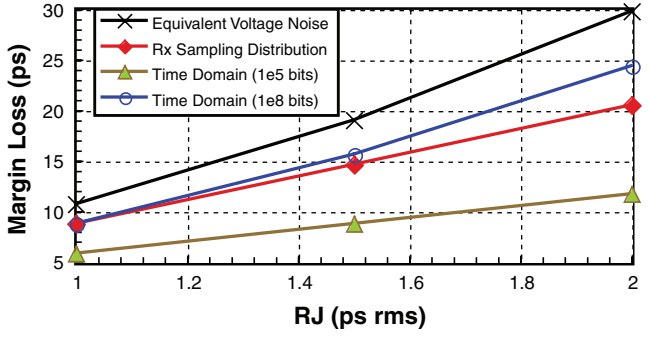


Figure 8.4 Comparison of Equivalent Voltage Model, Receiver Sampling Distribution, and Transient Simulation of White Gaussian Transmitter Jitter

8.2.3 Overall Statistical Link Simulation Flow

This section presents a general simulation overview and flowchart. Figure 8.5 illustrates a step-by-step procedure to compute BER, which is described next.

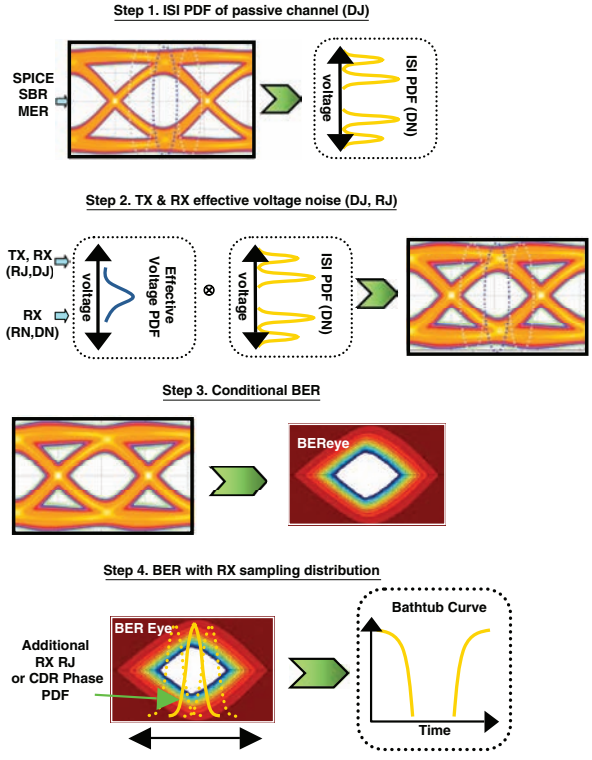


Figure 8.5 Flowchart of BER Calculation Using a Statistical Eye

The first step is to characterize the ISI PDF of a passive channel, which can be calculated different ways. SPICE simulation is a brute-force way to compute ISI PDF. One can also use fast transient simulation techniques, based on the superposition of single-bit or edge responses to compute this PDF more efficiently. Chapter 9 describes these simulation techniques, along with an acceleration technique used to estimate the bound of PDF, to further speed up the calculation. Section 8.3.1 describes a more commonly used approach based on the convolution of the ISI cursors in a single-bit response. This technique can be extended to the rising and falling edge responses [16].

The second step is to compute the equivalent voltage noise due to the transmitter and receiver jitter. As mentioned previously, receiver jitter (and some components of transmitter jitter) can be modeled as the receiver sampling distribution (which is handled in the fourth step). After the equivalent voltage noise is obtained, the noise is convolved with the previously computed ISI PDF to obtain the final PDF.

The third step involves computing a 2-D BER eye diagram, based on (8.8).

For the fourth and final step, the final BER of the system is calculated by taking into account the receiver sampling distribution based on (8.9). The timing bathtub is calculated by sweeping the sampling distribution offset.

8.3 Intersymbol Interference Modeling

This section describes the ISI PDF modeling of passive channels based on a commonly used convolution technique. The first part of this section describes the ISI PDF based on a single-bit response (SBR). The second part of this section extends this approach to DCD modeling. However, this convolution approach fails when the non-linearity of the system is severe, or when the data pattern is non-white. (Chapter 9 describes a time-domain approach that overcomes these limitations, but at the expense of increased simulation time.)

8.3.1 ISI PDF Calculation Based on Single-Bit Response

When an ideal pulse propagates along a lossy transmission line, the pulse distorts (for example, becomes widened and attenuated) and creates several non-zero residues at before and later sampling points, due to attenuation and reflection. Figure 8.6 illustrates this using a parallel I/O channel. The cursor points represent data and edge-sampling locations. In this example, the residues at the data sampling locations are small, because the channel is short. ISI PDF represents the noise distribution resulting from these residues.

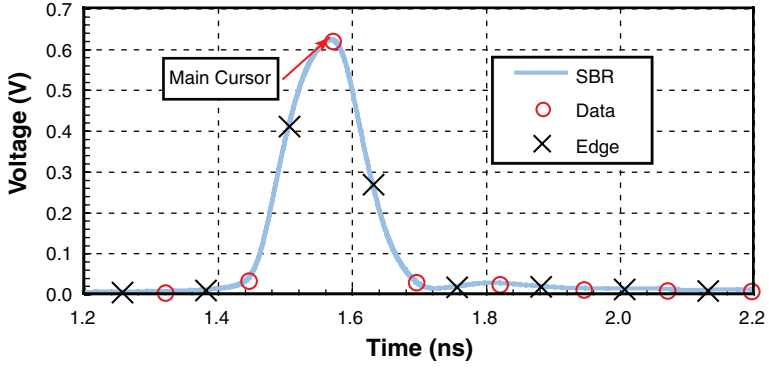


Figure 8.6 The Single-Bit Response of an On-Board Parallel Bus with Equalization

Now, consider three ISI cursors to demonstrate the ISI PDF calculation process. Figure 8.7 illustrates the convolution process using these three cursors. This process assumes a random bit pattern (in other words, there is an equal probability that any individual bit will be either a 1 or a 0), and that the bits are uncorrelated. Based on this assumption of equal probability, the ISI distribution from the first post cursor (30mV) is either –30mV or 30mV, with an equal probability of 0.5, as shown in Figure 8.7. Because the bits are uncorrelated, the ISI distributions from different cursors are convolved to derive the final ISI distribution. As shown in Figure 8.7, we start with two ISI cursors and perform the convolution to obtain four cursors. Then, we add additional cursors and perform convolution with the previous cursors. The final ISI distribution consists of eight voltage locations, each with an equal probability of occurrence (which is 0.125).

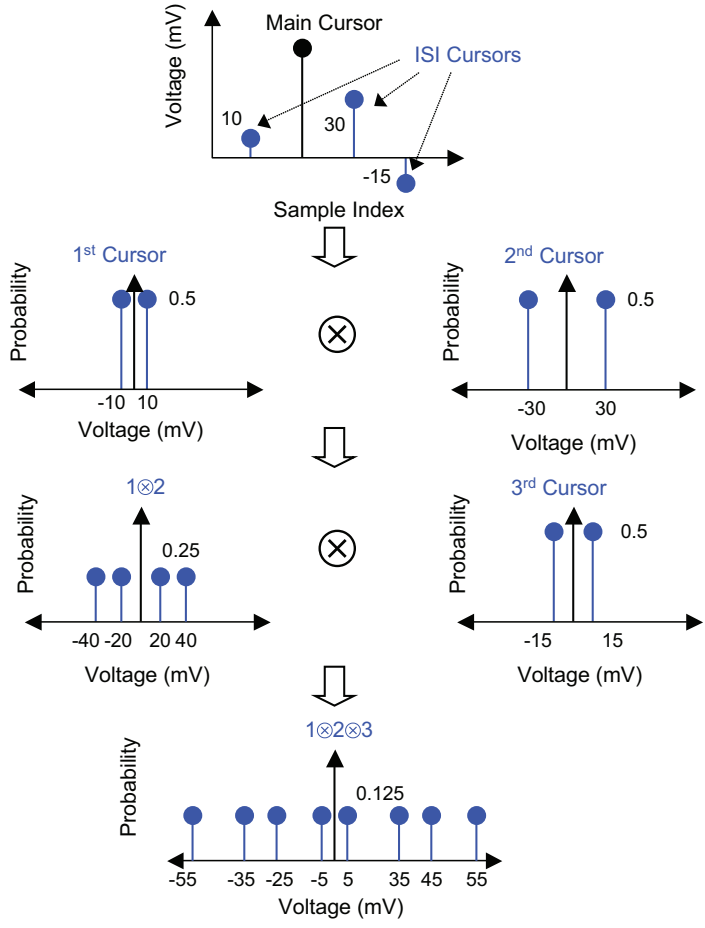


Figure 8.7 ISI PDFs Calculation Flow for Three Cursor Cases

This process can be quite time consuming when the number of cursors is large. Typical single-bit responses are often more than 100 bits long. A simple way to reduce this calculation time is to bin the cursors after each convolution. However, the voltage bin size must be sufficiently small to avoid any accumulation errors during multiple convolution steps. A significant amount of time can be saved by applying the divide-and-conquer approach as described by Sanders, Resso, and D'Ambrosia [4]. With this approach, the original *N*-cursor problem is divided into two *N*/2 problems and then computes the ISI PDF for individual *N*/2 problems. The final distribution is calculated by convolving these two ISI PDFs. This approach can be generalized to a multilevel divide-and-conquer problem, but, in practice, the optimum performance is achieved using only a few levels.

Note that at different phase locations (that is, sampling points), the amplitude of the main and ISI cursors are different. Figure 8.8 illustrates ISI distributions using the data sampling locations and the transient edge locations (which are offset 90 degrees from data). By adding the main cursor and ISI PDF at different phase locations, a complete eye diagram is constructed, as shown in Figure 8.9.

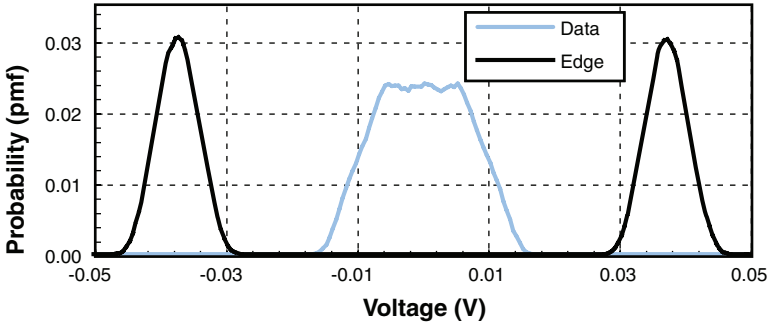


Figure 8.8 ISI PDFs for Data and Edge Sampling Locations

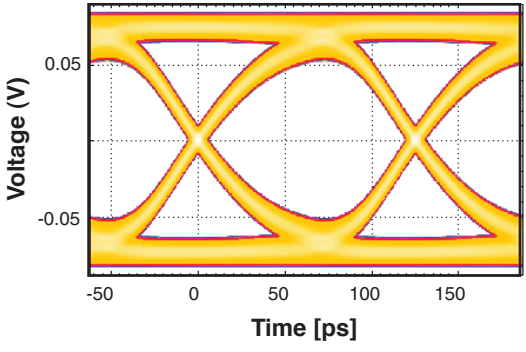


Figure 8.9 Eye Diagram Obtained from ISI PDFs

Extending this method to handle coupled transmission line systems is straightforward: One can excite an aggressor line with a single bit and observe the signal response at the end of the victim line. (The only difference between this and the previous case is that all the cursors including the main one are noise and must be included in the convolution process.) The final ISI PDF for the coupled line system is obtained by convolving the crosstalk and ISI PDFs.

8.3.2 Transmitter Duty-Cycle Distortion (DCD) Modeling

Non-idealities, such as asymmetric rising and falling edges of the clock path, result in deterministic jitter called *duty-cycle distortion* (DCD). Compared with other jitter components, DCD can

be particularly detrimental, because it directly modulates the width of the transmitted pulses, which may be significantly amplified by the channel. As shown by Oh, et al. [3], transmitter DCD causes odd and even bits to have different bit widths and channel responses. If the even bit is assumed to be shorter, it produces a smaller swing and eye than the nominal case. The wider odd bit creates bigger ISI, which exacerbates the reduced eye of the even bit. Compared to TX DCD, RX DCD is usually less detrimental. RX DCD shifts the data and edge sampling locations for alternate bits. The modified sampling locations impact both the adaptation of the equalization, and the CDR phase probability distribution.

TX DCD can be modeled in a manner similar to the way random jitter is treated (described later in Section 8.4.1 using the EVN model). TX DCD can be treated as impulses at the edges of the ideal waveform as shown in Figure 8.10; however, the resulting model is limited to small DCD values. This section reviews a more rigorous approach that is applicable to large DCD values [3]. This approach captures the DCD impact on channel characteristics by separately computing the SBRs for the odd and even bits. Figure 8.11 shows the SBRs for a sample channel with 10% transmitter DCD.

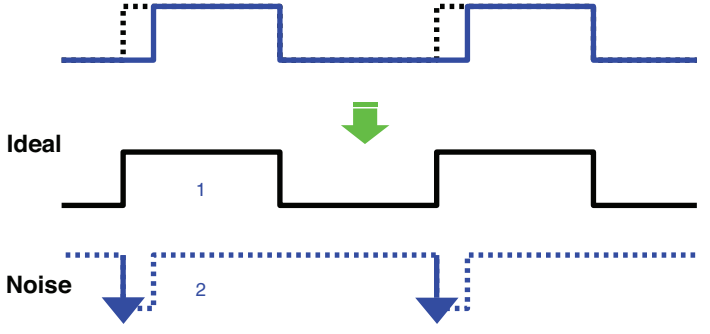


Figure 8.10 Impulse Representation of Transmitter DCD

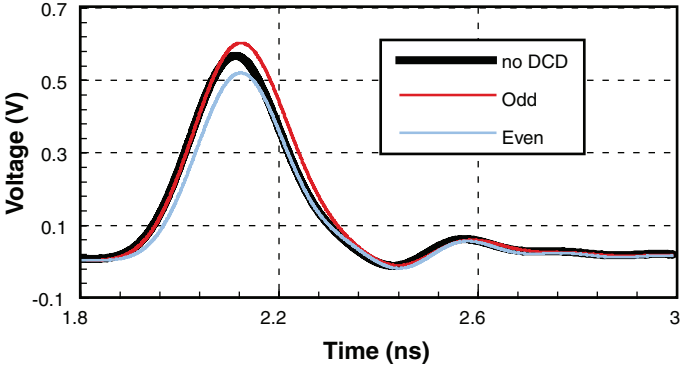


Figure 8.11 Single-Bit Responses for Odd and Even Bits with 10%Transmitter DCD and no DCD

Using the odd and even bit responses, the received signal is calculated by simply shifting and adding the corresponding single-bit responses. For example, the channel response to an input sequence b_k with the b_0 bit at the even bit time is obtained by

$$y_m = \sum_k b_{2k-1} p^{odd}((m - 2k - 1)T) + \sum_k b_{2k} p^{even}((m - 2k)T) \tag{8.10}$$

where p^{even} and p^{odd} are the odd and even bit responses, respectively. Equation (8.10) shows that the ISI contributions from the other bits, as applied to the current bit, are interleaved among the odd and even bits. Therefore, we compute the ISI PDFs for the odd and even bits by first interlacing the odd and even ISIs in time, and then computing the PDFs as usual. Figure 8.12 shows the ISI PDFs for an odd bit, an even bit (10% transmitter DCD), and an ideal bit (no DCD).

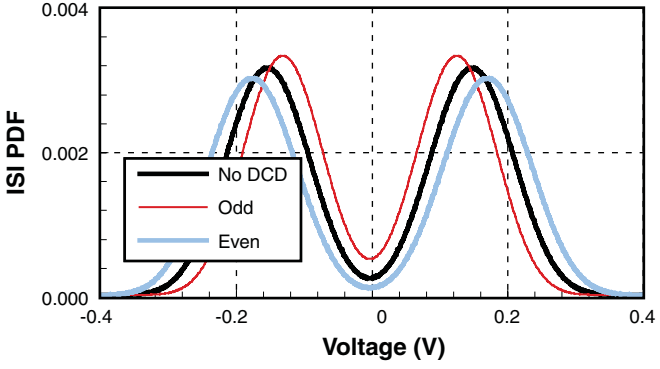


Figure 8.12 ISI PDFs for Odd and Even Bits with 10% Transmitter DCD and no DCD

In the presence of DCD, the receiver sees two different eyes: one for the odd bits, and one for the even bits. After computing the individual odd and even BER eyes, based on the ISI calculations presented earlier, the final BER contour (shown in Figure 8.13) is generated by averaging the odd and even eyes. Note that the worst case among the odd and even eyes defines the performance of the link. As expected, the even eye is much worse than the odd eye, due to a smaller pulse width and a larger ISI impact. In this example, the even eye primarily determines the final overall link performance.

8.4 Transmitter and Receiver Jitter Modeling

In the previous section, the statistical simulation of passive channels is considered. This section considers device jitter due to the transmitter and receiver. Many of the published statistical

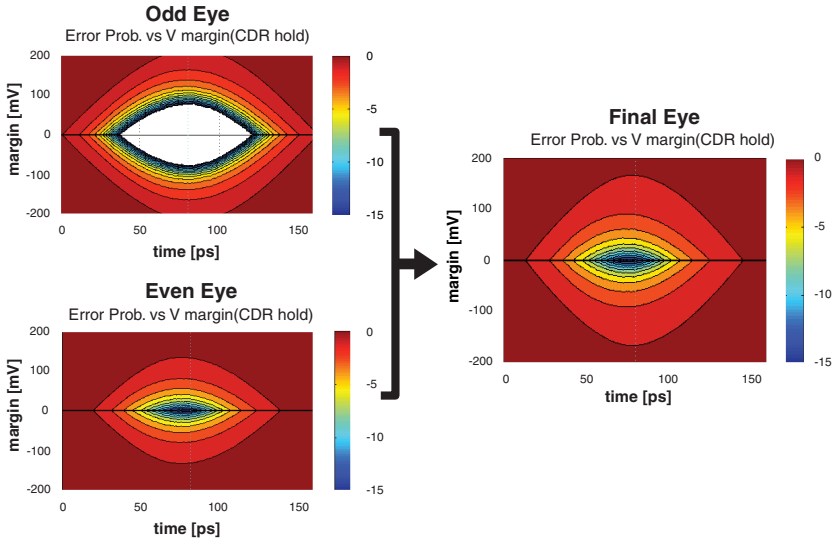


Figure 8.13 BER Calculation in Presence of DCD

methods share a common approach to passive channel modeling, but differ in their modeling of device jitter. Most of them ignore the jitter spectrum [4] [6] [7] [10] [15], whereas Stojanovic and Horowitz's work [1] models colored Gaussian random jitter. This section describes the methods that can handle various types of jitter, including colored bounded jitter [1] [3].

8.4.1 Transmitter Jitter Modeling

Transmitter jitter modeling is complicated, because the equivalent voltage noise, n^{TX}, due to the transmitter jitter, is colored by the channel impulse response, as shown in Equation (8.4). Modeling transmitter jitter as simple receiver jitter will result in significant errors, especially for high-frequency jitter components. To derive the model for n^{TX}, we rewrite the transmitter jitter in (8.4) as

$$\begin{aligned} n^{TX} &= -\sum_{k}(b_k - b_{k-1})\varepsilon_k^{TX}h((m-k)T) \\ &= -\sum_{k=N_{pre}}^{-N_{post}} b_k(\varepsilon_k^{TX}h_{m-k} - \varepsilon_{k+1}^{TX}h_{m-k-1}) \\ &= \vec{a}^T\mathbf{W}\mathbf{H}^{TX}\vec{\varepsilon}^{TX} \end{aligned} \tag{8.11}$$

where $h_k = h(kT)$, and $\vec{a}$ is a vector of random transmit symbols, $\mathbf{W}$ is constructed from the transmitter equalizer taps $\vec{w}$,

$$\mathbf{W} = \begin{bmatrix} \vec{w} & 0 & ... & 0 \\ 0 & \vec{w} & 0 & ... \\ ... & 0 & \vec{w} & 0 \\ 0 & ... & ... & \vec{w} \end{bmatrix} \tag{8.12}$$

and $\vec{a}^T\mathbf{W}$ is the output of the transmitter equalizer. $\mathbf{H}^{TX}$ is constructed from h_k:

$$\mathbf{H}^{TX} = - \begin{bmatrix} -h_{-N_{pre}-1+m} & h_{-N_{pre}+m} & & & 0 \\ & -h_{-N_{pre}+m} & h_{-N_{pre}+1+m} & & \\ & & & ... & \\ 0 & & & -h_{N_{post}+m-1} & h_{N_{post}+m} \end{bmatrix}. \tag{8.13}$$

Here, N_{pre} and N_{post} are the numbers of the pre- and post-cursors, in the channel impulse response. There is no general formulation for calculating the PDF of n^{TX} for deterministic jitter. However, if ε^{TX} is sufficiently low frequency, such that all jitter terms in $\vec{\varepsilon}^{TX}$ are similar, ε^{TX} can be treated as receiver jitter. The dominant high-frequency transmitter jitter is DCD (refer to the ISI PDF discussion in Section 8.3.2).

The rest of this section is devoted to uncorrelated jitter.

When ε_m^{TX} is either white, or colored unbounded Gaussian, we can compute the effective sigma using autocorrelation, as follows [3]:

$$[\mathrm{tr}(V_{\vec{a}}\mathbf{W}\mathbf{H}^{TX}R_{\varepsilon}^{TX}\mathbf{H}^{TX^T}\mathbf{W}^T)]^{0.5} \tag{8.14}$$

where R_{ε}^{TX} is the autocorrelation matrix of ε^{TX} and tr is the trace operator. If the transmitter jitter is white, and $b_k - b_{k-1}$, in (8.11), is assumed to be an independent random variable for each k, then the summation terms in (8.11) are independent; so the final PDF is computed by convolving the summation terms [17].

On the other hand, if the transmitter random jitter ε^{TX} is bounded and colored, it can be first approximated as a white discrete random process, filtered by a coloring filter h_{color}, as follows [18]:

$$\varepsilon_k^{TX} = \sum_n h_{color}(n)\varepsilon_w(k-n) = \vec{h}_{color}^T\vec{\varepsilon}_W \tag{8.15}$$

where $\vec{\varepsilon}_W$ is a white random variable vector. Substituting (8.15) into (8.11), we have

$$n^{TX} = \vec{a}^T \mathbf{W}\mathbf{H}^{TX} \begin{bmatrix} \vec{h}_{color}^T & \underline{0} & ... & \underline{0} \\ 0 & \vec{h}_{color}^T & ... & \underline{0} \\ 0 & ... & \vec{h}_{color}^T & \underline{0} \\ 0 & ... & 0 & \vec{h}_{color}^T \end{bmatrix} \vec{\varepsilon}_W.$$

$$= \vec{a}^T \mathbf{M}\vec{\varepsilon}_w. \tag{8.16}$$

A brute-force approach to computing the PDF of n^{TX} is to average all the conditional PDFs with respect to $\vec{a}$. However, this approach is impractical, because the size of $\vec{a}$ can be very large due to the long channel impulse response time. To speed up this PDF calculation, $\mathbf{M}$ is decomposed using the singular value decomposition [3]:

$$n^{TX} = \vec{a}^T \mathbf{M}\vec{\varepsilon}_W = \vec{a}^T \mathbf{U}\mathbf{\Lambda}\mathbf{V}\vec{\varepsilon}_W = \sum_n (\vec{a}^T \vec{u}_n)\lambda_n(\vec{v}_n^T \vec{\varepsilon}_W) = \sum_n X_n. \tag{8.17}$$

In (8.17), $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices whose n^{th} column and row are $\vec{u}_n$ and $\vec{v}_n^T$, respectively. Matrix Λ is diagonal and its diagonal entries are $[\lambda_1,...,\lambda_{\text{Rank of } \mathbf{M}}]$. This equation expresses n^{TX} as the sum of dependent random variables X_n with variance $V_\varepsilon \lambda_n^2$. The variance of n^{TX} is $V_\varepsilon \sum \lambda_n^2$ where λ_n is the singular value of $\mathbf{M}$ and V_ε is the variance of $\vec{\varepsilon}_w$.

To simplify computation, X_n is assumed to be independent. Then, the PDF of n^{TX} is calculated by convolving only N PDFs of X_n. Note that the variance of the convolved PDFs is still $V_\varepsilon \sum \lambda_n^2$ (the same as the variance of n^{TX}). This approximation usually results in a slightly higher BER, and is used as an estimate of the PDF of n^{TX}. The largest singular value, λ_1, can be also used to produce a second approximation that usually yields a slightly lower BER. The difference between the two approximations provides a guideline with which to gauge the precision of the n^{TX} PDF estimates: A large difference between the approximations indicates a poor PDF estimate. In typical lossy channels, one or two singular values dominate the $\mathbf{M}$ matrix in (8.17). Figure 8.14 illustrates the impulse response for a relatively lossy channel. Figure 8.15(a) shows the corresponding distribution of singular values. As shown in this figure, there are only a few dominant singular values; the rest of the singular values are close to each other.

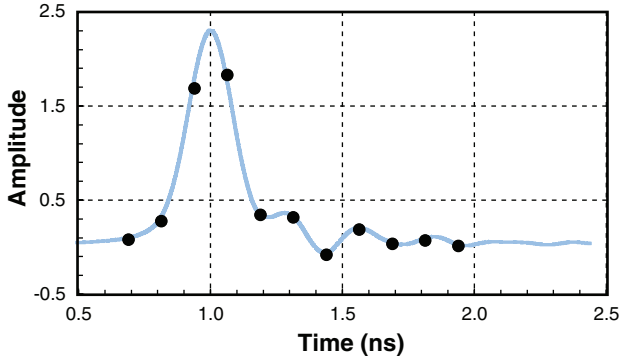


Figure 8.14 The Channel Impulse Response and Its Data-Rate Sampled Sequence

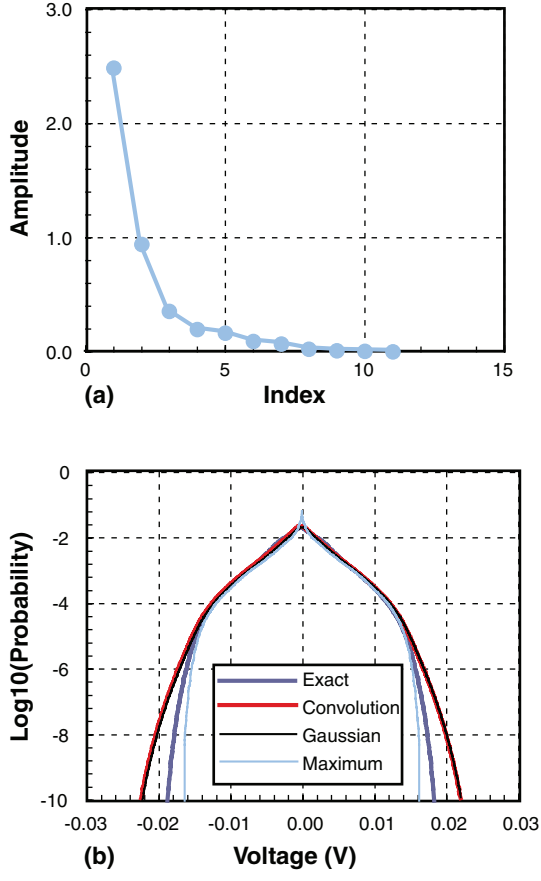


Figure 8.15 (a) Distribution of λ_n and (b) PDFs of Equivalent Voltage Noise Using Different Methods at the Eye Center Phase

To speed up the computation, a single Gaussian random variable is used to approximate all of X_n that have small variances. Figure 8.15(b) compares PDFs calculated using different methods for the sampled impulse response shown in Figure 8.14. The Convolution line is the convolved PDFs of X_n. The Gaussian line is based on the approximate PDF, where the bounded PDF, using the first two dominant λ_n, is convolved with a Gaussian PDF representing other small singular values. The Exact line is the exact PDF using the brute-force method. The Maximum line is the PDF using the maximum λ_n. The approximated PDFs capture the shape of the high BER portions of the exact PDF. As expected, the low BER tails of the approximated PDFs exhibit some deviation from the target due to the various simplification assumptions made previously, including assuming independence between X_n. The Gaussian approximation closely matches the PDF using all singular values, thereby demonstrating that it can be used to replace small singular values. Figure 8.16 compares system-level margins, using different PDF approximations. Gray curves are calculated using the largest singular value approximation, whereas black curves are calculated using the independence approximation. Both results showed a good match, indicating that the estimation is very accurate.

8.4.2 Receiver Jitter Modeling

As illustrated in Figure 8.5, the statistical framework, described in Section 8.2.3, separates passive channel and device jitter components. The ISI PDF and the equivalent voltage noise PDF distributions are independently calculated and convolved together to obtain the final PDF of total noise. Starting with the receiver-side jitter, the equivalent voltage noise of receiver jitter in (8.4) can be rewritten as

$$n^{RX} = \varepsilon^{RX} \sum_k b_k (h_{m-k} - h_{m-k-1}) = \varepsilon^{RX} \vec{a}^T \mathbf{W} \vec{H}^{RX} \tag{8.18}$$

where h_m is the symbol-spaced channel impulse response, $\vec{H}^{RX}$ is $[...,h_{m-k} - h_{m-k-1},...]^T$. The variance of n^{RX} is given by

$$E\left((n^{RX})^2\right) = E\left((\varepsilon^{RX}\vec{a}^T \mathbf{W} \vec{H}^{RX})^T \varepsilon^{RX} \vec{a}^T \mathbf{W} \vec{H}^{RX}\right) = V_a V_\varepsilon (\vec{H}^{RX})^T \mathbf{W}^T \mathbf{W} \vec{H}^{RX} \tag{8.19}$$

where V_ε is the receiver jitter variance, and V_a is the variance of the transmit symbols.

Because n^{RX} is a function of two independent random variables, $\vec{a}$ and ε^{RX}, its PDF is calculated by

$$P(n^{RX}) = P(\varepsilon^{RX}) P(\vec{a}^T \mathbf{W} \vec{H}^{RX}). \tag{8.20}$$

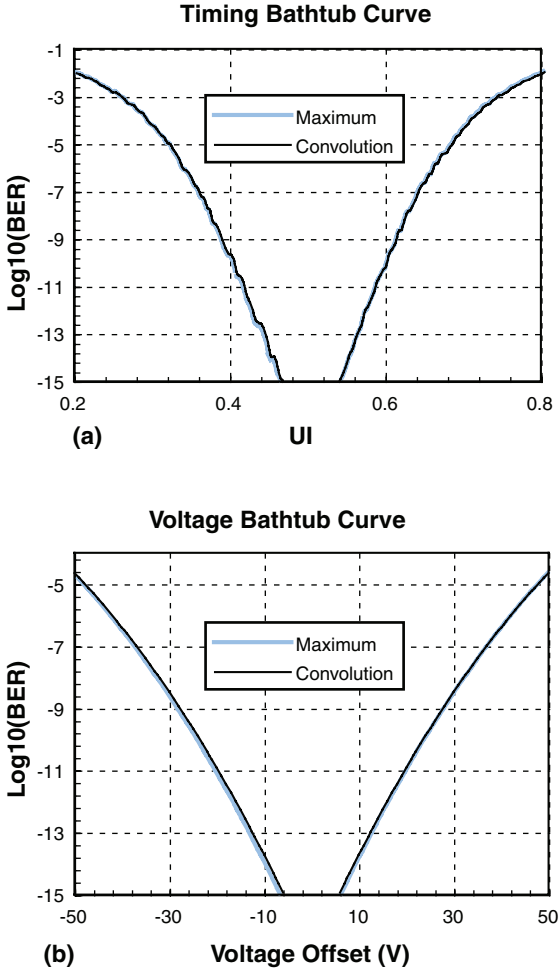


Figure 8.16 The Timing and Voltage Bathtubs Calculated Using Different PDFs from Figure 8.15(b)

To calculate $P(\vec{a}^T \mathbf{W} \vec{H}^{RX})$, the convolution method, used for ISI PDF calculation in Section 8.3.1 can be used. When ε^{RX} is bounded, the final PDF of n^{RX} is obtained by multiplying $P(\vec{a}^T \mathbf{W} \vec{H}^{RX})$ with $P(\varepsilon^{RX})$. On the other hand, when ε^{RX} is unbounded Gaussian, it can be approximated as a single Gaussian random variable with the same variance.

8.4.3 Receiver and Transmitter Jitter Examples

To demonstrate the equivalent voltage noise method, a simple lossy channel (shown in Figure 8.17) is considered with a data rate of 10Gb/s. The channel has roughly 10dB loss at 5GHz. 3ps of input Gaussian jitter is applied for both the transmitter and receiver cases. The EVNs for the

transmitter and receiver are calculated, and shown in Figure 8.18. The center vertical slices (at time 0) of the plots are the PDFs of EVNs at the data sampling location. The transmitter jitter clearly results in more EVN than the receiver jitter. This is because, unlike transmitter jitter, receiver jitter is not amplified by the loss of the passive channel. The voltage noise is calculated by multiplying the input jitter by the signal slope (as demonstrated in Chapter 10). Jitter amplification of the transmitter jitter due to the loss of the passive channel is one of the main performance-limiting factors for a forwarded clocking architecture. Chapter 10 discusses this issue in depth.

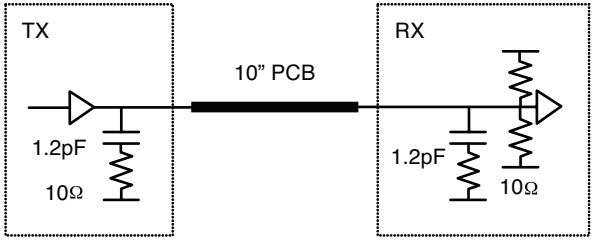


Figure 8.17 A Lossy Channel to Demonstrate EVN Method

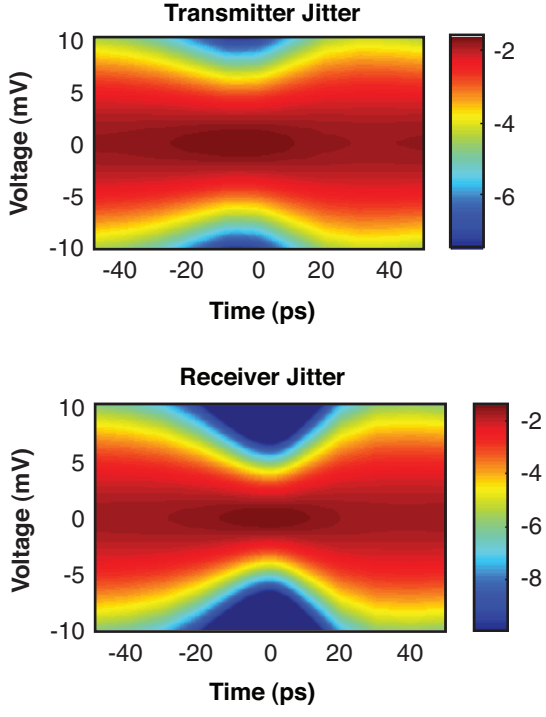


Figure 8.18 EVN Noise PDFs Over One Bit Time for Transmitter and Receiver Jitter

8.4.4 Device Jitter Simulation Validation

This section validates the jitter simulation methodology using a clock-forwarding parallel I/O interface. The simulation and lab environment comprises a 6" PCB link on a socket-based system-test board, running at 5Gb/s (see Figure 8.19). One can directly measure the parameters for the transmitter jitter distribution using an Agilent DCA-J. The parameters for the receiver jitter distribution are obtained by differentiating measured cumulative sampling distributions. Figure 8.20(a) shows the TX and RX jitter distributions. Because the test system is synchronous, the measured jitter for clock and data directly impacts the final sampling distributions. Incorporating a previously correlated S-parameter channel model, the statistical simulation produced a reasonably good estimation of actual link performance, as shown in Figure 8.20(b). The mismatch in the high BER region of the bathtub curve may be because individual circuit components, such as PLL noise, clock path, and phase nonlinearity of TX and RX, are not modeled in detail.

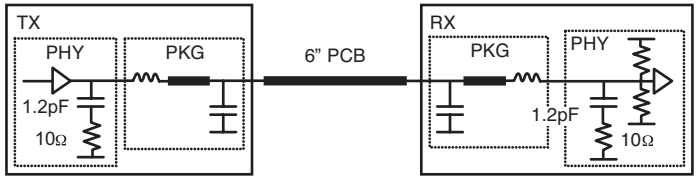


Figure 8.19 Sample Parallel I/O Link Set-Up

8.5 Periodic Jitter Modeling

One of the most dominant device jitter components in high-speed I/O devices is power supply noise–induced jitter (PSIJ) in the clock signal. As described in Chapter 14, "Supply Noise and Jitter Characterization," the PSIJ spectrum consists of distinct peaks at the reference clock frequency and its harmonics, in addition to the noise floor. The EVN concept, described in the previous section, can be used to model the impact of the noise floor accurately. However, for the periodic jitter (the distinct peaks in the PSIJ spectrum), the EVN method presented earlier is no longer accurate, as the noise components n^{ISI}, n^{TX} and n^{RX} in (8.5) are assumed to be independent.

Another approach is to use transient data to curve fit the statistical model, based on the dual-Dirac distribution [19]. Periodic jitter modeling is not specifically described by Balamurugan, et al. [19], because all jitter components are combined together into transient simulation, and curve fitting cannot distinguish the differences. Rao, Borich, Abebe, and Yan describe a more rigorous approach [20], but it handles a single-tone jitter only at the transmitter side.

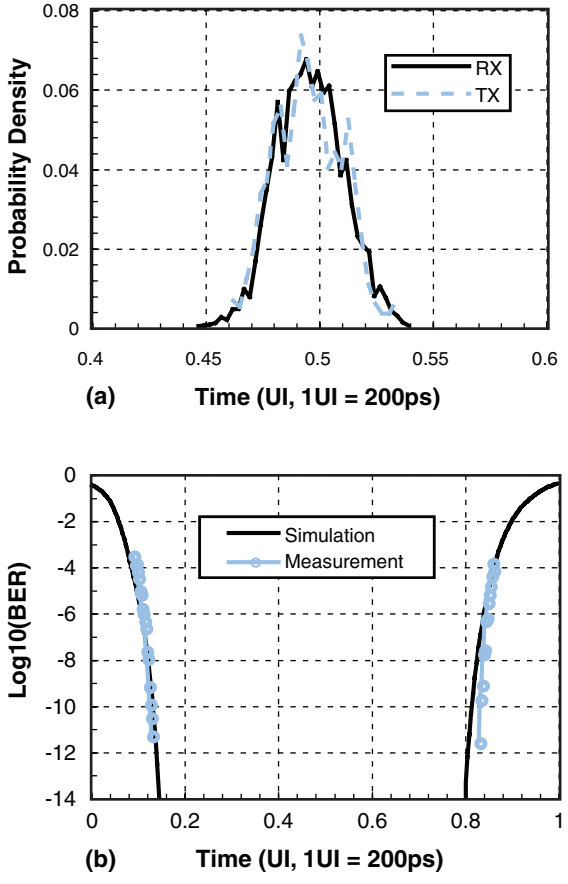


Figure 8.20 FlexIO (a) TX and RX Timing and (b) Bathtub Measurements

In this section, a more rigorous EVN model is derived [21]. This model extends the equivalent voltage noise concept from random jitter to periodic. In general, the impact of periodic jitter must be calculated along with channel ISI. This section also provides a comparison of this new model to the previous EVN approach, which ignores the jitter interaction with ISI.

8.5.1 Formulation Periodic Jitter with ISI Interaction

Because receiver jitter is not impacted by channel, it is naturally not correlated with channel ISI. This makes the previous EVN formulation, based on independent PDF models for channel and device jitter, still valid for the periodic jitter at the receiver. Alternatively, the receiver jitter can also be modeled as a sampling jitter distribution, making receiver-side periodic jitter modeling straightforward.

Here, the focus is on modeling transmitter periodic jitter. Because both periodic jitter and ISI are deterministic, treating them independently by convolving their noise PDFs without considering their correlation is pessimistic. To account for this correlation, we rewrite (8.4) using (8.11) as follows:

$$\begin{aligned} y_m &\cong y^M + y^{ISI} + n^{TX} \\ &= y^M + \vec{a}^T\vec{v}^{ISI} + \vec{a}^T\mathbf{M}_{TX}\vec{\varepsilon}_{TX} \\ &= y^M + \vec{a}^T\left(\vec{v}^{ISI} + \mathbf{M}_{TX}\vec{\varepsilon}_{TX}\right) \\ &= y^M + \vec{a}^T\vec{v}^{total} \\ &= y^M + n^{total} \end{aligned} \tag{8.21}$$

where the ISI vector, $\vec{v}^{ISI}$, is the symbol-spaced single-bit response with the main cursor set to 0 (only the ISI cursors are kept), $\vec{\varepsilon}_{TX}$ is the periodic transmitter jitter sequence, and $\mathbf{M}_{TX}\vec{\varepsilon}_{TX}$ computes the equivalent voltage noise sequence caused by $\vec{\varepsilon}_{TX}$. Figure 8.22 illustrates how the ISI vector is combined with the EVN sequence to generate the total voltage noise sequence $\vec{v}^{total}$. The final voltage noise, $n^{total} = \vec{a}^T\vec{v}^{total}$, includes the contributions from ISI and transmitter jitter.

In general, calculating n^{total} for an arbitrary jitter type is impossible, as it may require too many combinations of voltage noise sequence $\vec{v}^{total}$. Fortunately, it can be done effectively for a periodic jitter sequence, because only a limited number of combinations are possible. This is illustrated using sinusoidal jitter in Figure 8.21. For the sake of simplicity, consider a short channel whose single-bit response only lasts five bit times. Different EVN are calculated depending on the location of the main bit in the jitter sequence. Figure 8.22 shows three different jitter sequences, $\vec{\varepsilon}_1^{TX}$, $\vec{\varepsilon}_2^{TX}$ and $\vec{\varepsilon}_3^{TX}$. Based on (8.21), each jitter sequence is multiplied by the convolution matrix $\mathbf{M}$ to generate an EVN sequence. The EVN sequence is then added to the ISI vector, $\vec{v}^{ISI}$, to generate the total voltage noise sequence $\vec{v}^{total}$. For different jitter sequences, $\vec{v}^{total}$ is different. Therefore, the PDFs of n^{total} are different for different jitter sequences, as shown in Figure 8.22. As different periodic jitter sequences are equally probable, the final total voltage noise PDF is simply the average of the total voltage noise PDFs. For the link BER calculation, the PDF of the final overall voltage noise n^{total} can be treated similar to the ISI PDF in Section 8.3.1.

When the jitter frequency is not a harmonic or sub-harmonic of the data rate, or there are multiple tones, the possible number of PDFs to be averaged could be quite large. To reduce computational time, a few total voltage noise sequences can be selected, which result in the largest bounds to calculate PDFs. The bound of a noise sequence is the sum of the absolute noise values. Although

this approximation generally results in a pessimistic estimation, it provides good accuracy for low BERs, such as 10^{-12}, because the boundary values of PDF dominate the low BER region.

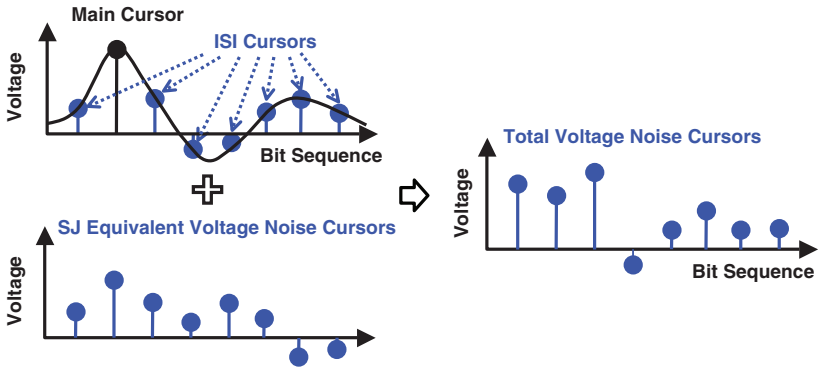


Figure 8.21 The Total Voltage Noise Sequence Generation

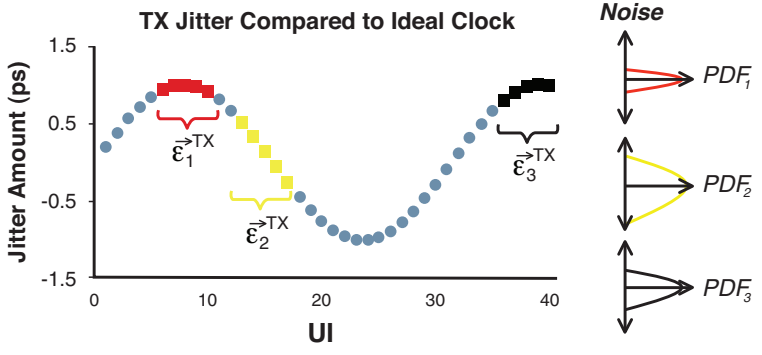


Figure 8.22 Different Jitter Sequences from Sinusoidal Jitter

8.5.2 Numerical Examples

Three examples are shown here to validate the method described in this section. The first example considers a high-speed parallel link, as shown in Figure 8.23 [21]. An 8Gb/s data is transmitting through a package-on-package (PoP) system. At the controller side, a die is bonded to a substrate package, while a two signal-layer package with wire-bond is used at DRAM side. All signals in the substrate and PCB are routed using microstrip lines. The total length, including package and PCB traces, is about 3 cm. The statistical eye with only ISI and without equalization is shown in Figure 8.24, with 54ps timing margin left. With transmitter equalization, (0.8,–0.2), the timing margin is increased to 98ps, as shown in Figure 8.25.

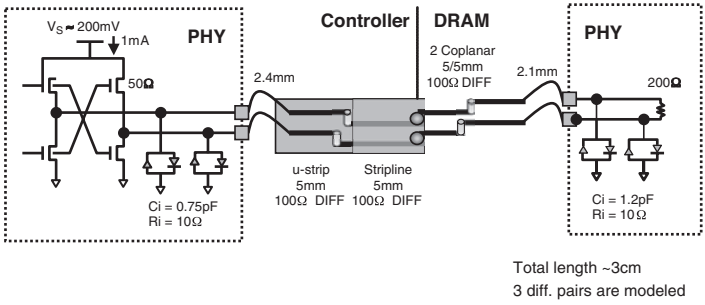


Figure 8.23 Memory I/O Interface in a Package-on-Package System

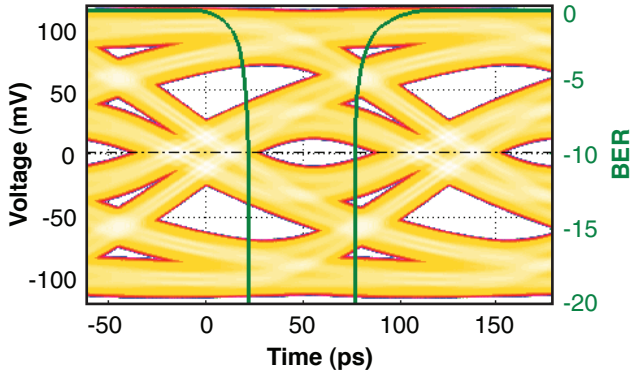


Figure 8.24 Unequalized Eye Diagram with ISI

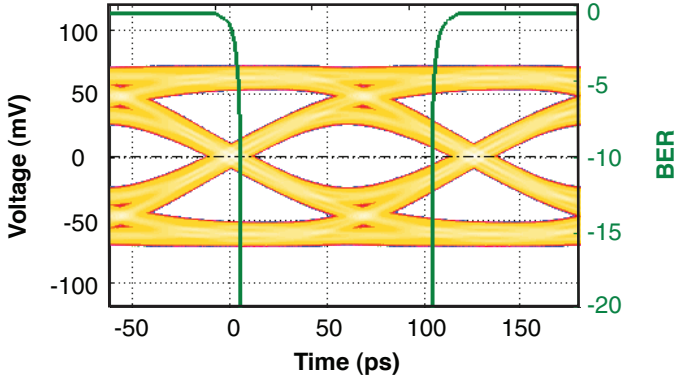


Figure 8.25 Equalized Eye Diagram with ISI

Figure 8.26 shows a statistical eye with a 100MHz, 20ps peak-to-peak transmitter sinusoid jitter, which models the correlation between the ISI and EVN of jitter. Observe the additional 20ps link margin loss, when comparing this figure to Figure 8.24. We also simulate a case that assumes ISI and EVN are independent; Figure 8.27 illustrates the resulting eye diagram. The assumption produced a more pessimistic estimation, resulting in a 54ps margin loss, as compared to the expected value of 20ps. The same comparison is performed with TX equalization; the results are shown in Figures 8.28 and 8.29. Again, the eye diagram with the ISI and EVN correlation model accurately predicts the extra 20ps margin loss, as expected. When TX jitter was modeled as independent jitter, the predicted margin loss was 30ps. The estimated loss, in this case, is not as pessimistic as in the previous, unequalized case. This example illustrates that the correlation between ISI and EVN is more important when ISI is large.

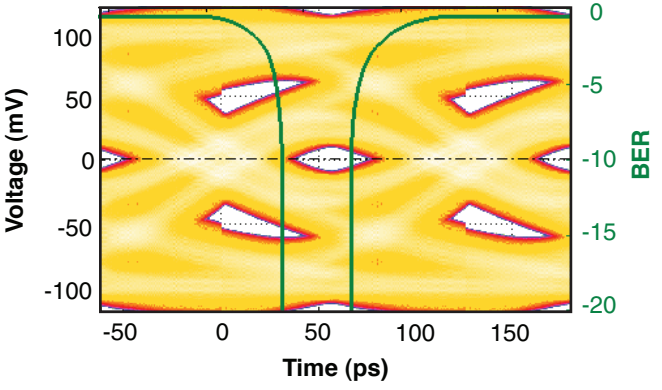


Figure 8.26 Eye with Correlation Between ISI and TX SJ EVN Modeled (No EQ)

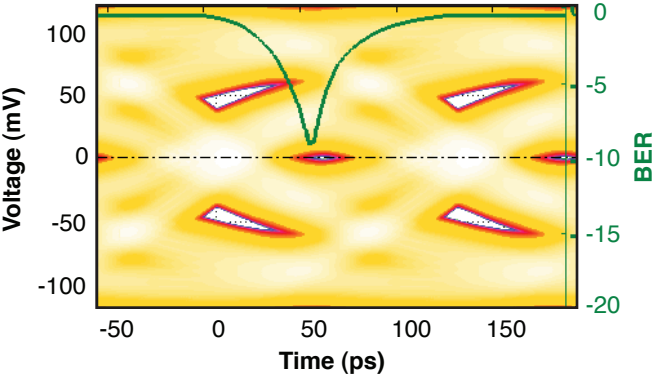


Figure 8.27 Eye with ISI and TX SJ EVN Modeled Independently (No EQ)

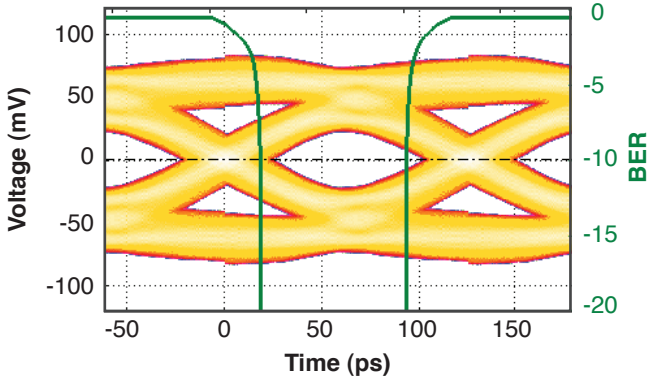


Figure 8.28 Eye with Correlation Between ISI and TX SJ EVN Modeled (EQ)

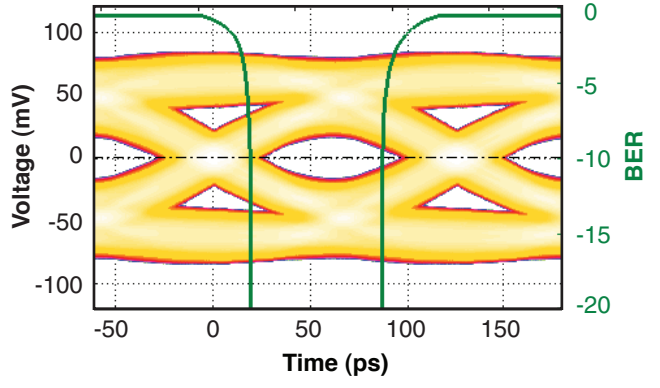


Figure 8.29 Eye with ISI and TX SJ EVN Modeled Independently (EQ)

Figure 8.30 illustrates a second example; this time a parallel link with a clock forwarding channel at a 3.2Gb/s data rate. The on-chip clock distribution on the receiver introduces a 2ns skew between the data and clock signals. Sinusoidal jitter of 30ps peak-to-peak, at 50 and 150MHz, is injected at the transmitter. Figure 8.31 shows that the simulated bathtub curves correlate excellently with the measurement data. Note that, due to low-frequency jitter tracking, the 50MHz jitter impact is smaller than the 150MHz jitter. Chapter 10 describes the method used to handle jitter tracking between the clock and data signals.

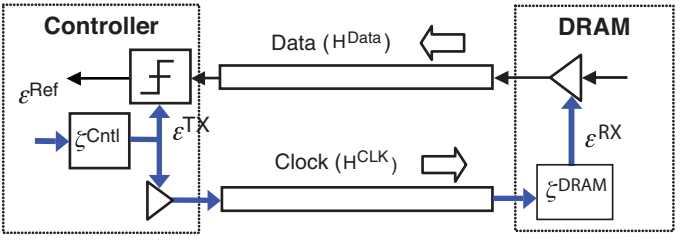


Figure 8.30 Clock-Forwarded Architecture with Common Jitter Source

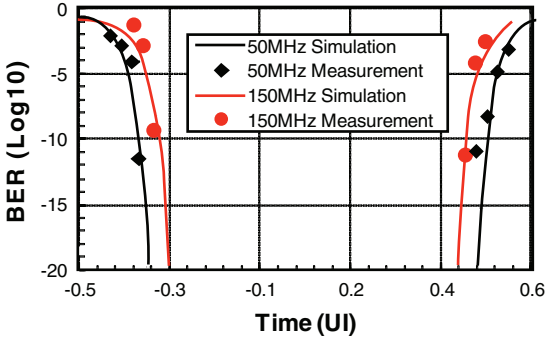


Figure 8.31 Correlation with Lab Measurement at Different SJ (Tracking)

8.6 Summary

This chapter reviews the statistical link simulation framework and discusses a general statistical formulation for computing link BER. It also reviews a fast statistical method to compute ISI PDFs, along with accurate device jitter models. The transmitter DCD is modeled, using short and long single-bit responses. The chapter covers how to model transmitter and receiver jitter, based on the equivalent voltage noise (EVN) concept. Finally, the chapter covers how to apply EVN to Gaussian noise, bounded random noise, and periodic jitter.

The statistical approach described in this chapter assumes a system that is linear, with uniform data patterns. A more typical system requires the transient simulation methods discussed in Chapter 9. Finally, an accurate jitter model must also account for clocking topology and jitter tracking. Chapter 10 covers this in detail.

References

1. V. Stojanovic and M. Horowitz, "Modeling and analysis of high-speed links," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2003, pp. 589–594.
2. D. Oh, F. Lambrecht, S. Chang, Q. Lin, J. Ren, J. Zerbe, C. Yuan, C. Madden, and V. Stojanovic, "Accurate method for analyzing high-speed I/O system performance," presented at the IEC DesignCon, Santa Clara, CA, 2007.
3. D. Oh, F. Lambrecht, S. Chang, Q. Lin, J. Ren, C. Yuan, J. Zerbe, and V. Stojanovic, "Accurate system voltage and timing margin simulation in high-speed I/O system designs," *IEEE Transaction on Advanced Packaging*, vol. 31, no. 4, pp. 722–730, Nov. 2008.
4. A. Sanders, M. Resso, and J. D'Ambrosia, "Channel compliance testing utilizing novel statistical eye methodology," presented at the IEC DesignCon, Santa Clara, CA, 2004.
5. C. W. Helstrom, "Calculating error probabilities for intersymbol and cochannel interference," *IEEE Transactions on Communications*, vol. COM-34, pp. 430–435, May, 1986.
6. B. K. Casper, M. Haycok, and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," in *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 2002, pp. 54–57.
7. B. Ahmad, "Performance specification of interconnect," presented at the IEC DesignCon, Santa Clara, CA, 2003.
8. D. Oh, "Multiple edge responses for fast and accurate system simulations," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 163–166.
9. J. Ren and D. Oh, "Multiple edge responses for fast and accurate system simulations," *IEEE Transaction on Advanced Packaging*, vol. 31, no. 4, pp. 741–748, Nov. 2008.
10. P. Patel, J. Cutcher, T. Donisi, M. Tsuk, and S.G. Pytel, "BladeServer 10Gb/s Ethernet backplane design with equalization using statistical channel analysis," presented at the IEC DesignCon, Santa Clara, CA, 2008.
11. R. I. Mellitz, M. Tsuk, T. Donisi, and S. Pytel, "Strategies for coping with nonlinear and time variant behavior for high speed serial buffer modeling," presented at the IEC DesignCon, Santa Clara, CA, 2008.
12. K. Xiao, B. Lee, and X. Ye, "A flexible and efficient bit error rate simulation method for high-speed differential link analysis using time-domain interpolation and superposition," in *Proceedings of International Symposium on Electromagnetic Compatibility*, Detroit, MI, Aug. 2008, pp. 1–6.

13. S. Chang and D. Oh, "Fast ISI characterization of passive channels using extreme value distribution," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2007, pp. 127–130.
14. Algorithm Modeling API (AMI) support in IBIS, IBIS ATM, 2007.
15. J. Caroselli and C. Liu, "An analytic system model for high speed interconnects and its application to the specification of signaling and equalization architectures for 10Gbps backplane communication," presented at the IEC DesignCon, Santa Clara, CA, 2006.
16. D. Oh, "Method for computing statistical system performance of high-speed links with the rising and falling edge responses," Idea Disclosure, Rambus Inc., Feb. 2005.
17. H. Hatamkhani, F. Lambrecht, V. Stojanovic and C. K. Ken Yang, "Power-centric design of high-speed I/Os," in *Proceedings of Design Automation Conference*, July 24–28, San Francisco, 2006, pp. 867–872.
18. J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*, 3rd ed., KAP, 2004, ch. 7, pp. 298–299.
19. G. Balamurugan, B. Casper, J. E. Jaussi, M. Mansuri, F. O'Mahony, and J. Kennedy, "Modeling and analysis of high-speed I/O links," *IEEE Transactions on Advanced Packaging*, vol. 32, no. 2, pp. 237–247, May 2009.
20. F. Rao, V. Borich, H. Abebe, and M. Yan, "Rigorous modeling of transmit jitter for accurate and efficient statistical eye simulation," presented at the IEC DesignCon, Santa Clara, CA, 2010.
21. S. Chang and D. Oh, "System-level modeling and simulation of periodic jitter in high-speed links," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2010, pp. 117–120.

This page intentionally left blank

CHAPTER 9

Fast Time-Domain Channel Simulation Techniques

Jihong Ren, Sam Chang, and Dan Oh

Statistical link simulation flow (described in Chapter 8, "Link BER Modeling and Simulation") provides an efficient way to predict link performance with device jitter. However, it has the following limitations:

- It assumes a white data pattern (or true random bit patterns), in order to calculate ISI PDF efficiently by convolution of pre- and post-ISI cursors (as explained in Section 8.3.1). This assumption is invalid when there is coding.
- Jitter modeling in the statistical domain usually assumes small jitter or a white jitter spectrum. This assumption is not valid in some cases, such as large sinusoidal jitter.
- It assumes that the system response is linear, making it less effective for systems with strongly non-linear drivers. Statistical approaches are based on the superposition of single-bit response (SBR) to calculate the ISI distribution efficiently. For SBR-based techniques to accurately simulate system response, the system must be linear time invariant (LTI), or able to be closely approximated as an LTI system. Differential signaling systems, with differential drivers, generally satisfy this criterion. In contrast, single-ended signaling systems, popular in memory interfaces, typically have different rising and falling edge responses. This variation in response is the result of either asymmetric I/O design or mismatches between the pull-up and pull-down drivers. SBR-based techniques are not applicable, or not sufficiently accurate, for these systems.

Fast time-domain simulation techniques were developed to overcome some of these limitations [1–5]. Even with these fast time-domain simulation techniques, the number of bits that can be simulated within a reasonable amount of time is still limited. Therefore, they are not suitable for simulating low probability events, such as random jitter. By mixing the statistical and transient simulation methods, one can take advantage of the strengths of both approaches [5]. For

example, we first rely on transient simulation to capture system non-linearity in histogram generation. Then, we separately simulate deterministic and random jitter, one in time-domain to capture jitter non-linearity, and the other in statistical domain, to capture random jitter tail statistics.

This chapter reviews such a hybrid framework. Section 9.1 provides a brief overview of the general time-domain simulation flow in conjunction with statistical link analysis. It also offers a short introduction to AMI, due to its popularity. Section 9.2 reviews different time-domain histogram generation methods, based on single-bit response (SBR), double-edge responses (DER), and multiple-edge responses (MER). Next, the extreme-value distribution (EVD) method is described, which extrapolates ISI PDFs based on time-domain histograms. Section 9.3 presents a numerical example for a complete link analysis, using a GDDR system to illustrate how the impact of coding on system performance can be simulated using time-domain flow. Section 9.4 provides a comparison of different jitter modeling methods and offers recommendations based on jitter characteristics. Section 9.5 reviews the peak distortion analysis for the SBR case, and ends with a detailed description of the dynamic programming algorithm, which searches for worst-case bit patterns for DER and MER in linear time.

9.1 Fast Time-Domain Simulation Flow Overview

Figure 9.1 summarizes a typical fast time-domain simulation flow. Such link analysis takes into account the link architecture, passive channel, and equalization algorithms, as well as various noise sources, in order to evaluate the performance of the system using BER (bit error rate) as the metric. The high-level steps are summarized as follows.

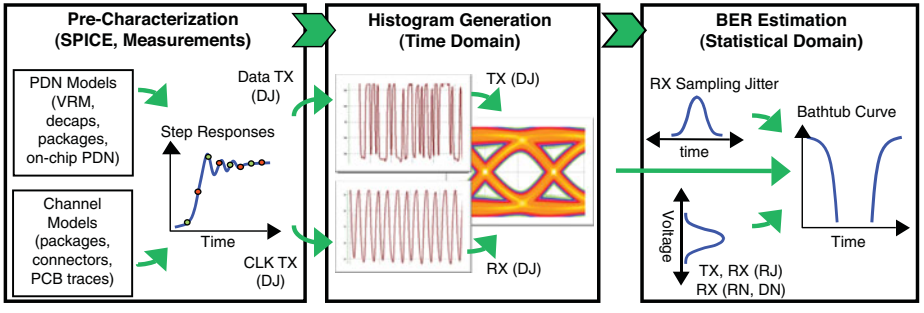


Figure 9.1 Hybrid Statistical and Time-Domain Simulation Flow with Signal and Power Integrity Models

First, HSPICE decks for the entire link are set up for pre-characterization, in order to obtain channel single-bit responses, or edge responses. The HSPICE decks include detailed channel models with packages, connectors, PCB traces, and so on. Moreover, the impact of a non-ideal power distribution network (PDN) can also be taken into account by including detailed PDN

models with VRM, decaps, packages, on-chip PDN, and so on in the HSPICE deck. By doing this, simultaneous switching noise (SSO) can be treated as another form of crosstalk. (Chapter 12, "SSN Modeling and Simulation," presents detailed information about SSO noise simulation.)

Second, with the extracted step or edge responses, the fast time-domain engine constructs the system response for millions of bits, and gathers the signal histogram. Jitter non-linearity is captured by including deterministic jitter in the time-domain histogram generation. Random jitter is addressed in the statistical domain by converting it into equivalent voltage noise (EVN), as seen by the receiver. The equivalent noise concept converts timing jitter to effective voltage noise (refer to Chapter 8). This allows for efficient simulation, because capturing the tail statistics of random jitter in the time-domain is difficult. In addition, a clock channel is simulated for source-synchronous systems with a forwarded clock, in order to generate receiver sampling clock ticks for the data channel. This accurately captures jitter tracking between the clock and data.

Finally, based on the time-domain histogram, ISI PDF is estimated with extreme value distribution (EVD) [6] [7]. The bathtub curves that are generated include ISI PDF, and equivalent voltage noise from RJ, as well as other noise sources (such as additional independent receiver jitter, sampling offsets, and receiver voltage noise).

9.1.1 Algorithmic Modeling Interface (AMI)

The *Algorithmic Modeling Interface* (AMI) is a modeling standard for SerDes transceivers, and is part of the IBIS 5.0 specification from August 2008. AMI allows models from different semiconductor vendors to work together, while simultaneously protecting the vendor's IP, because the models cannot be reverse-engineered. This allows semiconductor vendors to decide how much detail to expose to the user. As a result, AMI is widely supported by EDA vendors. This section briefly introduces the AMI simulation flow and methodology [1].

An AMI model consists of two parts: the analog model and the algorithmic model. The analog model includes the unequalized passive channel, Tx output impedance and parasitics, and the Rx input termination network and parasitics. The algorithmic models come from the SerDes IP vendors, and are supplied as a dynamically linked library. The algorithmic models include models for equalization, such as transmitter equalization, receiver linear equalization, and receiver decision feedback equalization, as well as clock recovery.

The first step in an AMI simulation is to characterize the analog model by its impulse response.

Next, choose one of two processing modes for the algorithmic models: in Mode 1, they process the impulse response at initialization; in Mode 2, they process the time-domain waveforms during time-domain simulation.

In Mode 1, the post-processed impulse response (for example, equalized impulse response) is used to generate either a statistical eye diagram, or a time-domain waveform. (Note that, in this mode, a linear time invariant system can only be modeled; for example, if only linear equalization is present and the equalization settings are set.)

In Mode 2, time-domain simulation is performed to model non-LTI behavior (such as adaptation and clock recovery), or when different encoding schemes are applied. Note that, in this

case, the analog model is still modeled as an impulse response. Bit patterns are convolved with the impulse response of the channel. The resulting waveform is first passed to the Tx AMI model, and then to the Rx AMI model for post-processing. Millions of bits can be simulated in minutes.

9.2 Fast System Simulation Techniques

This section introduces three fast time-domain system simulation techniques: single-bit response method, double-edge response method, and multiple-edge response method. For systems that can be well approximated as a linear time-invariant (LTI) system, these three techniques are equivalent. The latter two methods are developed to improve simulation accuracy for systems with non-linearities such as asymmetric I/O design or mismatches between the pull-up and pull-down drivers.

9.2.1 Single-Bit Response (SBR) Method

The single-bit response (SBR) method constructs the system response to an arbitrary input data pattern, using SBR from either simulation or measurement. Assuming the system is linear, the response to any data pattern is the linear sum of the shifted versions of SBR. Let $p(t)$ be the SBR of the channel, t_s be the data sampling phase, T be the bit time, and b_m be the mth transmitted symbol. The voltage seen by the receiver's data sampler at the mth data sample is determined by

$$y_m = \sum_k b_k p(t_s + (m - k)T). \tag{9.1}$$

Generally, a system's response to millions of bits is required to estimate system performance and predict BER. The memory of the passive channel often determines the length of the single bit response. The response time must be long enough to capture any major reflections.

Note that the single-bit response is generated by driving a single bit through a channel using a realistic driver model, which includes the non-linearity of the drivers, if present. The fundamental assumption of the SBR method is that a system can be accurately approximated as an LTI system. For systems with asymmetric rising and falling edge responses, the SBR method results in spurious glitches between consecutive ones. Figure 9.2 illustrates these glitches. Most single-ended signaling systems often have different rising and falling edge responses, due to either asymmetric I/O design or to mismatches between the pull-up and pull-down drivers. In contrast, most differential signaling systems do not suffer from this problem, because they have inherently symmetric rising and falling edge responses (as long as the mismatch between the two complementary drivers is small).

To handle the more general cases, with asymmetric rising and falling edges, the system response can be constructed in terms of edge transitions instead of bit responses. The balance of this section first reviews the double-edge response (DER) method, and then extends it to the multiple-edge response (MER) method to handle more general non-linearity.

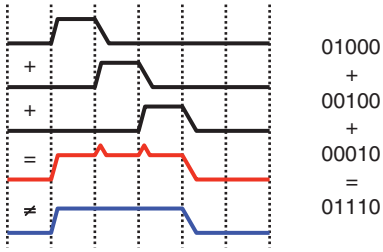


Figure 9.2 SBR Example with Artificial Glitches

9.2.2 Double-Edge Response (DER) Method

The DER method decomposes the input data pattern, in terms of rising and falling edge transitions. The system response can be calculated by superimposing the shifted versions of the rising and falling edge responses:

$$y_m = \sum_k (b_k - b_{k-1}) s_k(t_s + (m - k)T) + y_{-\infty} \tag{9.2}$$

where

$$\begin{aligned} s_i(t) &= r(t) - V_{low} \quad \text{if } (b_i > b_{i-1}) \\ &= V_{high} - f(t) \quad \text{otherwise.} \end{aligned} \tag{9.3}$$

$r(t)$ and $f(t)$ are the rising and falling edge responses, respectively. V_{high} and V_{low} are the steady state DC levels, in response to a constant stream of ones and zeros, respectively. $y_{-\infty}$ is the initial DC state (either V_{high} or V_{low}). Figure 9.3 illustrates the construction of the single-bit response, using rising and falling edge responses. Similar to the SBR method, the edge responses with a realistic driver model is used to drive the passive channel. Therefore, the non-linearity of the driver is present in the edge responses.

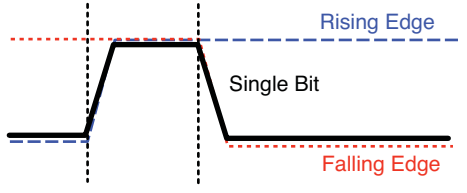


Figure 9.3 Construction of Single-Bit Using Double-Edge Responses

9.2.3 Multiple-Edge Response (MER) Method

Both the SBR and DER methods have limited capability when modeling non-linear driver effects. The DER approach has better simulation accuracy than the SBR approach, when the non-

linear driver effect is limited by the asymmetric rising and falling edge responses within a bit time. However, it cannot completely capture the non-linear effect for the rest of the responses, because the driver switching activity is not considered beyond the current bit time.

To illustrate this point, consider an artificial non-linear driver with three different driver impedances, at three different voltage levels: x, y, and z, for the low, transition, and high levels, respectively (see Figure 9.4). Tag the rising and falling edge responses with driver impedance settings: $r_{x,y,z,y,x}(t)$ and $f_{x,y,z,y,x}(t)$. The single bit response is denoted as $p_{x,y,z,y,x}(t)$. Because of the impedance changes, the single-bit response can no longer be accurately constructed by shifting and adding the rising and falling edge responses with normal impedance conditions (x,y,z,z,z) and (z,z,z,y,x). Instead, a hypothetical falling edge response (x,y,z,y,x) can be derived from the single-bit response and the normal rising edge response to capture the non-linearity of the system, as follows:

$$\begin{aligned} p_{x,y,z,y,x}(t) &= r_{x,y,z,z,z}(t) + f_{x,y,z,y,x}(t - T) - V_{high} \\ &\neq r_{x,y,z,z,z}(t) + f_{z,z,z,y,x}(t - T) - V_{high}. \end{aligned} \tag{9.4}$$

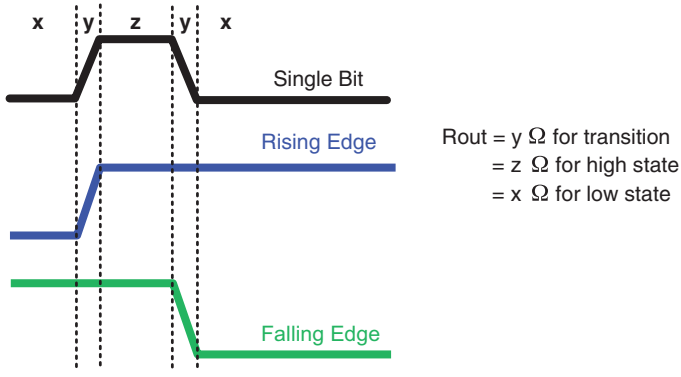


Figure 9.4 Various Responses of an Artificial Non-Linear Driver with Three Driver Impedances

The hypothetical edge response is used later, to construct system responses. This is the basic idea behind the MER method. The rest of this section presents the MER formulation in detail.

The MER method constructs multiple rising and falling edges, depending on the previous bit patterns, as follows:

$$\begin{aligned} f^{11}(t) &= y^{110}(t), \\ f^{01}(t) &= y^{010}(t) - y^{001}(t + T) + V_{high} \end{aligned} \tag{9.5}$$

where $f^{b_{-2}1}(t)$ is the falling edge response to a falling edge at time 0 with the previous bit b_{-2}, and $y^{b_{-m}\ldots b_{-1}b_0}$ is the response to the bit pattern $b_{-m}\ldots b_{-1}b_0$. Similarly, for the rising edge, we have

$$r^{00}(t) = y^{001}(t),$$
$$r^{10}(t) = y^{101}(t) - y^{110}(t + T) + V_{low}. \tag{9.6}$$

Figure 9.5 shows the corresponding MER responses graphically. With this construction, the previous single-bit response (in Figure 9.4) can be readily recovered from $f^{01}(t)$ and $r^{00}(t)$ as follows:

$$p_{x,y,z,y,x}(t) = y^{010}(t) = r^{00}(t + T) + f^{01}(t) - V_{high}. \tag{9.7}$$

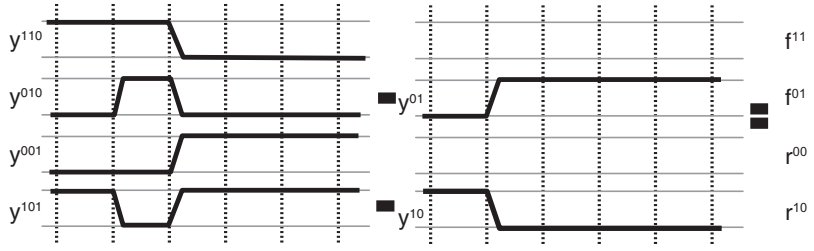


Figure 9.5 Second-Order MER Examples

This formulation of multiple-edge responses (MER) is referred to as the second-order MER, because it considers the impact of the previous two bits on the current bit. Note that the first-order MER is equivalent to DER. The preceding formulation can be generalized to a higher order to achieve greater accuracy, as follows:

$$r^{b_{-m},\ldots,b_{-2}0}(t) = y^{b_{-m},\ldots,b_{-2}01}(t) - y^{b_{-m}b_{-m},\ldots,b_{-k+1}b_{-k}}(t + kT) + V_{low}$$
$$f^{b_{-m},\ldots,b_{-2}1}(t) = y^{b_{-m},\ldots,b_{-2}10}(t) - y^{b_{-m}b_{-m},\ldots,b_{-k+1}b_{-k}}(t + kT) + V_{high} \tag{9.8}$$

where k is the right-most bit index of the bit pattern $b_{-m},\ldots,b_{-2}b_{-1}$, such that $b_{-k+1} \neq b_{-k}$ and $b_{-m}b_{-m},\ldots,b_{-k+1}b_{-k}$ is the right shifted version of $b_{-m}b_{-m},\ldots,b_{-1}b_0$, where the left-most bits exposed by the right shift are filled with b_{-m}.

Based on the constructed multiple-edge responses, the system response can be estimated in a manner similar to (9.2), except now the edge responses are selected based on previous bit patterns. For example, with a third-order MER, when a rising edge is preceded by 00, 01, 10, and 11 data patterns, $r^{000}(t)$, $r^{010}(t)$, $r^{100}(t)$, and $r^{110}(t)$ responses are selected, respectively.

The mth order MER, which requires a total of 2^m rising and falling edge responses, can be prepared using 2^m data patterns. Although higher-order MERs take longer to prepare input waveforms, the impact on the actual simulation time is negligible. For many applications, the second-order MER, which is based on rising and falling edges and single-bit up and down responses, provides reasonable accuracy. Most of this chapter's numerical examples use this simple version.

Section 9.2.4 presents a simple strategy to predetermine the MER order required to achieve good simulation accuracy for a given system.

9.2.4 Numerical Examples

This section presents numerical examples with which to compare the SBR, DER, and MER methods against HSPICE. These numerical examples show that different signaling systems have different requirements for MER levels, in order to achieve good simulation accuracy. This section presents a simple strategy to predetermine the MER level required for a given link.

For the first example, consider a simple differential signaling system using a current-mode driver (as shown in Figure 9.6). Figure 9.7 shows the single-bit response, and the rising and falling edge responses, from the HSPICE simulation. The system response of the 0111001010 data pattern is approximated using SBR, DER, and a second-order MER. Figure 9.8 compares the simulation results and illustrates an excellent match for all cases. Figure 9.9 shows the error plot. In this example, all three approaches result in similar errors.

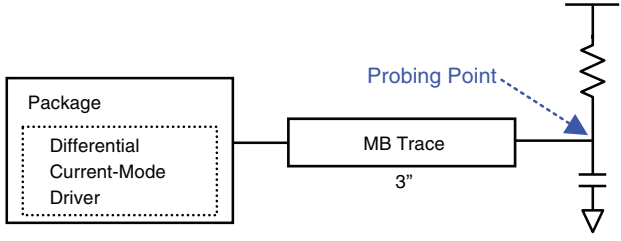


Figure 9.6 Sample I/O Channel with a Differential Current Driver

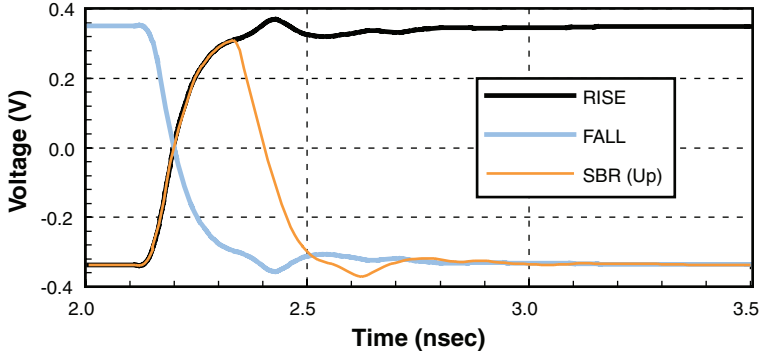


Figure 9.7 Single-Bit Rising and Falling Edge Responses

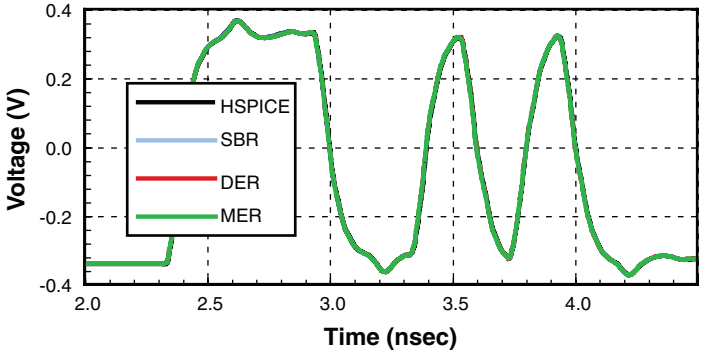


Figure 9.8 Responses to Data Pattern from Differential Signaling System using SBR, DER, Second-Order MER, and HSPICE

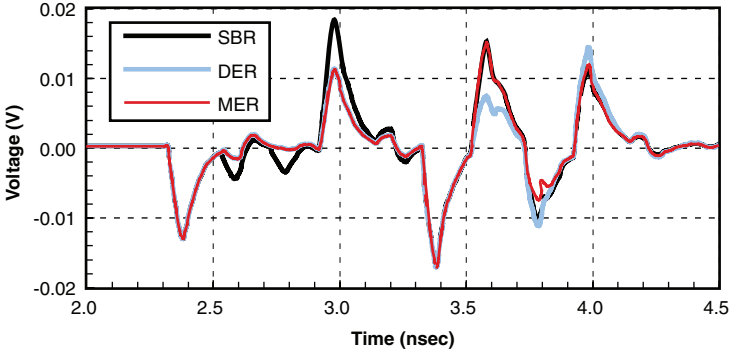


Figure 9.9 Error Compared to HSPICE for Differential Signaling System (Error Normalized with Respect to Received DC Swing)

For the second example, consider an RDRAM memory channel [8]. The overall system consists of an RDRAM controller with two 16-device memory modules (as shown in Figure 9.10). A WRITE transaction is simulated from the controller to a DRAM. Figure 9.11 shows the waveforms used by the second-order MER for this single-ended signaling system. Figure 9.12 shows the calculated waveforms for the same 0111001010 data pattern. Figure 9.13 shows the error plot. As expected, the response calculated from SBR shows spurious ripples during a long period of 1s.

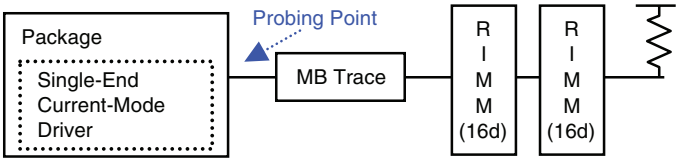


Figure 9.10 Sample Single-Ended Signaling System: RDRAM Driver with Two 16-Device Memory Modules

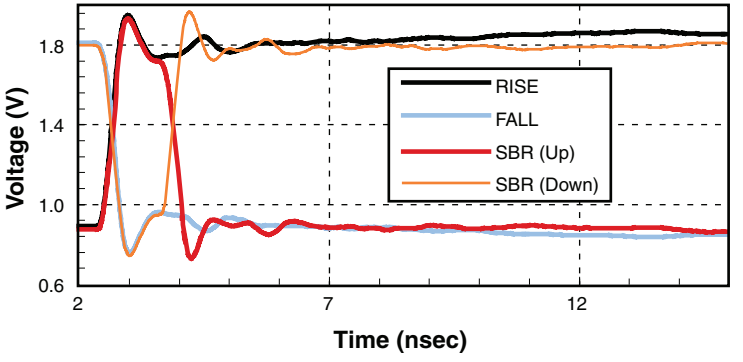


Figure 9.11 Single-Bit Rising and Falling Edge Responses of Single-Ended Signaling System in Figure 9.10

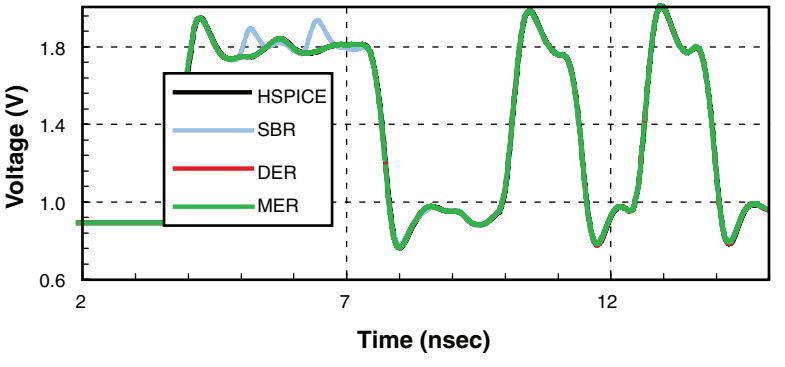


Figure 9.12 Responses to Data Pattern from Single-Ended Signaling System Using SBR, DER, Second-Order MER, and HSPICE

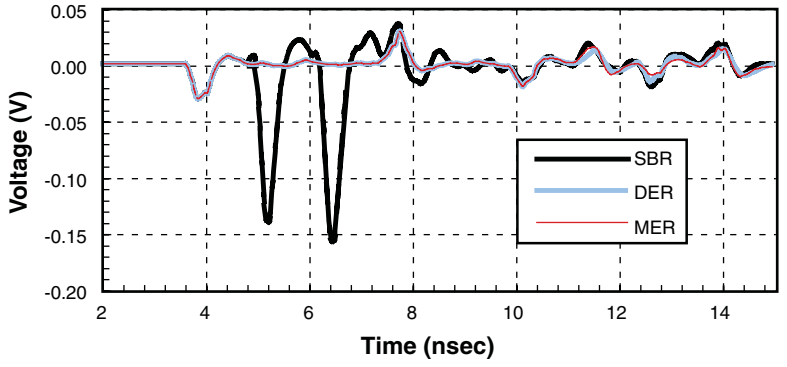


Figure 9.13 Error Compared to HSPICE Case for Single-Ended Signaling System (Error Normalized with Respect to Received DC Swing)

Figure 9.14 compares the eye diagram from the HPSICE simulation to that from the second-order MER simulation for a 1024-bit pseudo-random data pattern. The eye diagrams further confirm that MER offers reasonable accuracy.

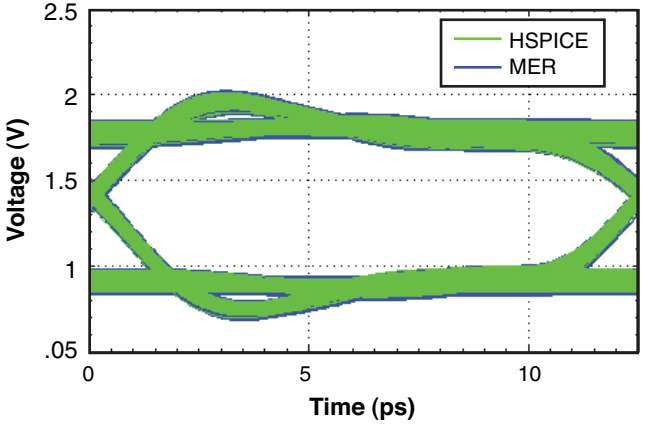


Figure 9.14 Eye Diagrams for a 1024-Bit PRBS Pattern Simulated with Second-Order MER and HSPICE

So far, the examples have only used the second-order MER, which considers single-bit responses together with rising and falling edge responses. It has proved to be accurate, when compared to the HPSICE simulation.

However, to demonstrate the need for a higher-order MER, the next example is slightly more challenging. This example is based on another popular single-end signaling system with

Pseudo Open Drain (POD I/O). POD I/O is widely used for high-speed graphic memory applications. Figure 9.15 shows the test system. Note from the figure that a significant termination mismatch has been introduced, in order to make the simulation more challenging. This termination mismatch shows up in the edge responses (see Figure 9.16). In this system, DER introduces a large error, and fifth-order MER is required to achieve the necessary accuracy (see Figures 9.17 and 9.18).

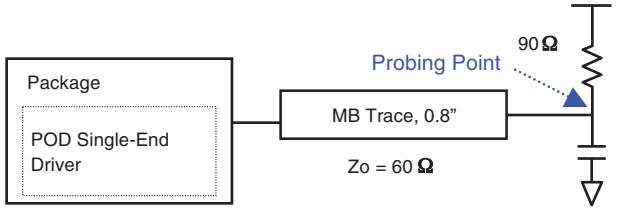


Figure 9.15 Sample Single-Ended Signaling System Based on POD Signaling

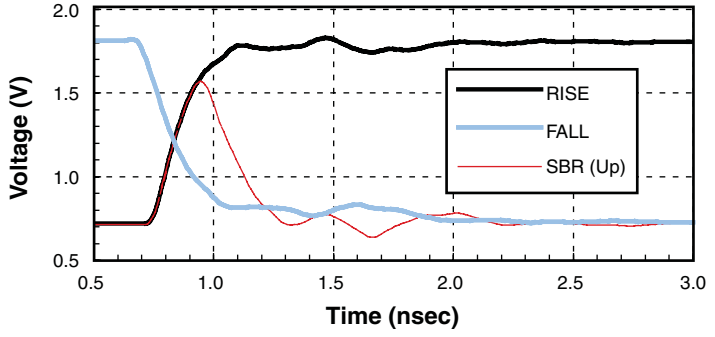


Figure 9.16 Single-Bit Rising and Falling Edge Responses of Single-Ended Signaling System in Figure 9.15

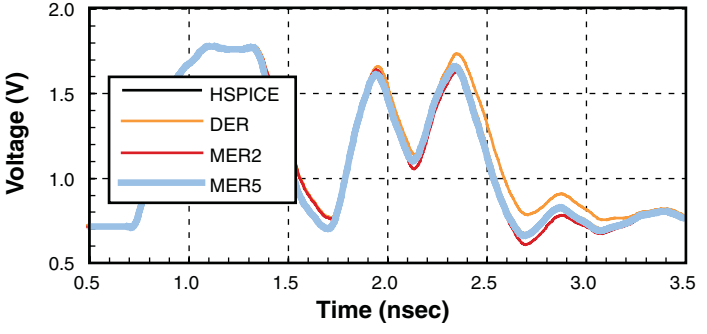


Figure 9.17 Responses to Data Pattern of Single-Ended Signaling System Using DER, Second-Order MER, Fifth-Order MER, and HSPICE

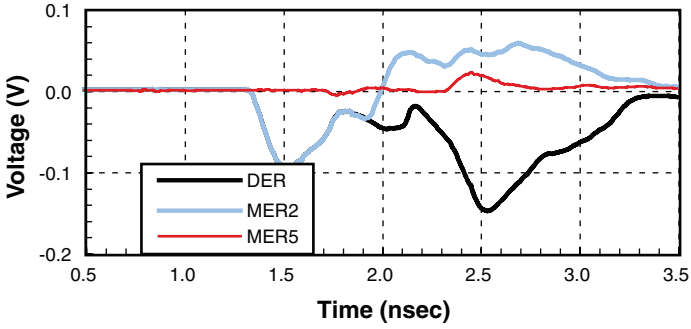


Figure 9.18 Error Compared to HSPICE for Single-Ended Signaling System in Figure 9.15 Using DER, Second-Order MER, and Fifth-Order MER (Error Normalized with Respect to Received DC Swing)

The following steps can be used to predetermine the required MER level heuristically, before running the MER simulation:

1. Run spice simulation with the clock pattern preceding a rising edge; for example, 0101010111111.
2. Run spice simulation with the clock pattern of the same length without the following rising edge; for example, 0101010000000.
3. Subtract the two responses to obtain the edge response corresponding to the clock pattern as a leading bit pattern: 0101010.
4. Compare the response with the 01 edge response. The required MER level should cover the time the two edge responses differ significantly.
5. Repeat steps 1–4 for the falling edge response.

As part of this demonstration, the motherboard trace length of the POD example is swept (0.4″, 0.8″, and 1.8″). Figure 9.19 shows the rising edge responses for no leading bit pattern and the clock pattern 010101010101010101, as well as the difference between the two edge responses for the motherboard trace length of 0.8″. The two rising edge responses differ significantly for roughly 1ns. At 500Mb/s, this means a MER order of 5 is required to capture the nonlinearity. The edge response differences for 0.4″ and 1.8″ last roughly 5 bits and 7 bits, respectively, at 500Mb/s. Figure 9.20 shows the maximum normalized error for the three cases, with different methods and MER levels. We calculate the error for a 200-bit long pattern, consisting of a random bit pattern of 150 bits, followed by a 50-bit worst-case pattern for each method. The SBR and DER methods significantly deviate from the HSPICE simulation results. Higher-order MER methods achieve good accuracy (error less than 5%) with levels 5, 5, and 7, respectively. This is consistent with the predetermined MER simulation levels.

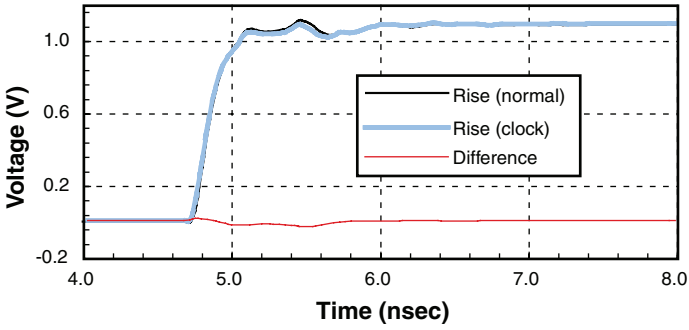


Figure 9.19 Rising-Edge Response Compared with Rising-Edge Response Lead by a Clock Pattern for POD Motherboard Trace Length of 0.8″

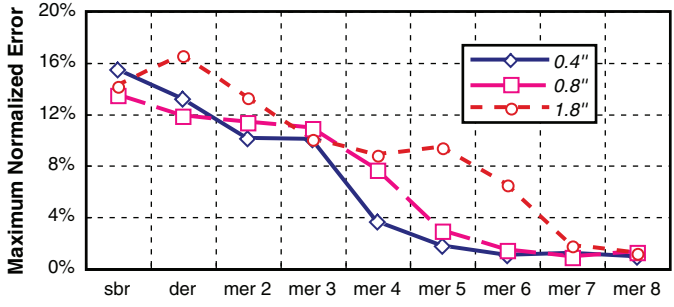


Figure 9.20 Maximum Normalized Error for POD Systems with Motherboard Trace Lengths of 0.4″, 0.8″, and 1.8″

Similar to the SBR method, the MER method is solely based on simple operations, like shift and addition. Figure 9.21 illustrates that the running time is linear for the length of the data pattern and the over-sampling rate. A laptop computer, with a 3GHz Intel Xeon™ CPU and 1GByte of memory, can run a simulation of 100 million bits in less than 4 minutes.

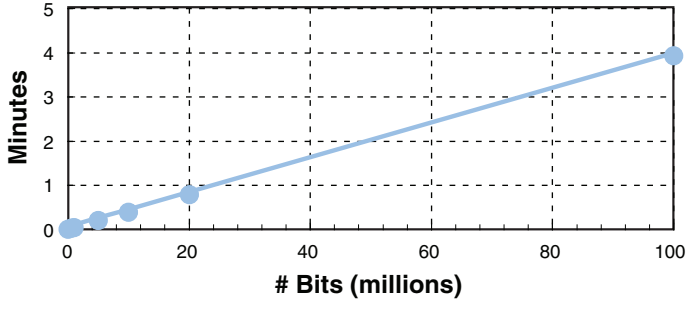


Figure 9.21 Simulation Time as a Function of Data-Pattern Length (Simulation Is Performed on Linux Platform with Intel Xeon 3GHz Processor)

9.2.5 Extreme Value Distribution

Accurate BER simulation requires accurate ISI noise distribution, which, in turn, requires simulating an extremely large number of data bits. For example, for a typical backplane channel, the SBR could be more than 100 bits due to reflections. To capture the exact ISI distribution, we need to simulate 2^{100} bits. Simulating a limited number of bits results in truncation of the ISI distribution, as shown in Figure 9.22. It compares the exact PDF with the histogram generated by a time-domain simulation of 10^5 bits. The exact PDF ends around 10^{-10}, whereas the time-domain histogram is truncated at around 10^{-5}. Figure 9.23 demonstrates the impact of truncation on the system performance estimation. As Figure 9.23 indicates, the time-domain approach, using a truncated distribution, significantly underestimates the deterministic noise at the lower BER.

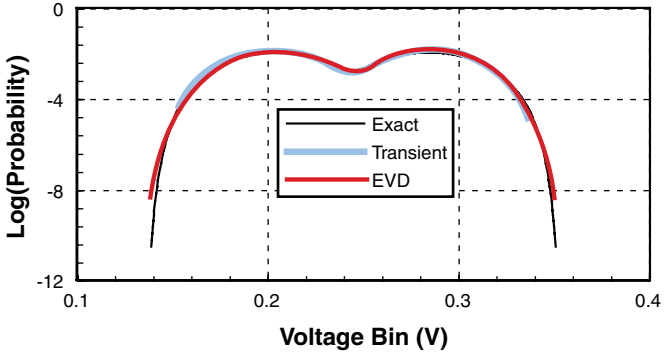


Figure 9.22 Comparison of Exact, Transient, and EVD-Extrapolated PDFs

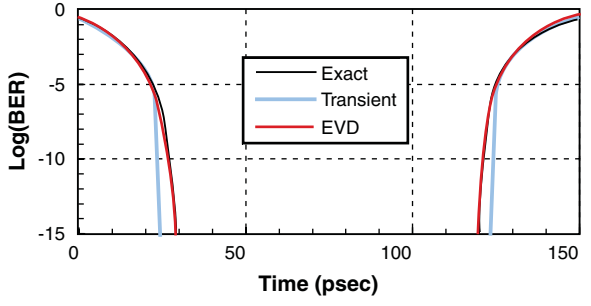


Figure 9.23 Comparison of Timing Bathtub Curves From Exact, Transient, and EVD-Extrapolated PDFs

Extrapolation generally provides a better estimate of the performance at lower BER, but it assumes Gaussian distribution of random noises [9]. Fundamentally, this approximation assumes that unbounded Gaussian noises generate the tail of the distribution, which makes extrapolation

inadequate for bounded distributions, such as an ISI distribution. The alternative method uses extreme value distribution (EVD) to estimate the tails of the cumulative distribution function (CDF). In the field of computational biology, CDFs are used to evaluate the likelihood of a protein structure prediction being correct [10] [11]. EVD theory states that the asymptotic distributions of maximum (or minimum) values in a very large collection of samples from one random variable belong to three general distribution families, regardless of the random variable type [6]. These families are Gumbel, Frechet, and Weibull. Their CDFs are defined by Equations (9.9a), (9.9b), and (9.9c), respectively.

$$F(x) = e^{-e^{-(x-\mu)/\delta}} \text{ for } x \in \Re \tag{9.9a}$$

$$F(x) = \begin{cases} 0 & x \le \mu \\ e^{-((x-\mu)/\delta)^{-a}} & x > \mu \end{cases} \tag{9.9b}$$

$$F(x) = \begin{cases} e^{-((x-\mu)/\delta)^{a}} & x \le \mu \\ 1 & x > \mu \end{cases}. \tag{9.9c}$$

The second and third types of CDF are linked to the first type by shifting and taking the natural log operation [6]. Depending on the application, one may choose to use any of the three types. For example, for lifetime-distribution modeling in biology, Weibull (9.9c) is the best choice, because time is always positive, and Weibull guarantees positive $F(x)$. For modeling ISI distribution, Gumbel (9.9a) is selected as a starting point to curve-fit the tail portion of the ISI distribution.

To obtain better fitting, the original Gumbel distribution is slightly modified using a second-order polynomial:

$$P(X < x) = e^{-e^{(ax^2+bx+c)}}. \tag{9.10}$$

The results of ln(-ln($F(x)$)) are used to fit the second-order polynomial series:

$$ax^2 + bx + c = 1\text{n}(-1\text{n}(P(X < x))) \tag{9.11}$$

where the a, b, and c parameters are estimated using the least square approximation. Next, we estimate the tail of the exact CDF, and extrapolate it to a desired bound from the truncated time-domain histogram. Finally, the PDF is calculated by differentiating the approximate CDF tail.

Figure 9.22 shows a good match between the extrapolated PDF and the exact PDF. EVD extrapolated the PDF bounds accurately and extended the bound from 10^{-5} to $10^{-8.5}$. This bound is sufficient to model BER values below 10^{-12}, when combined with random jitter (see Figure 9.23).

9.3 Simultaneous Switching Noise Example

Simultaneous Switching Noise (SSN) is one of the major performance bottlenecks in single-ended signaling systems. (Chapter 12 covers SSN in detail.) This section provides a brief overview of how the SSN impact can be modeled in link margin analysis based on MER [5]. The GDDR system for high-end graphics applications is used as an example. The GDDR system uses pseudo open drain circuits, with push/pull drivers of 60Ω and 40Ω, respectively. The channel model consists of ten coupled transmission lines for DQ and RQ. All DQ and RQ lines are point-to-point. By only exciting the driver for the victim channel, and observing the signal at the victim channel output (just before the slicer), one can extract the edge responses for the victim channel. Similarly, by exciting the aggressor channel and keeping the victim channel quiet, one can extract the edge responses for the crosstalk. With this setup, the crosstalk from the nearest neighbors is only considered, and the crosstalk generated from the second-nearest neighbors is ignored. For worst-case SSN generation, we have all other lines switching at the same time, and capture the noise generated on the victim line. Note that, with the PDN model incorporated, the crosstalk generated includes the effects of SSN. MER naturally simulates SSN without any increase in computational complexity. Figure 9.24 shows the correlation between MER and HSPICE for SSN-only, crosstalk-only, and final victim-channel response.

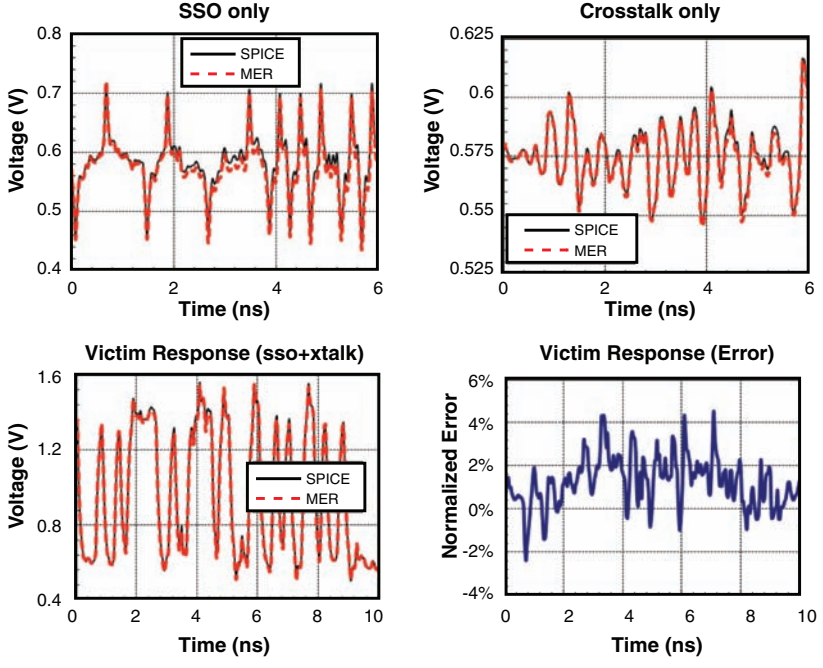


Figure 9.24 MER- and HSPICE-Simulated Waveforms

Figure 9.25(a) shows the eye diagram for the victim channel, operating at 5Gb/s with a random data pattern. The SSN from the other 39 DQs are included. Because MER is time-domain based, a hybrid statistical and time-domain flow is used to study the effectiveness of different coding schemes on the final system margin. Figure 9.25(b) shows the system margin improvement using data bus inversion (DBI-DC) coding, which we use in GDDR4 systems to reduce the impact of SSN (see Chapter 13, "SSN Reduction Codes and Signaling"). Compared with random data, DBI-DC improves the timing margin from 13ps to 25ps at 10^{-20} BER. Figure 9.26 compares the ISI distribution and SSN distribution, on the victim channel, for random data and DBI-DC coded data. The figure indicates that DBI-DC shifts the ISI distribution, and makes the upper and lower eye asymmetric. Statistical simulators cannot capture this effect.

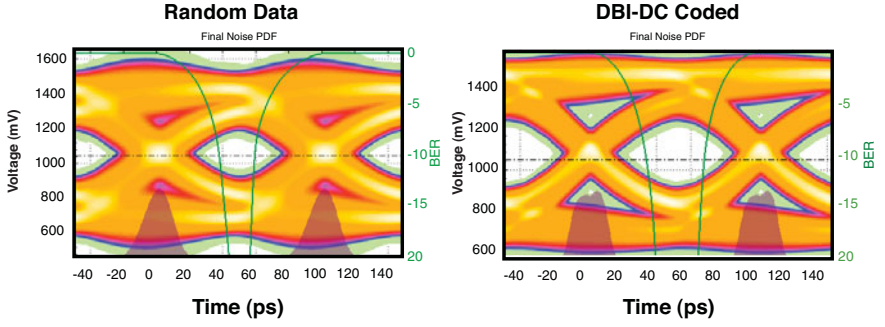


Figure 9.25 Simulated Statistical Eye Diagrams and Bathtubs for a 32-bit GDDR System with No Coding and DBI-DC Coding

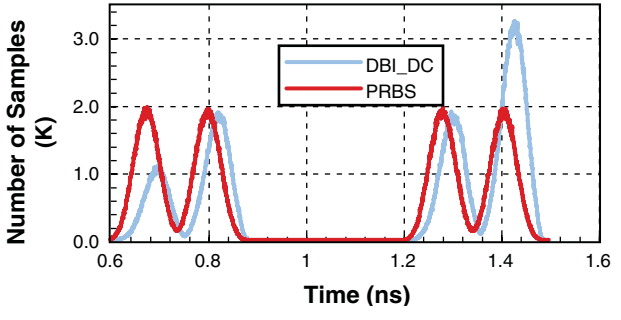


Figure 9.26 ISI Histograms at Data Sampling Location Without SSO Noise

9.4 Comparison of Jitter Modeling Methods

One of the major advantages of the hybrid approach is the freedom to choose different jitter modeling methods, based on jitter characteristics. Many methods have been proposed, and one can

choose from a variety of statistical and time-domain based methods. Statistical jitter modeling methods include the equivalent voltage noise (EVN) method [12], the Rx sampling method [13], and the segment-based method [14]. This section covers using the hybrid simulation framework to demonstrate the pros and cons of different jitter modeling methods, using sinusoidal jitter and Gaussian jitter (RJ). (Because the segment-based method can only model transmitter jitter with a white spectrum, it is excluded from this comparison.) The link used for this demonstration is a short (3") FR4 channel, with Tx/Rx parasitic capacitance of 1.5pF, running at 8Gb/s. At Nyquist frequency, the channel has a –10dB loss. Timing margin is calculated at 10^{-15} BER.

First, consider white Gaussian transmitter jitter. As shown in Figure 8.4, the margin loss predicted by the time-domain simulation increases as more bits are simulated. It is impractical to capture the impact of RJ at low BER (such as 10^{-15}) using time-domain simulation as it requires too many bits. The Rx sampling distribution method is the most optimistic, because it ignores the jitter amplification of the passive channel. The EVN method efficiently captures both jitter amplification and the tail statistics of RJ, and so predicts the worst margin loss.

Unlike Gaussian transmitter jitter, high-frequency deterministic jitter (such as high frequency SJ) is efficiently simulated in the time-domain with a limited number of bits (for example, 10^7 bits). Figure 9.27 compares different methods of simulating sinusoidal jitter (20ps peak to peak) across different jitter frequencies. The jitter amplification factor is roughly 1 for the Rx sampling distribution method, as it ignores the jitter amplification of the channel. For low-frequency jitter, the time-domain method predicts roughly the same jitter amplification factor of 1, because low-frequency jitter is not amplified by the channel. At high frequency, the jitter amplification of the passive channel takes effect. The EVN method, which independently considers ISI and equivalent voltage noise, drastically overestimates the impact of SJ. After the correlation between ISI and equivalent voltage noise is considered, the predicted jitter amplification is closer to that predicted by the time-domain method.

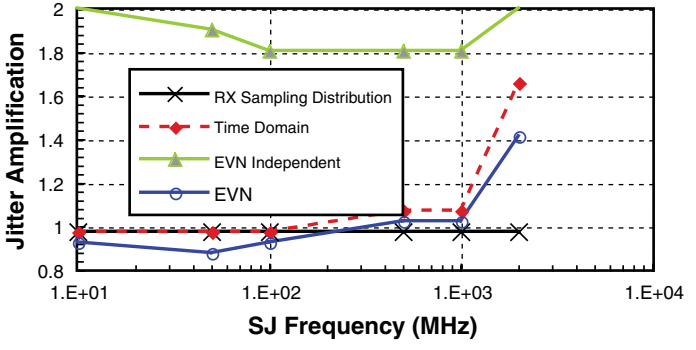


Figure 9.27 Comparison of Link Margin Loss Due to Sinusoidal Transmitter Jitter

Choosing the appropriate modeling methods, based on the location of the jitter source (for example, transmitter, receiver, or common between the two), as well as the jitter characteristics (such as jitter spectrum and distribution) is important. Table 9.1 shows the recommended jitter modeling methods, based on jitter characteristics. For example, if the jitter is only present on the receiver, or it is low-frequency deterministic jitter, the Rx sampling distribution method is the most efficient and accurate, because it avoids time-domain simulation and EVN's first-order approximation. On the other hand, if it is high-frequency deterministic jitter from the transmitter, the time-domain jitter model is most accurate, in terms of capturing jitter amplification. For transmitter RJ, the equivalent voltage noise method is recommended due to its efficiency.

Table 9.1 Recommended Jitter Modeling Methods

Location	Jitter Noise Type	Recommended Method
TX and Common	LF DJ	Rx Sampling Distribution
	HF DJ	Time Domain
	RJ	Equivalent Voltage Noise
RX	DJ/RJ	Rx Sampling Distribution

9.5 Peak Distortion Analysis

Peak distortion analysis is commonly used to extract a worst-case eye diagram [15-19]. This section first describes peak distortion analysis techniques based on single-bit response and multiple-edge response methods. Typical peak distortion analysis is done at the data sampling phase. Therefore it only gives worst-case voltage margin. This section extends peak distortion analysis to multiple phases to generate worst-case eye that has not only the worst-case voltage margin but also the worst-case timing margin. At the end, numerical examples are presented to compare the worst-case eye margin extracted using different methods.

9.5.1 Single-Bit Response (SBR) Method

Given an SBR, the largest undershoot can be found by picking the data sequence that results in the most negative total ISIs, as follows. Beginning with the symbol-spaced single-bit response $p_{\Delta t}$ at phase Δt, let $p^M_{\Delta t}$ be the main cursor, and $\vec{p}^{\,ISI}_{\Delta t}$ be a vector that contains all the ISI terms of the single-bit response at phase Δt. The lowest possible undershoot is given by $p^M_{\Delta t} - \left|\left|\vec{p}^{\,ISI}_{\Delta t}\right|\right|_1$, where $\left|\left|\;\right|\right|_1$ denotes the l_1-norm, the sum of the absolute values of the elements.

9.5.2 Double-Edge Response (DER) and Multiple-Edge Response (MER) Methods

As shown previously, in non-linear systems, the ISI contributed from one bit not only depends on the sign of the bit itself, but also on the signs of its predecessors. For example, the ISI from a single 1 is different from the ISI from a 1 bit preceded by another 1 bit. Therefore, the peak distortion method described for SBR does not apply to DER and MER. There is a greedy algorithm designed to find the worst-case data pattern for DER [18]. The algorithm starts from the sample bit of interest and then works backwards, bit by bit, to find the data pattern that may result in the worst-case overshoot or undershoot at the sample bit. Unfortunately, unlike the SBR case, the greedy algorithm does not always result in the worst-case pattern. For example, if a rising edge results in a disturbance of –0.1 at the sample bit, and is therefore chosen, it eliminates the possibility of a rising edge one bit earlier, which might result in a disturbance of –0.2 on the sample bit. The algorithm described by Drabkin et al. [19] is similar. It avoids the problem written about by Lambrecht, Huang, and Fox [19] by always looking only at the local minimums and maximums of the edge responses. However, this algorithm only works for symmetric or nearly symmetric rising and falling edge responses, because it has to ensure the alignment of the edge response's minimums and maximums. For a case like this, we can always use SBR instead of DER.

The remainder of this subsection introduces peak distortion analysis for DER and MER, using dynamic programming [20], which is guaranteed to generate the worst-case data pattern for general applications. Similar to peak distortion analysis based on SBR, this method not only returns the worst-case eye height, but also the worst-case input sequences that produce the worst-case eye.

The basic idea is as follows. Look at the worst-case undershoot first. For the mth order MER, let $A_{b_{-m},\ldots,b_{-1}}(j)$ be the worst-case accumulated signal level at bit M, caused by inputs from 1 to j, where the input ends with pattern $b_m,\ldots,b_{-1}$ at the jth bit. For the next bit, $A_{b_{-m},\ldots,b_{-1}}(j+1)$ can only come from either $A_{1,b_{-m},\ldots,b_{-2}}(j)$ or $A_{0,b_{-m},\ldots,b_{-2}}(j)$ by appending b_{-1}. The resulting equation is

$$A_{b_{-m},\ldots,b_{-1}}(j+1) = \min\begin{cases} A_{1,b_{-m},\ldots,b_{-2}}(j) + (b_{-1} - b_{-2})S_{\Delta t}^{1,b_{-m},\ldots,b_{-1}}(N-j) \\ A_{0,b_{-m},\ldots,b_{-2}}(j) + (b_{-1} - b_{-2})S_{\Delta t}^{0,b_{-m},\ldots,b_{-1}}(N-j) \end{cases} \tag{9.12a}$$

where

$$\begin{aligned} S_{\Delta t}^{1,b_{-m},\ldots,b_{-1}} &= r_{\Delta t}^{1,b_{-m},\ldots,b_{-2}} - V_{low} \quad \text{if } b_{-1} > b_{-2} \\ &= V_{high} - f_{\Delta t}^{1,b_{-m},\ldots,b_{-2}} \quad \text{if } b_{-1} < b_{-2} \\ &= 0 \qquad\quad \text{if } b_{-1} = b_{-2}. \end{aligned} \tag{9.12b}$$

N is the length of the edge response in bit time, and $r_{\Delta t}(f_{\Delta t})$ is the symbol-spaced rising (falling) edge response at phase Δt. $S_{\Delta t}^{0,b_{-m},\ldots,b_{-1}}$ is defined similarly.

At initialization, we assume the link is at either a high, or low, steady state:

$$\begin{aligned} A_{0,\ldots,0}(0) &= V_{low} \\ A_{1,\ldots,1}(0) &= V_{high} \\ A_{others}(0) &= \infty \ . \end{aligned} \tag{9.13}$$

Because we are searching for the bit pattern that results in the worst-case undershoot at sample bit M, $A_{b_{-m},\ldots,b_{-1}}(M)$ with $b_{-1} = 0$ are set to be ∞, because the input bit at bit M is high. After filling in the A matrix, the worst-case undershoot is given by $\min(A(N))$. Tracing back from $\min(A(N))$, we get the worst-case input pattern that results in maximum undershoot.

The recursive relation for the input pattern that results in worst-case overshoot at sample bit M is similar to the preceding, except that now, at initialization $A_{others}(0)$ in (9.13) and $A_{b_{-m},\ldots,b_{-1}}(M)$ are set to be $-\infty$. Moreover, instead of taking the minimum, we choose the maximum in (9.12a).

The algorithm fills in the matrix A of size 2^m by N, where m is the order of MER (m=1 for DER), and N is the length of the edge responses. Therefore, the running time of the algorithm is $O(2^m N)$. Note that generally m is much smaller than N. Therefore, the running time is roughly $O(N)$.

A trellis graph [21] better illustrates the algorithm. In a trellis, each node corresponds to a distinct state at a given time, and each arrow represents a transition to some new state at the next instant of time. Figure 9.28 shows an example of a trellis graph. For the sake of simplicity, the example is based on DER. Therefore, there are only two distinct states, ($s = 1$) and ($s = 0$) For the mth order MER, the size of the state space is 2^m. Consider three post-cursor bits ($j = 0,1,2$) and one precursor bit ($j = 4$)

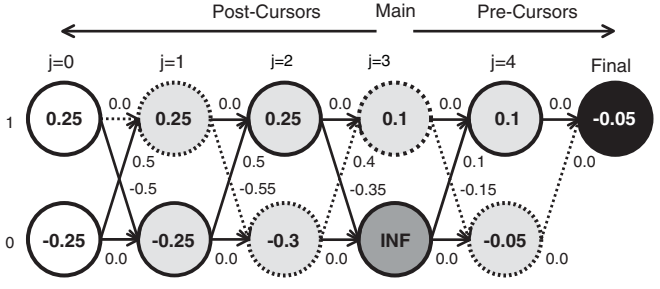


Figure 9.28 Trellis Graph for Worst-Case Pattern Search Algorithm

The number associated with each node (s, j) is the worst-case accumulated signal level at the main bit ($j = 3$) if the system is at state s at time j. It is the same definition as $A_{b_{-m},\ldots,b_{-1}}(j)$ in

(9.12a), with state s being (b_{-1}). For example, the possible states are initially 1 for DC high, and 0 for DC low. Therefore, the numbers associated with these two nodes are 0.25 (DC high) and –0.25 (DC low), respectively.

In this example, we look for the worst-case pattern for the upper eye (for example, the worst-case pattern that generates the lowest signal level at the main bit, given the main bit is a 1). Therefore, node (s=0, j=3) is invalid, and is shaded dark gray in Figure 9.28. INF (∞) is associated with the node to ensure that the optimal path is not going through it.

The number associated with each transition is the impact on the worst-case accumulated signal level at the main bit, if that transition happens. For example, for the transition from node (s=1, j=0) to node (s=1, j=1), there is no edge transition, therefore, there is no impact on the worst-case accumulated signal level at the main bit. For (s, j), the possible paths are from (1, j-1) and (0, j-1). Choose the path that generates the lower signal level, and represent it with a dotted line. For example, for (s=1, j=3), the path from (s=1, j=2) results in a worst-case accumulated signal level of 0.25, while the path from (s=0, j=2) results in a worst-case accumulated signal level of 0.1. Therefore, we choose the path from (s=0, j=2), and represent it with a dotted line. By following the dotted lines, the worst-case bit pattern is obtained. The number associated with the final node is the worst-case eye height.

Note that multiple paths can result in the same worst-case signal level. In this case, the algorithm can either remember all the choices, or simply pick one of them. The algorithm makes no assumptions about the system responses, other than that it can be characterized by MER.

9.5.3 Worst-Case Eye

So far, the discussion has covered peak distortion analysis, as applied to one sampling phase location. Although this results in the worst-case voltage opening at one sampling point, it does not result in worst-case link performance, because link performance depends on both voltage and timing margins. One can derive the worst-case data pattern for the overall system by performing the peak distortion analysis at multiple phases across the entire bit time. This results in a worst-case eye diagram that not only gives the worst-case eye height, but also the worst-case eye width. Because a typical channel is lossy and low pass, a few phases across the bit time are needed to trace out the inner eye. For highly reflective channels that exhibit more high-frequency components in their responses, more phase locations should be considered than are required with purely lossy channels.

9.5.4 Numerical Examples of Peak Distortion Analysis

Table 9.2 compares the worst-case eye heights, simulated by the SBR, DER, and MER methods. Additionally, the estimated worst-case eye heights are compared to the eye heights simulated by HSPICE, for the same worst-case data patterns. These simulations use the three examples previously described. In the differential signaling case, the SBR, DER, and MER methods are equivalent, and estimate the same worst-case eye height, which is very close to what HSPICE predicts. In all cases, the worst-case data pattern extracted by MER produces the worst-case eye opening seen in the HSPICE simulation. In the RDRAM example, DER provides sufficient accuracy. For

the POD example, a fifth-order MER is used to achieve the required accuracy. In this case, both the SBR and DER methods deviate significantly from the HSPICE simulation results.

Table 9.2 Worst-Case Eye Opening Estimated by SBR, DER, and MER for Three Systems (Units are in mV.)

	SBR/HSPICE	DER/HSPICE	MER/HSPICE
Differential signaling	604/604	604/604	604/604
RDRAM	785/802	797/803	793/798
POD	434/406	311/396	392/394

As discussed previously, peak distortion analysis is generally used to determine the worst-case eye height of the link at the sampling point. To estimate the worst-case eye width, the worst-case eye diagram is extracted by generating a worst-case data pattern at multiple phases across one bit time. Figure 9.29 compares the eye diagrams generated by the worst-case data pattern for the data sampling point, and by the worst-case data pattern for five phases across 1UI, using a 7" FR4 backplane channel with two connectors. The worst-case eye diagram generated for five phases produces a smaller inner eye than the one for a single phase, and therefore provides a more accurate estimation of the system margin. For example, the worst-case eye width is reduced from 125ps to 103ps, when the worst-case data pattern for five phases is used. The voltage margin at the center remains the same as in the single-phase case; however, in general, the worst-case eye opening in both voltage and timing is achieved.

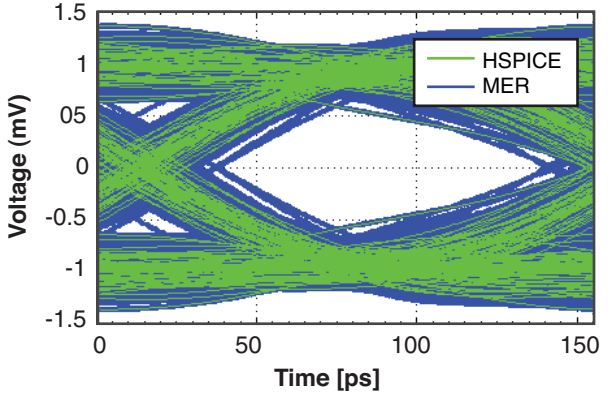


Figure 9.29 Simulated Eye Diagrams for the Worst-Case Data Pattern at the Sampling Point and at Five Phases Uniformly Distributed Across 1 UI (7" FR4 Backplane Channel with Two Connectors at 6.4Gb/s)

9.6 Summary

This chapter presents a hybrid simulation flow that mixes statistical and time-domain simulation methods to overcome some of the limitations of the statistical link analysis methods. Typically, fast time-domain simulation techniques are based on the superposition of either single-bit responses or edge responses. Of the edge responses, multiple-edge response (MER) is the method that captures system non-linearity without sacrificing simulation speed. It simulates millions to billions of bits in a matter of minutes; therefore, it gathers enough statistics for ISI probability distribution in a short time. By plugging the ISI distribution into statistical engines (such as LinkLab [22] and StatEye [23]), we can efficiently estimate BER for links with non-linear behavior.

Additionally, this chapter presents a worst-case pattern search algorithm for DER and MER. The dynamic programming algorithm described is guaranteed to find the worst-case pattern for systems that can be characterized with DER and MER. By extracting the worst-case data pattern at multiple phases, one can more accurately estimate the worst-case eye diagram. Although this chapter only demonstrates the method using a single-input, single-output system, the method can easily be extended to account for crosstalk in a multi-input, multi-output system.

References

1. M. Steinberger, T. Westerhoff, and C. White, "Demonstration of SerDes modeling using the Algorithmic Model Interface (AMI) standard," presented at the IEC DesignCon, Santa Clara, CA, 2008.
2. D. Oh, "Multiple edge responses for fast and accurate system simulations," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 163–166.
3. J. Ren and D. Oh, "Multiple edge responses for fast and accurate system simulations," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 4, pp. 741–748, Nov. 2008.
4. J. Ren, D. Oh, S. Chang and F. Lambrecht, "Statistical link analysis of high-speed memory I/O interfaces during simultaneous switching events," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2008, pp. 25–28.
5. D. Oh, S. Chang, and J. Ren, C, "Hybrid statistical link simulation technique," *IEEE Transactions on Advanced Packaging*, vol. 1, no. 5, pp. 772–783, May 2011.
6. http://en.wikipedia.org/wiki/Extreme_value_distribution
7. Y. Chang and D. Oh, "Fast ISI characterization of passive channels using extreme value distribution," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2007, pp. 127–130.

8. C.-C. Huang, D. Nguyen, D. Oh, W.-Y. Yip, and D. Secker, "RDRAM channel design with 32-bit 4.8GB/s memory modules," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2002, pp. 19–22.
9. K. Xiao, B. Lee, and X. Ye, "A flexible and efficient bit error rate simulation method for high-speed differential link analysis using time-domain interpolation and superposition," in *Proceedings of International Symposium on Electromagnetic Compatibility*, Detroit, MI, Aug. 2008, pp. 1–6.
10. H. J. Feldman and C. W. V. Hogue, "Probabilistic sampling of protein conformations: new hope for brute force?," *Proteins: Structure, Function and Genetics* 46:8–23.
11. D. C. Sullivan and I. D. Kuntz, "Distributions in protein conformation space: implications for structure prediction and entropy," *Biophysical Journal*, vol. 87, pp. 113–120, July 2004.
12. V. Stojanovic and M. Horowitz, "Modeling and analysis of high-speed links," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2003, pp. 589–594.
13. D. Oh and S. Chang, "Clock jitter modeling in statistical link simulation," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2009, pp. 49–52.
14. G. Balamurugan, J. E. Jaussi, M. Mansuri, F. O'Mahony, and J. Kennedy, "Modeling and analysis of high-speed I/O links," *IEEE Transactions on Advanced Packaging*, vol. 32, no. 2, pp. 237–244, Nov. 2009.
15. J. G. Proakis, *Digital Communications*, 4th ed., Singapore: McGraw-Hill, 2001, pp. 617–618.
16. H.-J. Liaw, X. Yuan, and M. A. Horowitz, "Technique for determining performance characteristics of electronic devices and systems," US Patent 6920402, Mar. 7, 2001.
17. B.K. Casper, M. Haycock, and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," in *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 2002, pp. 54–57.
18. F. Lambrecht, C.-C. Huang, and M. Fox, "Technique for determining performance characteristics of electronic systems," U.S. Patent 6775809, Mar. 14, 2002.
19. V. Drabkin, C. Houghton, I. Kantorovich, and M. Tsuk, "A periodic resonant excitation of microprocessor power distribution systems and the reverse pulse technique," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2002, pp. 175–178.
20. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, 2001, pp. 323–370.
21. D. J.C. Mackay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003, pp. 326.

22. D. Oh, F. Lambrecht, S. Chang, Q. Lin, J. Ren, C. Yuan, J. Zerbe, and V. Stojanovic, "Accurate system voltage and timing margin simulation in high-speed I/O system designs," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 4, pp. 722–730, Nov. 2008.
23. A. Sanders, M. Resso, and J. D'Ambrosia, "Channel compliance testing utilizing novel statistical eye methodology," presented at the IEC DesignCon, Santa Clara, CA, 2004.

This page intentionally left blank

CHAPTER 10

Clock Models in Link BER Analysis

Sam Chang, Jihong Ren, and Dan Oh

Jitter, associated with the clocking circuitry, is one of the largest timing error components in high-speed links. Clock jitter is generated by clock generation (PLL) circuits, and clock distribution (buffers) circuits. The dominant source of clock jitter is power supply noise (see Chapter 14, "Supply Noise and Jitter Characterization" and Chapter 2, "High-Speed Signaling Basics" Section 2.2.5). Because the jitter induced by power supply noise is strongly colored due to the frequency-dependent supply impedance profile and circuit sensitivity, modeling the frequency content of this jitter is crucial, in order to determine accurate channel margin. Traditionally clock jitter is modeled as an additional jitter term in the timing specification. This can lead to an overly pessimistic timing specification (as discussed in Chapter 6, "Channel Voltage and Timing Budget"), because the correlation between the clock and data signals is ignored. However, depending on the clocking topology, a significant amount of clock jitter could be ignored due to the jitter cancellation between the data and clock signals at the receiver. Modeling this jitter cancellation requires an advanced statistical link simulator, which models both the data and clock channels, and the spectrum of jitter sources.

To illustrate this point, let us consider a forwarded clocking system as an example. Figure 10.1 illustrates the channel description, along with various noise sources. Note that there is no PLL present at the receiver in this example. The transmit side noise induces jitter, which is common to both the data and forwarded clock paths. At the receiver side, the sampler uses this jittery clock to sample data signals that contain the same jitter, so the transmitter jitter is canceled at the receiver. This cancellation occurs only if the delays in the data and clock paths are identical. This is a very hard requirement to meet in practice, because there is always some skew between the data and clock paths because clock is distributed to multiple data pins. Consequently, a typical clock path is usually longer, due to the clock distribution at the receiver side, resulting in imperfect jitter cancellation. However, if the frequency of the jitter is sufficiently low, then the clock to

data skew can be neglected. Figure 10.1 also shows several common power supply–related noise sources. The frequency contents of these noise sources range from a few kHz to hundreds of MHz. Finally, note that the jitter added at the receiver side cannot be canceled, because it is only added to the clock path.

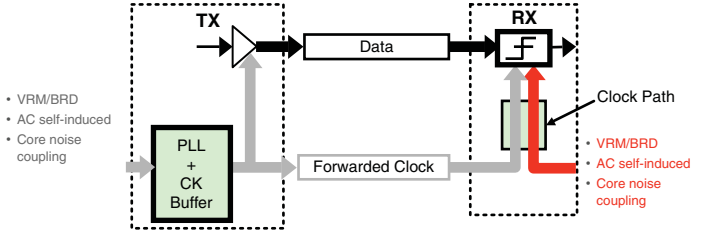


Figure 10.1 Forwarded Clock System with Various Noise Sources

As shown in this example, clock jitter modeling is very complicated, as it needs to account for jitter spectrum, passive data and clock channels, and the on-chip clock path. In fact, jitter cancellation or tracking is effective only when there is minimum jitter amplification on the data and clock paths, because the jitter amplification behaviors of the data and clock channels are distinct.

This chapter extends the statistical link simulation framework (presented in Chapter 8, "Link BER Modeling and Simulation") to model common clocking architectures [1] [2]. First, the general formulation (shown in Section 8.2.1) is modified to incorporate a common clock jitter source (described in Section 10.1). This formulation extends the equivalent voltage noise concept to common jitter source. The resulting model accounts for any jitter tracking between the data and clock signals. Section 10.2 describes the detailed models for commonly used clocking architectures. Section 10.3 discusses Clock and Data Recovery (CDR) modeling issues. Finally, Section 10.4 presents a jitter impulse function model of a passive clock channel, and explains jitter amplification due to channel ISI.

10.1 Independent and Common Clock Jitter Models

The mathematical formulation for the equivalent voltage noise concept can be extended to include a jitter source that is common to both transmitter and receiver. One example is a forwarded clock used to receive data. In this case, the receiver clock jitter has the same jitter source as the data signal. Another example is a common clock architecture that shares one clock source for both the transmitter and receiver. Adding this common jitter term to (8.4), we have:

$$y_m \cong y^M + y^{ISI} + n^{TX} + n^{RX} + n^{Common} \tag{10.1}$$

where y^M is the ideal received signal without ISI, and y^{ISI} is the amount of ISI at the current sampled location. n^{TX} and n^{RX} represent the equivalent voltage noise for the independent transmitter

and receiver timing jitter, ε^{TX} and ε^{RX}, respectively. n^{Common} represents the equivalent voltage noise for the common source timing jitter, $\vec{\varepsilon}^{Common}$.

The expressions of n^{TX} and n^{RX}, given in Equations (8.11) and (8.18), can be generalized to include models for on-chip clock paths, as follows:

$$n^{TX} = \vec{a}^T \mathbf{W} \mathbf{H}^{TX} \mathbf{J}_M(\vec{\zeta}^{TX}) \vec{\varepsilon}^{TX} \tag{10.2a}$$

$$n^{RX} = \vec{a}^T \mathbf{W} \vec{H}^{RX} \left(\vec{\zeta}^{RX} \right)^T \vec{\varepsilon}^{RX} \tag{10.2b}$$

where $\vec{\zeta}^{TX}$ and $\vec{\zeta}^{RX}$ are the jitter impulse functions (JIF) of Tx and Rx clock paths, respectively. $\mathbf{J}_M(\vec{\zeta}^{TX})$ is the JIF matrix that translates the input jitter to the jitter at the driver location. Each row of $\mathbf{J}_M(\vec{\zeta}^{TX})$ is the delayed version of JIF, and the inner product of these rows with $\vec{\varepsilon}^{TX}$ gives the final transmitter jitter at the driver. Section 8.4 of Chapter 8 provides the definitions for the remaining symbols, along with Equations (8.11) and (8.18). The JIF concept is covered later in Section 10.4.

Similarly, deriving the expression of n^{Common} is not hard:

$$n^{Common} = \vec{a}^T \mathbf{W} \begin{bmatrix} \mathbf{H}^{TX} & \vec{H}^{RX} \end{bmatrix} \begin{bmatrix} \mathbf{J}_M(\vec{\zeta}^{TX}) \\ (\vec{\zeta}^{RX})^T \end{bmatrix} \vec{\varepsilon}^{Common}. \tag{10.2c}$$

The preceding equations have the same format as the previous expressions in Chapter 8. Therefore, the same algorithms and procedures (described in Section 8.4) can be applied to calculate probability density functions (PDFs).

10.2 Modeling Common Clocking Schemes

Equations (10.2a, b, and c) are in a general form, which can be used for any clocking topology. This section presents explicit jitter modeling formulae for common interconnect systems, such as serial links, parallel buses, and memory channels [1] [2]. The clocking architectures that are discussed include memory interfaces (such as DDR, GDDR, XDR, and Mobile XDR systems) and serial interfaces (such as PCIe, FlexIO, and Elastic Interface systems).

First, the expressions for the transmitter and receiver input jitter sources are derived and then these expressions are used to excite the input sources in a time-domain simulation (described in Chapter 9, "Fast Time-Domain Channel Simulation Techniques"). Finally, the equivalent voltage noise expressions for statistical simulation are derived. This section also provides short descriptions of the pros and cons of each clocking architecture.

10.2.1 CDR-Based Serial Links

Figure 10.2 illustrates a CDR-based system. (Refer to Section 10.3 for a discussion of the basics of a CDR-based link.) This section focuses on the modeling aspects of a CDR-based link. First consider the transmitter jitter model. The Tx jitter, $\vec{\varepsilon}^{TX}$, is the input reference clock jitter $\vec{\varepsilon}^{Ref1}$ convolved with the jitter impulse function of the Tx clock path $\vec{\zeta}^{TX}$, as follows:

$$\vec{\varepsilon}^{TX} = \vec{\zeta}^{TX} * \vec{\varepsilon}^{Ref1}. \tag{10.3a}$$

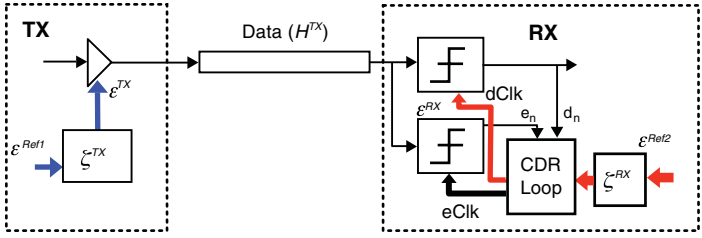


Figure 10.2 CDR-Based Serial Link

The jitter impulse function, $\vec{\zeta}^{TX}$, models the transmitter on-chip clock path, which includes the PLL and clock distribution. The transmitter jitter, $\vec{\varepsilon}^{TX}$, is mapped into the equivalent voltage noise seen at the receiver, as follows:

$$n^{TX} = \vec{a}\mathbf{W}\mathbf{H}^{TX}\mathbf{J}_M(\vec{\zeta}^{TX})\vec{\varepsilon}^{Ref1}. \tag{10.3b}$$

In a CDR-based link, the receiver sampling clock jitter, ε^{RX}, consists of two components: the native receiver clock jitter from the reference clock, $\vec{\varepsilon}^{Ref2}$, and the CDR phase index dithering ε^{CDR}:

$$\varepsilon^{RX} = \varepsilon^{CDR} + (\vec{\zeta}^{RX})^T \vec{\varepsilon}^{Ref2}. \tag{10.4a}$$

The jitter impulse function, ζ^{RX}, models the receiver on-chip clock path, including the PLL and clock distribution. Note that the native receiver clock jitter has an impact on the CDR dithering, because CDR tracks the low-frequency receiver clock jitter.

The CDR dithering behavior can be modeled in either a time domain [5], or in a statistical domain using a first-order Markov chain. (Section 10.3.2 describes the Markov chain model in detail.) The balance of this subsection briefly discusses the time-domain model and the pros and cons of the time and statistical domain approaches.

In a time-domain simulation, the CDR phase index is generated by applying the CDR filtering logic to the incoming data and edge samples. Any additional receiver clock jitter can be added to the CDR phase indexes, when the data and edge samples are collected. By doing so, the impacts of ISI, the CDR filtering algorithm, and the receiver clock jitter are modeled naturally.

However, time-domain simulation is generally time-consuming. Simulating random jitter in the time-domain is impractical due to the limited number of bits.

On the other hand, the impact of receiver random jitter on CDR dithering can be captured by mapping the random jitter into equivalent voltage noise seen by the receiver:

$$n^{RX} = \vec{a}^{T}\mathbf{W}\vec{H}^{RX}\left(\vec{\zeta}^{RX}\right)^{T}\vec{\varepsilon}^{Ref2}. \tag{10.4b}$$

The impact of this reference clock jitter on the CDR phase dithering is accounted for by including the equivalent voltage noise in the state transition probability of the Markov chain model (see Section 10.3.2).

The Markov chain model efficiently captures the impact of ISI and edge selection algorithms on CDR dithering, as well as the noise averaging effect of the CDR filtering algorithm, as long as a strong correlation between noise and ISI exists only within a window covered by the CDR loop filter. Therefore, any low-frequency jitter components, which are within the tracking bandwidth of the CDR, should be carefully removed in the simulation.

However, the Markov chain model does not capture the impact of CDR loop latency, which may cause jitter peaking if it is excessive [6]. Time-domain simulation is required to model CDR loop latency. Unfortunately, time-domain simulation is slow and impractical when modeling the impact of random jitter and noise sources on CDR dithering. Figure 10.3 shows the impact of CDR loop latency on jitter tolerance for a 2x-oversampled CDR, simulated with a time-domain CDR behavior model in AMI (refer to Chapter 9). The channel model, used in the simulation, represents a simple setup for receiver testing, consisting of S-parameters for package and test traces. The data rate is 6.4 Gb/s. Note that the CDR loop latency has less impact on the jitter tolerance at higher frequencies. At lower frequencies, large loop latency results in less jitter tolerance, resulting in poorer CDR performance. This kind of time-domain phenomenon can be accurately captured in time-domain simulation, but not in statistical domain.

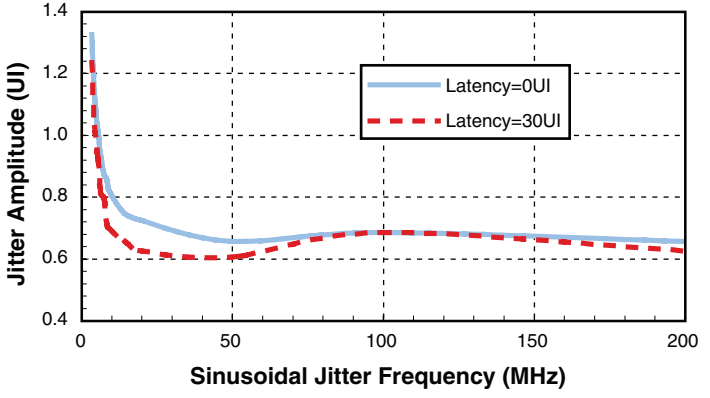


Figure 10.3 CDR Jitter Tolerance vs. Loop Delay

Finally, here are a few pros and cons of CDR: Because CDR does not require a clock to be forwarded, it requires less I/O pins. It also allows the Tx and Rx clock frequencies to differ slightly. However, its usage with an on-board I/O interface (such as parallel bus or memory interface) has the following limitations:

- Data coding is required to provide edge transitions.
- Jitter tracking of the data signal is limited to low frequencies (<10MHz).
- For bidirectional links, the bus turnaround time can be quite large.

10.2.2 PCIe Channels with a Common Clock Source

PCIe channels, in on-board applications, commonly share one clock source between both the transmitter and receiver (see Figure 10.4). Based on the figure, the time-domain jitter model can be written as:

$$\vec{\varepsilon}^{TX} = \vec{\zeta}^{TX} * \mathrm{TD}(\vec{H}^{Clk1}, \vec{\varepsilon}^{Ref}) \tag{10.5a}$$

$$\varepsilon^{RX} = (\vec{\zeta}^{RX})^{T} \mathrm{TD}(\vec{H}^{Clk2}, \vec{\varepsilon}^{Ref}) \tag{10.5b}$$

where $\vec{\varepsilon}^{Ref}$ is the jitter source and $\vec{H}^{Clk1}$ and $\vec{H}^{Clk2}$ are the impulse function of on-board traces from the clock source to Tx and Rx, respectively. $\vec{\zeta}^{TX}$ and $\vec{\zeta}^{RX}$ are the jitter impulse functions of the on-chip Tx and Rx paths, which include both the PLL and on-chip clock distribution. First, a time-domain simulation is performed using $\vec{\varepsilon}^{Ref}$ and $\vec{H}^{Clk}$ to generate the jitter sequence. This operation is noted as TD in the previous equ ations. Then, this jitter sequence is convolved with the jitter impulse response of $\vec{\zeta}^{TX}$, to calculate the final jitter sequence $\vec{\varepsilon}^{TX}$. The final jitter sequence is used as the transmitter jitter sequence for the time-domain simulation of the data signal. A similar procedure is applied to calculate the receiver jitter sequence $\vec{\varepsilon}^{RX}$.

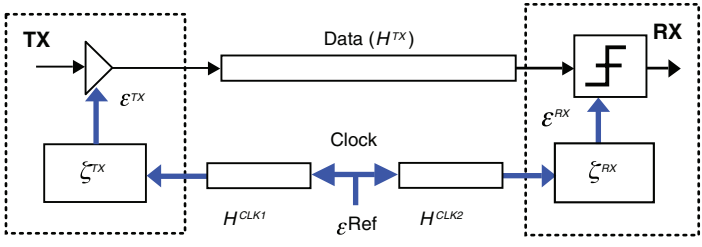


Figure 10.4 PCIe Channel

For the statistical-domain jitter model, the equivalent voltage noise seen by the receiver due to reference clock jitter is:

$$n^{Ref} = \vec{a}^T \mathbf{W}[\mathbf{H}^{TX} \quad \vec{H}^{RX}] \begin{bmatrix} \mathbf{J}_M\left(\vec{H}^{Clk1} * \vec{\zeta}^{TX}\right) \\ \left(\vec{H}^{Clk2} * \vec{\zeta}^{RX}\right)^T \end{bmatrix} \vec{\varepsilon}^{Ref}. \tag{10.6}$$

As shown in the preceding equation, the jitter tracking depends on the delay difference between the Tx and Rx clock paths, and the PLL characteristic difference between Tx and Rx. When the two clock paths are ideal (without any jitter amplification) and two PLLs on Tx and Rx sides are identical, the jitter tracking percentage can be defined as:

$$\begin{aligned} \text{Tracking Percentage}(\%) &= 100 - 100 \times \frac{\delta_{NetRJ}}{1.414\delta_{InputRJ}} \\ &= 100 - 100 \times \left[\frac{\int PSD(\omega) \left| \left(1 - \exp(j\omega\tau_{skew})\right) \right|^2 d\omega}{2\int PSD(w)dw} \right]^{0.5} \end{aligned} \tag{10.7}$$

where τ_{skew} is the skew between the Tx and Rx paths. $\delta_{InputRJ}$ represents the input random jitter, and δ_{NetRJ} represents the net link random jitter after tracking. Figure 10.5 (b) shows the design curves for the different input PSDs. Simplified input PSD responses shown in Figure 10.5(a) are used for this calculation. These design curves can be used to estimate the net jitter given the input noise bandwidth and skew.

10.2.3 Clock-Forwarding Scheme for Parallel Bus Interfaces

The clock-forwarding architecture, shown in Figure 10.6, is the most commonly used clocking scheme for I/O interfaces between on-board devices (such as parallel buses and memory channels). Based on Figure 10.6, the time-domain jitter model is expressed as:

$$\vec{\varepsilon}^{TX} = \vec{\zeta}^{Cntl} * \vec{\varepsilon}^{Ref} \tag{10.8a}$$

$$\varepsilon^{RX} = \left(\vec{\zeta}^{DRAM}\right)^T TD\left(\vec{H}^{Clk}, \vec{\zeta}^{Cntl}, \vec{\varepsilon}^{Ref}\right) \tag{10.8b}$$

where $\vec{H}^{Clk}$ is the impulse function of the on-board clock network. $\vec{\zeta}^{Cntl}$ is the jitter impulse function of the controller on-chip path, including the PLL and clock distribution, whereas, $\vec{\zeta}^{DRAM}$ is the jitter impulse function of the DRAM clock distribution (which may or may not have PLL or DLL). The jitter sequence for the transmitter jitter is generated by simply convolving $\vec{\varepsilon}^{Ref}$ and the jitter impulse response of $\vec{\zeta}^{Cntl}$. For the receiver jitter, we simulate the calculated transmitter jitter sequence with the clock impulse response in time domain to generate the jitter sequence at the DRAM. Then, this jitter sequence is convolved again, with the jitter impulse response of $\vec{\zeta}^{DRAM}$, to calculate the receiver sampling locations.

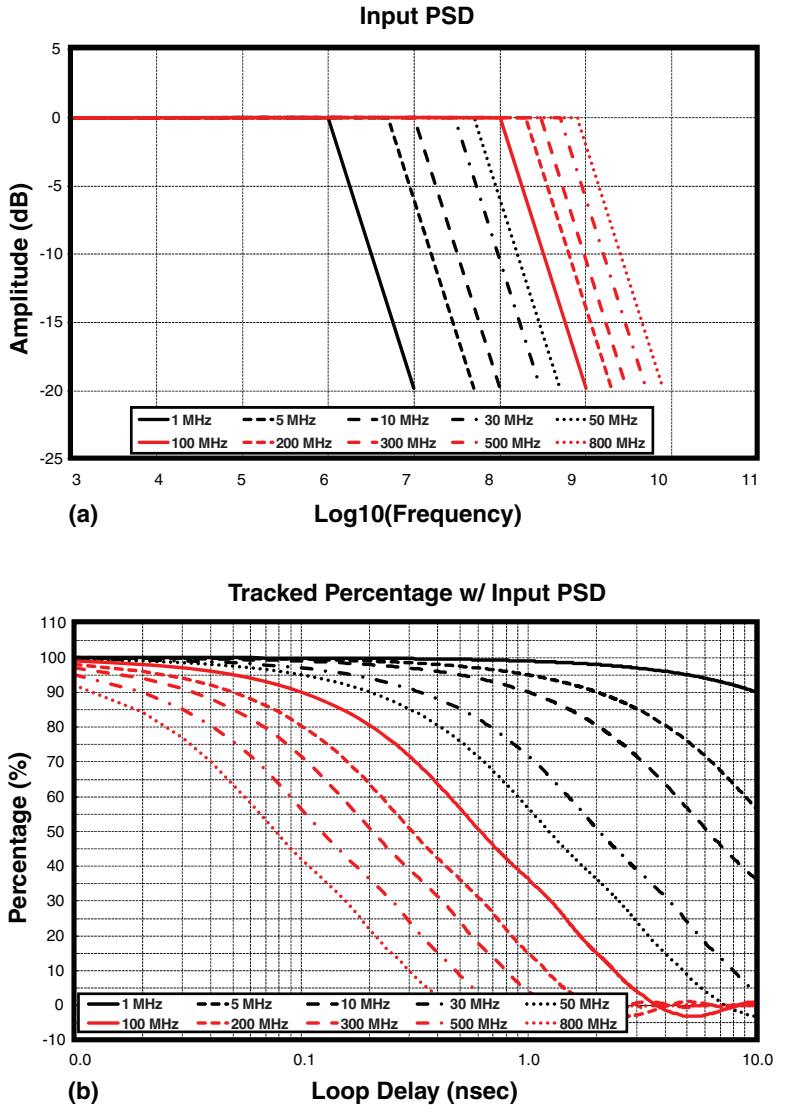


Figure 10.5 (a) Various Input PSD Bandwidths and (b) Corresponding Jitter Tracking Percentages as a Function of Skew

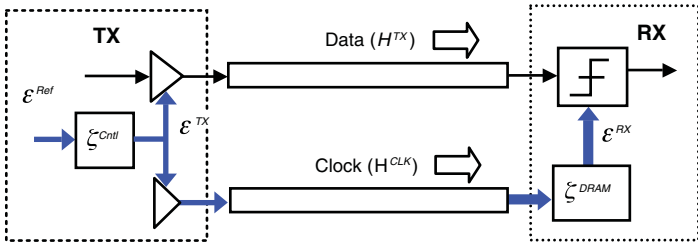


Figure 10.6 The Clock-Forwarding Architecture for Many Parallel Interfaces (Including FlexIO, ElasticIO, DDR, and GDDR Systems)

For the statistical-domain jitter model, the equivalent voltage noise seen by the receiver due to the reference clock jitter is:

$$n^{Ref} = \vec{b}^{T}[\mathbf{H}^{TX} \quad \vec{H}^{RX}] \begin{bmatrix} \mathbf{J}_M(\vec{\zeta}^{Cntl}) \\ (\vec{\zeta}^{Cntl} * \vec{H}^{Clk} * \vec{\zeta}^{DRAM})^T \end{bmatrix} \vec{\varepsilon}^{Ref}. \tag{10.9}$$

In this case, the jitter tracking depends on the delay difference between the data path and the clock path, including $\vec{H}^{Clk}$ and $\vec{\zeta}^{DRAM}$. Typically, on-board trace length mismatches are small, and the tracking depends mostly on $\vec{\zeta}^{DRAM}$.

Figure 10.7 demonstrates a jitter tracking mechanism for the forwarded clock architecture, which simulates eye diagrams with and without tracking. Figure 10.7(a) assumes 32ps (peak-to-peak), and 50MHz SJ is only injected in the data path, while Figure 10.7(b) assumes the same amount of SJ injected into both the data, and the forwarded clock. The eye diagrams clearly show that, in the latter case, most of the injected jitter is tracked out.

10.2.4 Asymmetric Clocking Scheme for Moderate Performance I/O Interfaces

While the previously described clock-forwarding architecture provides excellent performance, implementing it can be quite costly, because it requires a transmitter to continuously send a clock signal. Consequently, both ends of the devices need to have some sort of timing circuitry, such as PLL or DLL. This could be an issue for cost-sensitive applications, such as memory interfaces. To mitigate this cost issue, an alternative asymmetric clocking scheme, which requires timing circuitry at only one side of the interface, can be used. With memory interfaces, the timing circuitry often resides in the controller.

During a write operation, a common clock-forwarding scheme is used, as described in the previous section: The controller forwards a clock signal, along with the data (see Figure 10.6). During a read operation, there is no return clock, and the controller's internal clock is used to sample the data sent by the DRAM (see Figure 10.8). GDDR5 is the first system to use this asymmetric clocking scheme. Recently, Mobile XDR, which is a differential memory interface, also adopted a similar clocking architecture [7]. Because the skew between the data and the clock delay is significantly larger for read operations than it is for writes, keeping the passive channel

length short is important to minimize the skew and, in turn, maximize the jitter tracking between the data and clock.

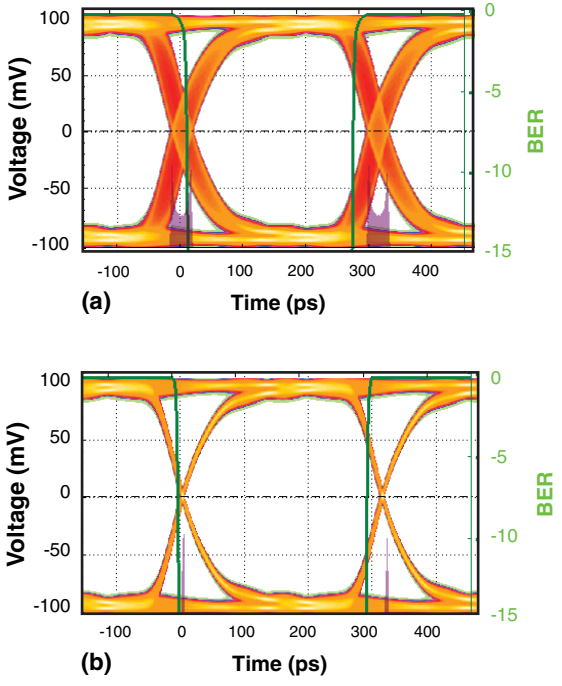


Figure 10.7 Received Eye Diagram with (a) Ideal Clock and (b) Jittery Clock

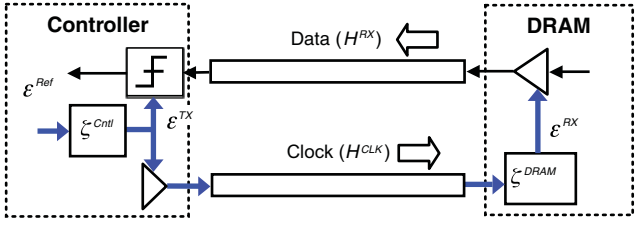


Figure 10.8 Read Case Clock Channel Model for GDDR5 or Mobile XDR Interfaces

Based on Figure 10.8, the time-domain jitter model for a read operation is expressed as:

$$\vec{\varepsilon}^{TX} = \vec{\zeta}^{DRAM} * TD\ (\vec{H}^{Clk} * \vec{\zeta}^{Cntl} * \vec{\varepsilon}^{Ref}) \tag{10.10a}$$

$$\varepsilon^{RX} = (\vec{\zeta}^{Cntl})^{T}\, \vec{\varepsilon}^{Ref}. \tag{10.10b}$$

In this case, the receiver jitter is the same as the transmitter jitter for the write case. To compute the transmitter jitter, the receiver jitter sequence is simulated in the time domain. Then, the resulting jitter sequence is convolved with the jitter impulse response of $\vec{\zeta}^{DRAM}$.

For the statistical-domain jitter model, we have:

$$n^{Ref} = \vec{b}^{T}\left[\mathbf{H}^{RX} \quad \vec{H}^{RX}\right] \begin{bmatrix} \mathbf{J}_M(\vec{\zeta}^{Cntl} * \vec{H}^{Clk} * \vec{\zeta}^{DRAM}) \\ (\vec{\zeta}^{Cntl})^{T} \end{bmatrix} \vec{\varepsilon}^{Ref}. \tag{10.11}$$

This clocking scheme suffers most, in terms of jitter tracking, due to the large skew between the data and the clock path. However, it does not require any timing circuitry on the DRAM side, so its implementation cost and power consumption are low. The skew amount directly affects the amount of jitter to be tracked out. Figure 10.9 represents a Mobile XDR running at 3.2Gb/s per link. The figure shows the timing margin loss, due to different sinusoidal jitter, while using the same amplitude of 28ps peak-to-peak. The simulation is performed with a 1.5-ns skew. As expected, jitter tracking depends on the source spectrum. In this example, any jitter below 150MHz is in the tracking region, because the jitter in the data and clock signals are in phase up to this frequency. The jitter from 150MHz to 500MHz is in the anti-tracking region, as the jitter in the two signals is out of phase.

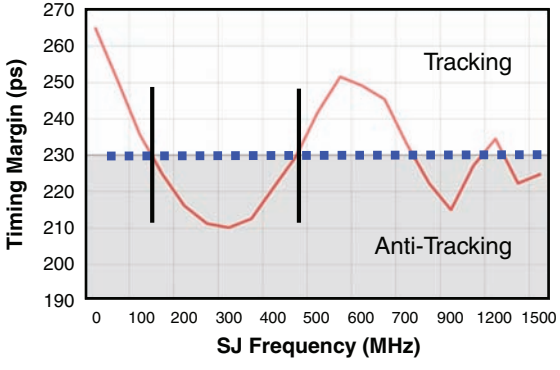


Figure 10.9 Link Margin vs. Sinusoidal Jitter

10.3 CDR Circuitry Modeling

Serial link applications typically use a timing recovery circuit to extract clocking information from the incoming data stream, which means no explicit clock signal is sent with the data signal. This clocking scheme is particularly useful for off-board I/O interfaces, where two transceiver devices may use different clock sources. Such systems are called plesiochronous systems. Because the clock signal is recovered from the data signal, the quality of the recovered clock in a CDR-based system is a strong function of the transmitted data signal. Thus, accurate modeling of CDR requires a complete link model. This section briefly reviews the CDR basics and covers the Markov-chain-based model [3] [4] for link analysis.

10.3.1 CDR Basics

Figure 10.10 shows a 2x oversampled CDR, commonly used in serial links. It uses data samples to detect transitions, and edge samples to detect timing errors. In Figure 10.10, the phase mixer creates a fixed number of phases from the PLL. The phase control logic retards or advances the phase mixer output, depending on an early-versus-late determination.

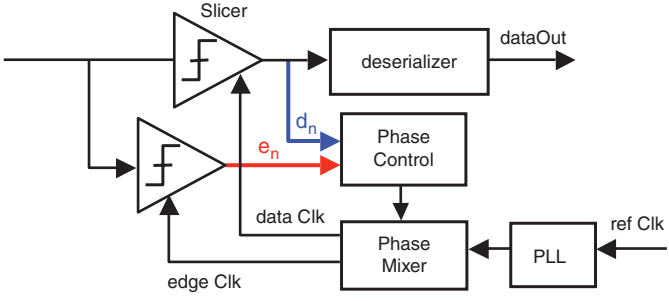


Figure 10.10 2x Oversampling CDR

Early and late information are generated based on the data sequence, d_n, and the edge sample, e_n. For example, Figure 10.11 shows a negative (positive) edge sample for a rising (falling) edge, which implies that the sampling clock is early with respect to the incoming data. Similarly, a positive (negative) edge sample for a rising (falling) edge implies the sampling clock is late. To reduce the impact of high-frequency noise and ISI on CDR dithering, we accumulate and filter early and late information, in order to generate up, down, and hold for the phase-control logic. The amount of accumulation and filtering determines the CDR bandwidth.

Nominally, CDR locks to the mean of the timing distribution at the transition, where about half of the edges are late and half are early. Therefore, CDR nominally locks to the mean zero-crossing phase for all edges (for NRZ) [11].

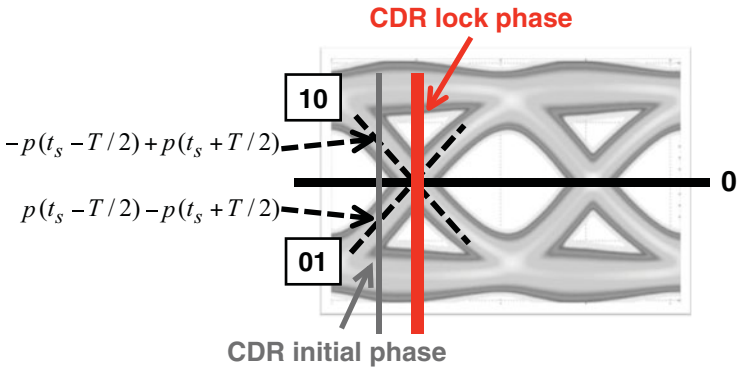


Figure 10.11 CDR Nominal Locking Phase (10 and 01 Data Transitions Shown)

Assume that $p(t)$ is the single-bit response of the channel, t_s is the data-sampling phase, and a_m is the mth transmitted symbol. The transition sample preceding the mth data sample is:

$$z_{m-1/2} = a_m p(t_s - T/2) + a_{m-1} p(t_s + T/2) + \sum_{k=1\ldots} a_{m-1-k} p(t_s + T/2 + kT). \quad (10.12)$$

For rising transitions ($a_m = 1, a_{m-1} = -1$), and assuming DC balance in the input data, the mean of the transition samples $z_{m-1/2}$ is $p(t_s - T/2) - p(t_s + T/2)$. (A similar analysis is used for the falling transitions.) When $p(t_s - T/2) - p(t_s + T/2) = 0$, one-half of the rising transitions are early and one-half are late, as shown in Figure 10.11 [11]. Thus, CDR nominally locks to the phase t_s that satisfies:

$$p(t_s - T/2) - p(t_s + T/2) = 0. \quad (10.13)$$

10.3.2 Statistical Model of CDR Based on Markov Chain

The CDR phase control logic is inherently a state machine. The probability for the CDR to change from one phase index to another is determined by noise amount, as well as the CDR filtering algorithm. This behavior can be naturally modeled by a Markov chain [3]. This section describes in detail a statistical CDR model, based on a first-order Markov chain [3]. Figure 10.12 shows possible phase positions of the recovered clock in a Markov chain. The transitions between the states are governed by the hold, up, and down decisions to hold, advance, or retard the current phase i. In an environment with noise and ISI, these transitions have associated probabilities, $p_{hold,i}$, $p_{up,i}$, and $p_{down,i}$ for every phase state. The transition probabilities are found from the statistics of the input data and noise. Moreover, to increase the probability of making the right decision, designers typically filter the early/late decisions. After the transition probabilities for

each of the phase states are found, the Markov chain transition matrix **T** is formed, and the steady-state phase probabilities are calculated by solving for transitions iteratively, as follows:

$$\vec{p}_{n+1}^{\,\phi} = \mathbf{T} \cdot \vec{p}_{n}^{\,\phi} \tag{10.14}$$

where $\vec{p}_n^{\,\phi}$ is the phase probability distribution vector, and **T** is the transition matrix, which is written by:

$$\mathbf{T} = \begin{bmatrix} p_{hold,1} & p_{dn,2} & 0 & 0 & \cdots & p_{up,L} \\ p_{up,1} & p_{hold,2} & p_{dn,3} & 0 & \cdots & 0 \\ 0 & p_{up,2} & p_{hold,3} & p_{dn,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ p_{dn,1} & 0 & 0 & 0 & \cdots & p_{hold,L} \end{bmatrix}. \tag{10.15}$$

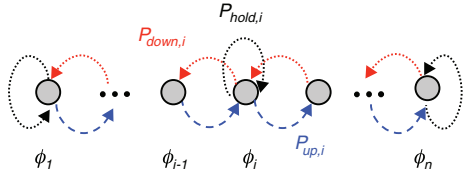


Figure 10.12 First-Order Markov Chain Model for CDR

Finally, the calculated probabilities, at every phase location, are modeled as the receiver sampling distribution (as described in Section 10.2.1).

10.3.3 Validation of CDR Model

This section discusses using a backplane serial link to demonstrate how to correlate the CDR model. The simulation and lab environment is composed of a 14-layer, 30" FR4 backplane channel, running at 5Gb/s (see Figure 10.13). The voltage transfer function for a 50-ohm reference is plotted (insertion loss over frequency). Although the length and transfer characteristics are typical for a backplane serial link, the 100-mil backplane via stub and the 60-mil line-card via stub result in a 22-dB insertion loss at 2.5GHz.

Transmitter jitter parameters are again captured by the DCA-J, while the receiver jitter parameters are extracted from the circuit model. To compare the modeled CDR processes to the real CDR behavior, data on the clock recovery circuitry is measured in the lab by collecting position information over a specified amount of time. The comparison between the lab measurement and the statistical simulation of the CDR phase position in Figure 10.14 shows a very good correlation.

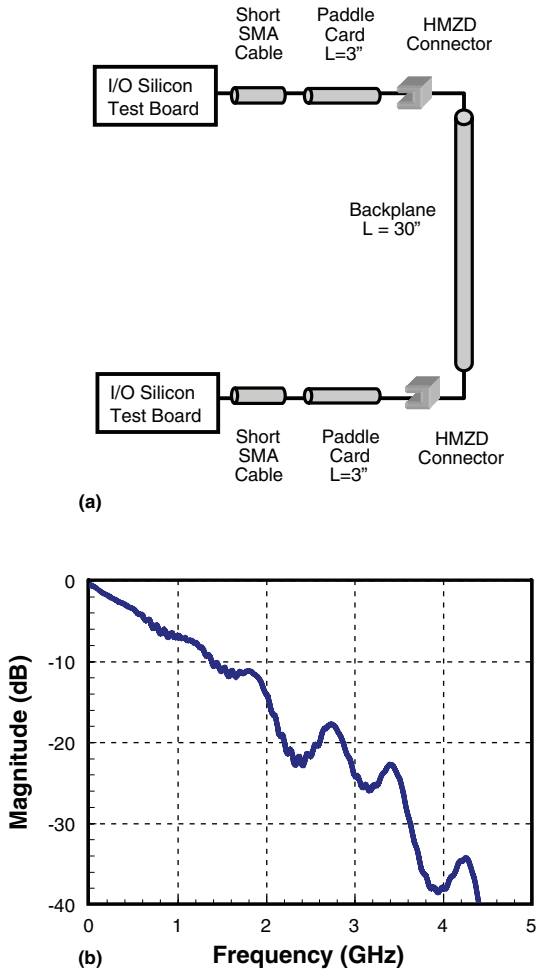


Figure 10.13 Typical High-Speed Backplane Channel and Its Transfer Function

The CDR position is critical, because it determines the sampling position, and thus the behavior of much of the receiver circuitry. Figure 10.15 illustrates performance comparisons between lab measurements and a LinkLab simulation. The figure provides results for both a 30″ backplane channel, and a 16″ backplane trace. This (+/– mV) voltage margin is defined to be at BER 10^{-15}. If this voltage margin is applied to the receiver as an additional offset, it would result in a BER of 10^{-15}. We use extrapolation, based on an error function, for the measurement data up to 10^{-6} BER. Again, the simulation produced a good estimate of the actual link performance.

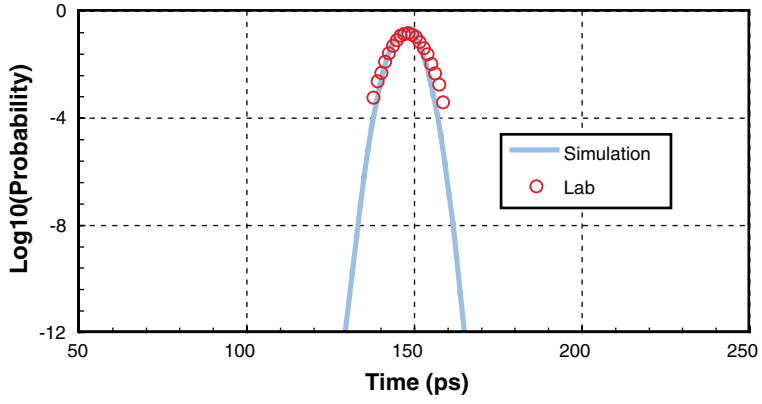


Figure 10.14 Lab vs. Simulation Comparison of CDR Phase Probability

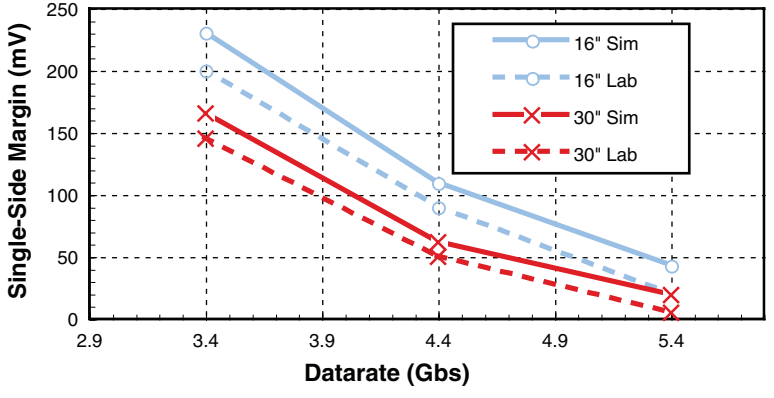


Figure 10.15 Lab vs. Simulation Comparison of Extrapolated Margin

Figure 10.16 shows the system voltage margin data with additional simulation results, using a simple receiver model that assumes an ideal sampling location at the center of eye (rather than using the more complete CDR behavioral model). As shown in this plot, the difference between including, and not including, the CDR model has an effect of ± 5 to 19 mV, or 10 to 38 mV peak-to-peak, for the given channel and data rate. This difference is not constant across frequency, and varies between channels; so, a simplified single offset term could lead to inaccurate simulation results.

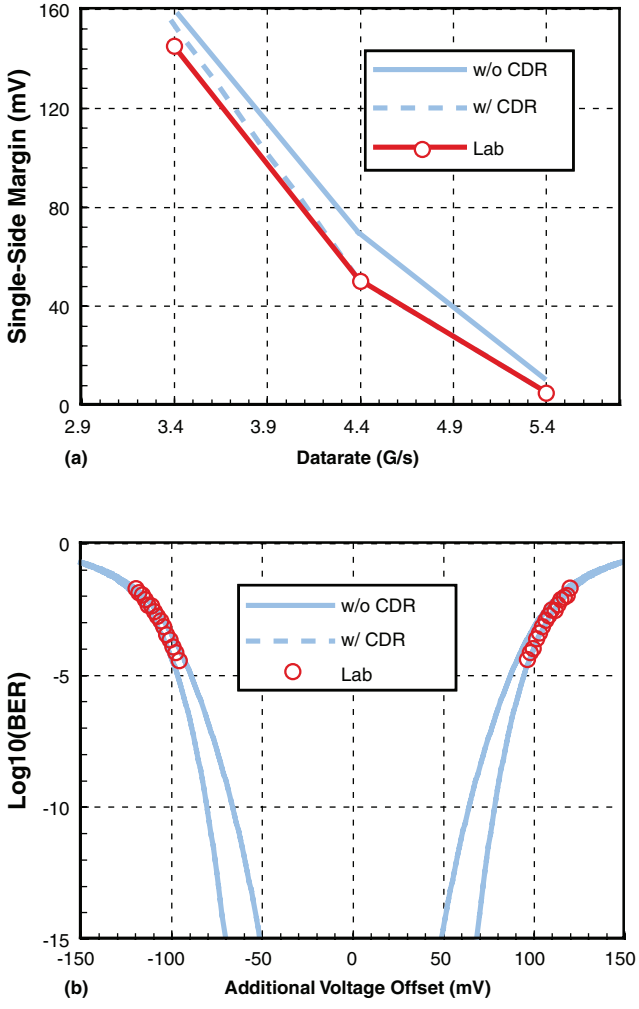


Figure 10.16 (a) Margin Results vs. Data Rate for Receiver Models with/without Statistical CDR Model, and Lab Measurement, and (b) Sample Voltage Bathtub Plot for Same 3 Data Sets at 4.4Gb/s Frequency Point

10.4 Passive Channel JIF and Jitter Amplification

As shown in the previous section, a clock is often forwarded along with data for high-speed parallel interfaces. At high frequency, channel attenuation [8–10] significantly amplifies clock jitter. This section describes a closed-form jitter impulse function model [10] for a passive channel, and illustrates the jitter amplification due to channel attenuation.

As discussed in Chapter 9, the output signal $y(t)$ can be expressed in terms of the superposition of progressively delayed channel step responses, as follows:

$$y(t) = \sum_k b_k p(t - kT) = \sum_k (b_k - b_{k-1}) s(t - kT). \tag{10.16}$$

Transmitter jitter on any one edge affects the superposed waveform in the vicinity of neighboring edges in a way determined by the data pattern. The equivalent voltage noise (n^{TX}) is the voltage noise at the main edge translated from local jitter and jitter on the neighboring edges in proportion to the slope of the channel step response at the appropriate time offset. Figure 10.17 illustrates this. The jitter components from all edges are converted to voltage noise through the corresponding signal slopes (S^S). The total equivalent voltage noise, at any point on the superposed waveform, is the sum of all the noise. Because of the additive voltage noise from random jitter, the eye diagram will be fuzzy when compared to an eye with only ISI.

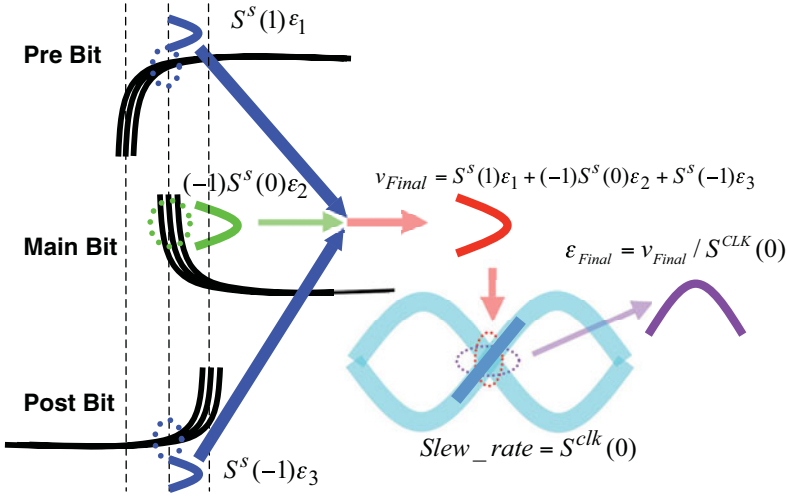


Figure 10.17 Illustration of Jitter Amplification Using Step Responses

For a clock pattern, the equivalent voltage noise at the zero crossings is the most important, because this can be translated back to total jitter by dividing by the slope at zero-crossing point. Substituting the following input clock pattern to the transmit equivalent voltage noise expression in (8.11)

$$\vec{a}^T = \pm [-1, +1, -1, ...] \tag{10.17}$$

we have

$$n^{TX} = 2\sum_{n=-M}^{N}(-1)^n h_n \varepsilon_{k-n}{}^{TX}. \tag{10.18}$$

In this expression, TX equalization is ignored for the sake of simplicity. The zero-crossing slope of the clock pattern is determined by:

$$S^{CLK} = 2\sum_{n=-M}^{N}(-1)^n\, h_n. \tag{10.19}$$

Note that for a quasi-sinusoid clock signal, the slope is largest at the zero-crossing point. The output jitter at the edge is given by:

$$\varepsilon_{OUT}^{TX} = \frac{n^{TX}}{S^{CLK}} = \frac{\sum_{n=-M}^{N}(-1)^n\, h_n\, \varepsilon_{k-n}^{TX}}{\sum_{n=-M}^{N}(-1)^n h_n} = \sum_{m=-M}^{N} h_m^{Jitter}\, \varepsilon_{k-m}^{TX} \tag{10.20a}$$

where the jitter impulse response is defined as:

$$\vec{h}^{Jitter} = [\ldots h_m^{Jitter} \ldots], \quad h_m^{Jitter} = (-1)^m\, h_m \Big/ \sum_{n=-M}^{N}(-1)^n\, h_n. \tag{10.20b}$$

The variance of the output jitter is determined as follows:

$$\mathrm{var}\left(\varepsilon_{OUT}^{TX}\right) = \left[\vec{h}^{Jitter}\right]^T R_{\varepsilon}^{TX}\, \vec{h}^{Jitter} \tag{10.21}$$

where R_{ε}^{TX} is the covariance matrix of the input transmitter jitter. The output jitter variance equation is very helpful when investigating the impact of passive channels on clock signals. It demonstrates that jitter amplification is dependent on both the jitter impulse function and the covariance matrix of the input jitter:

- When the input jitter is a very low-frequency jitter (compared to the channel response time), the output is the same as the input jitter. There is no jitter amplification. We can treat is as if it were random jitter from another receiver.
- When the passive channel is clean, the jitter transfer function is constant over the input jitter spectrum, and there is no change in transmitter jitter variance or jitter amplification. It can also be treated as another receiver random jitter.
- When the input is white, the variance of the output jitter is:

$$\mathrm{var}(\varepsilon_{OUT}^{TX}) = \mathrm{var}(\varepsilon^{TX}) \sum_{m} |\, h_m^{Jitter}\, |^2. \tag{10.22}$$

- If at least one of h_m^{Jitter} is larger than 1 in amplitude, the output variance is larger than the input variance, demonstrating jitter amplification. This happens when the slew rate at the clock edge is smaller than the slew rate at the edge of the channel step response, due to ISI. This holds true for most attenuation-dominated channels without strong reflections. If a passive channel has high reflection with proper timing, the clock edge rate can be larger than the input edge rate. In such cases, the output jitter variance can be smaller than the input jitter variable, due to jitter cancellation by correlated reflections.

To experimentally verify the jitter amplification in long PCB traces, an Agilent N4901B is used to launch 1010 pattern data, at rates up to 12Gb/s, into a four-connector, 24″ differential-microstrip PCB test structure. An Agilent 86100C DCA is used to measure both input and output jitter. The measured differential-mode insertion loss of the test structure matched reasonably well with the model in the 2GHz–6GHz range (for the clock fundamental frequency). However, because the model is for the trace only, and does not include the effect of the SMA connectors, the test structure loss is generally higher at higher frequencies. Despite this, Figure 10.18 shows reasonably good agreement between the statistical model, time-domain method from Chapter 9, and the measurements for white random jitter. Figure 10.19 shows a comparison of the jitter impulse response to a SPICE simulation. The jitter impulse response is defined as the jitter sequence due to a delta input jitter. Again, there is a good correlation.

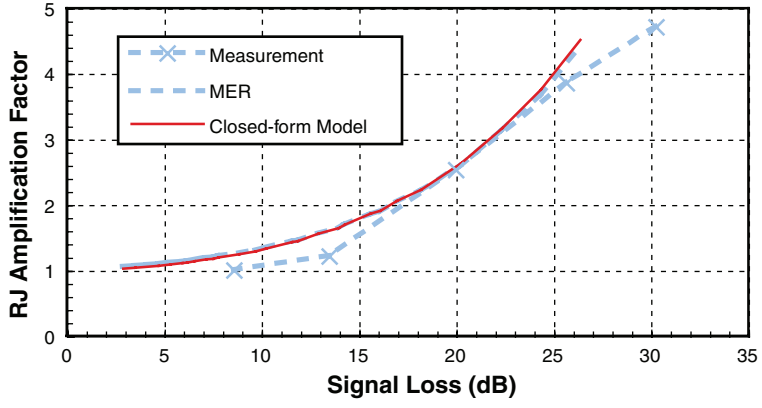


Figure 10.18 Comparison of White Gaussian Jitter Amplification with Measurement, MER Simulation, and Closed-Form Model

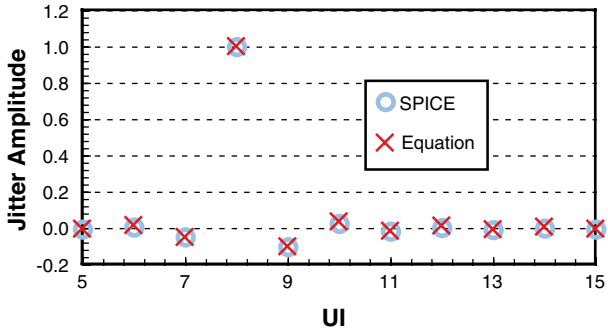


Figure 10.19 Comparison of JIF (Jitter Impulse Response) with Transient Simulation

10.5 Summary

Accurate modeling of the clock is crucial for modern high-speed link analysis, because jitter contributed by the clocking circuitry is one of the most dominant components. A brute-force method of budgeting for clocking jitter, without considering clock to data jitter tracking, often leads to pessimistic results. This chapter presents a generalized formulation that can be applied to model any clocking topology. It also shows how to derive specific formula for common serial interfaces (for example, SerDes and PCIe systems), as well as memory interfaces (for example, DDR3/GDDR5/Mobile XDR systems). Furthermore, the chapter describes the key parameters for each clocking topology and presents models for both CDR and jitter amplification due to passive channel ISI.

Finally, a few key points from this chapter:

- Clock jitter is one of the most dominant timing error components in high-speed interfaces.
- Jitter on data and clock can often be canceled or tracked out.
- The jitter cancelation is not perfect, and its effectiveness depends on the clocking architecture, data-to-clock skew, and the frequency of the jitter.
- Jitter can be amplified by passive channel ISI; however, the jitter amplification can be ignored for low-loss channels (<10dB).

References

1. D. Oh and S. Chang, "Clock jitter modeling in statistical link simulation," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Portland, Oregon, Oct. 2009, pp. 49–52.
2. D. Oh, S. Chang, and J. Ren, "Hybrid statistical link simulation technique," *IEEE Transactions on Advanced Packaging*, vol. 1, no. 5, pp. 772–783, May 2011.

3. V. Stojanovic and M. Horowitz, "Modeling and analysis of high-speed links," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2003, pp. 589–594.
4. D. Oh, F. Lambrecht, S. Chang, Q. Lin, J. Ren, C. Yuan, J. Zerbe, and V. Stojanovic, "Accurate system voltage and timing margin simulation in high-speed I/O system designs," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 4, pp. 722–730, Nov. 2008.
5. M. Steinberger, T. Westerhoff, and C. White, "Demonstration of SerDes modeling using the Algorithmic Model Interface (AMI) standard," presented at the IEC DesignCon, Santa Clara, CA, 2008.
6. J. Kim, "Design of CMOS adaptive-supply serial links," Ph.D. dissertation, Stanford University, Dec. 2002.
7. D. Oh, S. Chang, C. Madden, J.-H. Kim, R. Schmitt, M. Li, C. Yuan, F. Ware, B. Leibowitz, Y. Frans, and N. Nguyen, "Design and characterization of a 12.8GB/s low power differential memory system for mobile applications," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2009, pp. 33–36.
8. S. Chaudhuri, W. Anderson, J. Bryan, J. McCall, and S. Dabrai, "Jitter amplification characterization of passive clock channels at 6.4 and 9.6Gb/s," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 21–24.
9. C. Madden, S. Chang, D. Oh, and C. Yuan, "Jitter amplification considerations for PCB clock channel design," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2007, pp. 135–138.
10. S. Chang, D. Oh and C. Madden, "Jitter modeling in statistical link simulation," in *Proceedings of International Symposium on Electromagnetic Compatibility*, Detroit, MI, Aug. 18–22, 2008, pp. 1-4.
11. J. Ren, H. Lee, B. Leibowitz, Q. Lin, R. Ratnayake, K. Kelly, V. Stojanovic, D. Oh, J. Zerbe, N. Nguyen, "Performance comparison of data-based equalization and edge-based equalization for transmitter and receiver," presented at the IEC DesignCon, Santa Clara, CA, 2007.

Part III

Supply Noise and Jitter

This page intentionally left blank

CHAPTER 11

Overview of Power Integrity Engineering

Ralf Schmitt

High-performance electronic systems require high-quality power supply systems, in order to achieve their full potential. An ideal power supply network provides a constant voltage, at a nominal value, to all the system components. This voltage/value remains unchanged over time, regardless of temperature, activity, or interference from other system components. In simulation (for example, using the SPICE circuit simulation software), this ideal power supply is modeled as a global supply node with an ideal DC voltage source. In reality, supply voltage varies over time due to environmental changes (for example, temperature drift), changes in the circuit activity (and the resulting variations in current dissipation), and interference from other systems. Furthermore, at any given time, the supply voltage varies for different components in the system due to differences in the local environment, local current dissipation, and local current paths. These supply variations, the deviations of the supply voltage from the nominal value over time, and the different voltages for different components, have to be taken into consideration when designing an electronic system to assure functionality and performance.

Over time, the design of power distribution networks (PDNs) has become increasingly challenging. Since the early 1990s, the power dissipation for a single device has increased by close to two orders of magnitude (from a few Watts per device to more than 100 Watts today), while the supply voltage has scaled down from 5V to little more than 1V, leaving little room for supply voltage variations. An additional challenge is the broad spectrum of supply currents generated by modern electronic components. The core frequency of today's electronic components can exceed 3GHz, and high-speed interface systems are operating in the multi-Gigabit range, generating supply current components at frequencies well beyond 10GHz. Therefore, one must design the power distribution network for a frequency range of "from DC to daylight."

Beginning in the 1990s, systematic methodologies for the analysis and design of power distribution networks were developed to address these challenges. Excellent textbooks are available

that reflect the results of the last two decades of research work [1–3]. However, even with the latest advances in power-integrity engineering, providing stable power is becoming more difficult. This is because the process technology continues to improve, while the package design remains relatively unchanged. This is particularly true for I/O interface designs, because, as I/O speeds increase, stable power must be delivered at even higher frequencies, effectively making the existing package more inductive. As a result, power supply noise is one of the dominant noise sources in modern high-speed interface designs.

The primary focus of this chapter is to describe the impact of power supply noise on signal quality. It begins with an introduction to the basics of power integrity engineering. It discusses PDN design goals, and the budgeting of supply noise targets for the different components constituting the PDN. Next, it presents PDN modeling approaches, discusses the trade-off between accuracy and complexity for practical analysis steps, and demonstrates the interaction between different PDN components. Finally, it covers PDN design methodologies, and the trade-off between solutions in different components of the PDN.

Chapter 12, "SSN Modeling and Simulation," presents *simultaneous switching noise* (SSN) modeling and simulation methodology. SSN reduction, based on physical design improvements, is very challenging (or at least very expensive). On the other hand, architecture or signaling level solutions, based on bus coding techniques, are more cost effective. Chapter 13, "SSN Reduction Codes and Signaling," reviews these coding techniques. The supply noise induced jitter plays an important role in high-speed link design. Chapter 14, "Supply Noise and Jitter Characterization," discusses the modeling methodology for *power supply noise induced jitter* (PSIJ). Chapter 15, "Substrate Noise Induced Jitter," covers substrate noise modeling methodology to quantify the noise coupled through a common substrate which is a critical issue in system on chip (SOC) designs.

11.1 PDN Design Goals and Supply Budget

Power supply noise has a significant impact on the performance of electronic systems in general, and on interface systems in particular. Supply noise generates timing variations inside the interface link, which cause jitter and reduce the timing margin of the interface system. If supply noise is large, it can also generate signal distortions, reducing the voltage margin of the interface system, and eventually causing functional failures in the system. The power distribution network must provide power supplies of sufficient quality to meet the functionality and performance targets of the electronic system.

Figure 11.1 shows the range of possible supply voltages at a system component, and a sample waveform. The supply voltage varies over time around an average voltage, which can be different from the nominal supply voltage due to component tolerances in the voltage regulator module, or voltage drops due to resistance in the supply path.

Silicon devices are designed to operate over a range of variations in process, supply voltage, and temperature (PVT) parameters. Extensive simulations are required to verify that the device will function, despite any combination of PVT parameters the device might encounter

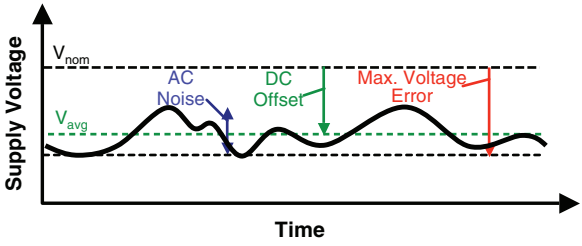


Figure 11.1 Example of Supply Voltage over Time at a System Component

during operation. The voltage range for these simulations covers the entire range of possible supply voltage values at the device, generated by both DC offset and AC noise. Keeping the supply voltage in this predefined voltage range, for all operating conditions, is one major goal of PDN design.

However, meeting the "voltage range" requirement does not necessarily guarantee system performance. Simulations over the PVT variation range verify that the circuit will function correctly for any voltage in the specified range. However, these simulations assume a constant supply voltage over time. Changes in the supply voltage over time (shown as AC supply noise in Figure 11.1) can cause delay variations in the circuit components, and distortions in the signal waveforms, which are not covered by PVT corner simulations. For example, power supply noise induced jitter (PSIJ), caused by AC supply noise, is a major source of jitter in electronic systems. PSIJ can limit the achievable clock frequency of devices, as well as the data rate of I/O interface systems. The sensitivity of system performance to AC supply noise is often a strong function of the noise frequency itself. Supply noise of fixed amplitude can have very different impacts on system performance, depending on the frequency of the noise. The PDN design must limit the margin loss to an acceptable level by controlling the amount of AC supply noise. The corresponding AC noise targets are, in general, frequency dependent, reflecting the sensitivity of the system to noise at different frequencies.

11.2 Power Supply Budget Components

The power supply budget specifies the maximum supply voltage range and AC supply noise in the system. Usually, the budget is derived from a cost/effort analysis of the system design. For example, specifying tighter supply noise targets generally simplifies the component design, because circuits have to meet their functional requirements throughout a smaller voltage range, and less timing must be reserved for supply noise induced jitter. However, at the same time, it increases the complexity of the PDN design, which now requires more resources (like capacitors, supply planes in package and PCB, and additional pads and metal layers on the silicon chip), thereby increasing the cost of the supply system implementation. Many of these resources are limited due to system design constraints like space limitations, form factor, thermal design requirements, and, especially, system cost (which is often the most important design constraint).

The power supply budget must define supply noise targets that optimize the system design by balancing the design effort and implementation cost of both the circuits and the system's power distribution network, while meeting the design constraints.

Figure 11.2 shows the elements of a typical power distribution network. It illustrates some of the challenges faced when designing a supply system. The power distribution network consists of many elements, on various levels of the design hierarchy: the voltage regulator, the PCB and its decoupling capacitors, the package, and the power distribution on the silicon chip itself. Each of these elements adds distortion to the supply system, often in different frequency ranges, as shown in Figure 11.3. The supply's quality at the end of this chain, formed by the superposition of all distortions in the PDN, is the most relevant to the performance of the interface system. The supply system design has to guarantee that the supply noise, accumulated over all the elements of the PDN, meets the design goal.

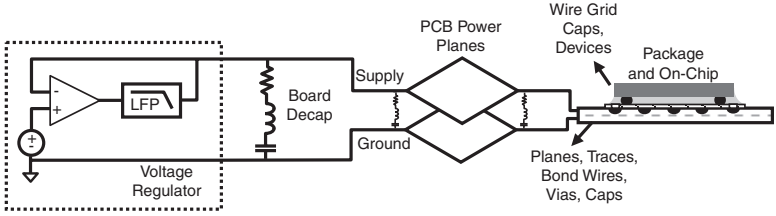


Figure 11.2 Schematic of Typical Power Distribution Network

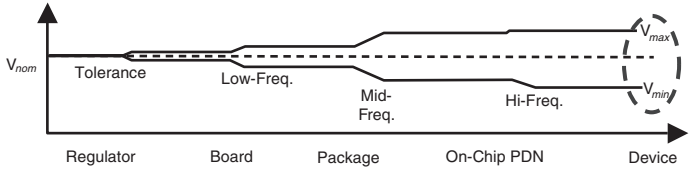


Figure 11.3 Accumulation of Supply Variations along PDN Path

A special challenge facing supply system designers is that, in most system designs, different groups or even companies design different levels of the PDN. The on-chip power distribution design usually comes from the chip design group, which is different from the package design group, which is different again (in most cases) from the system (PCB) design group. The contribution of each of the design hierarchies must be controlled, in order to meet the supply noise targets at the end of the supply path. The power supply budget not only defines the supply noise target at the end of the supply path, it also specifies the targets to be met at each level of the system hierarchy.

Consequently, the power supply budget has two purposes.

One is that it specifies the supply noise target for the devices in the system. This means that it specifies the voltage range (V_{max} and V_{min}) at the end of the PDN path (that is, the voltage range in which the system has to function), as well as the acceptable AC noise spectrum, to ensure system performance.

Another is that the power supply budget breaks down the supply noise targets into individual specifications (one for each of the various PDN components). To accomplish this, the budget balances the constraints of the different components to optimize the implementation effort and system cost of the supply network implementation.

The fact that elements of the PDN (on different design levels) often dominate supply noise at different frequencies makes the task of dividing the supply noise budget into component-level targets a little easier. Table 11.1 lists the important components in a typical PDN, and their contributions to supply noise in the system. The table demonstrates that each component mainly contributes to supply noise within a limited frequency range, and that there is only limited overlap between the frequency ranges of different components in the PDN.

Table 11.1 Contribution of PDN Elements to System-Level Power Supply Noise

PDN Element	Noise Component	Dominating Frequency Range	Notes
VRM + Bulk capacitors	VRM tolerance, load shift response	DC ... < 10 kHz	Intentional variations possible (for example, VID, AVP)
PCB + PCB capacitors	IR drop, low-freq. AC noise	DC ... < 100 MHz	Major impact usually limited to 1kHz ... 10 MHz
Package	Mid-freq. AC noise	1 MHz ... 250 MHz	Package / Chip resonance
On-chip	On-chip IR drop	DC	Location dependent
	HF AC noise	> 100 MHz	

The voltage regulator module (VRM) and the bulk capacitors on the PCB dominate the DC offset and supply noise at frequencies below 10 kHz, which is typically caused by major transitions in the power dissipation of the system. An example of a low-frequency event is a major system component's transition from "stand-by" to "active" mode, which causes a step-response by the PDN system. The nominal voltage, provided at the output of the VRM, can also show intentional variations. In a system using voltage ID (VID) codes, the nominal supply voltage is dependent on a code determined during initial device testing, and stored inside the device. The code is provided to the system during boot-up. Based on the VID code, the nominal voltage of the VRM is adjusted to meet the speed (and power dissipation) requirements of the system.

Note that a device manufactured in a slow process corner requires a higher supply voltage to achieve a fixed target speed, than a device manufactured in a faster process corner.

Another example of intentional variations in the VRM output voltage is *adaptive voltage positioning* (AVP), where the output voltage of the VRM is dependent on the current power dissipation in the system. In contrast to a VID setting, AVP changes the supply voltage over time, during the operation of the system, generating a low frequency AC offset in the process.

The printed circuit board (and the capacitors placed on it) usually dominate supply noise in the frequency range of 1 kHz to 10 MHz. The PCB can also have a minor impact on the DC offset due to IR drop. However, this impact is usually small, and, if necessary, it can be compensated for with a feedback sense line, running from the component at the end of the supply path to the VRM. Determining the optimum value or type, number, and placement of PCB capacitors is a major focus of the power-supply design methodology.

The package of a component affects supply noise in a comparably narrow frequency range, but this frequency range is often the most critical, due to an effect called "package/chip resonance." During package/chip resonance, the inductance of the package supply, together with inductance contributions from the PCB environment, create an LC resonance with the capacitance of the on-chip power distribution system. When excited by the system, this resonance can generate excessive supply noise, and must be carefully controlled during the design of a PDN. The frequency and amplitude of the package/chip resonance is dependent on the supply design at the PCB, the package, and the on-chip supply network. This is a good example of the interaction between supply design components on different design levels. Each of the contributing PDN components must be optimized to control this supply noise component.

Lastly, the on-chip power distribution contributes to supply noise primarily in two separate frequency ranges: at DC and at high frequencies (>100 MHz). On-chip wires are highly resistive due to the very thin on-chip metal layers, causing resistive voltage loss (IR drop) on the on-chip power grid. In contrast to the IR drop in the PCB and package, on-chip IR drop is location dependent; that is, some circuits close to supply pads might see no IR drop at all, whereas other circuits far away from supply pads experience the maximum possible IR drop. Consequently, compensating for on-chip IR drop by raising the nominal voltage is not possible (for example, using a feedback sense line to the VRM). Additionally, the switching activity of the on-chip circuits generates high-frequency supply noise, which is controlled using on-chip capacitors. The capacitance of the on-chip supply system also affects supply noise during package/chip resonance at medium frequencies, as pointed out previously.

This overview of supply noise components shows that, for most frequency ranges, a single PDN component dominates the supply system response. This allows defining specifications for each PDN component, with minimum overlap or dependency on the other components. An exception is the medium-frequency range, when package/chip resonance occurs. Here, several PDN components on different design levels are interfering with each other, which require a coordinated specification for all the involved components.

11.3 Deriving a Power Supply Budget

Deriving a power supply budget is an iterative process that balances the requirements of the following design aspects:

- Defining the supply noise targets that can be achieved in a system with reasonable effort and cost.
- Defining a reasonably tight supply-voltage range for circuit operation. As the supply range increases, guaranteeing correct functional operation over the entire range of the supply becomes increasingly challenging, and often more expensive in terms of area and power dissipation.
- Verifying that the margin loss due to power supply noise is acceptable, and accounted for in the margin budget of the system.

Table 11.2 shows an example of a power supply budget. It also shows the parameters, calculated during the budgeting process, used to verify that this budget is meeting the requirements listed previously. Figure 11.4 shows a flowchart of the budgeting process leading to a power supply budget, as shown in Table 11.2.

Table 11.2 Example Power Supply Budget

Component	DC Offset ($\%V_{nom}$)	AC Noise ($\%V_{nom}$)	Frequency (MHz)	Max. Noise ($\%V_{nom}$)	PSIJ Sensitivity ($\%UI / \%V_{nom}$)	PSIJ (%UI)
On-chip IR drop	0 / -4	0	DC	0 / -4	0	0
VRM/PCB, LF noise	±3	±2	<1	±5	<0.5	±1
Package/chip resonance, MF noise	0	±4	1MHz ... 500MHz	±4	<2.0	±8
HF noise	0	±4	> 500MHz	±4	0.25	±1
Total Budget	+3 / -7	±10		+13 / -17		±10

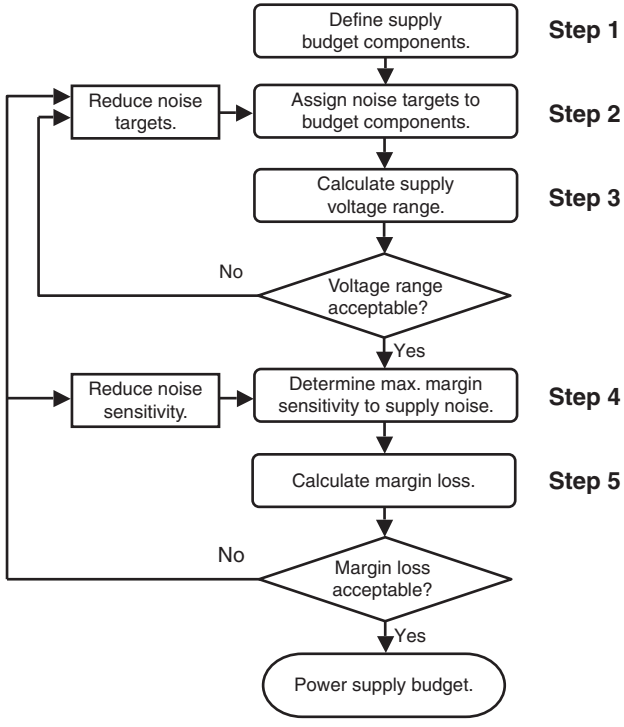


Figure 11.4 Power Supply Budgeting Flowchart

11.3.1 Dividing the Supply-Noise Range into Noise Components

The range of power supply distortions is divided into noise contributions at different frequency ranges and is controlled by different components of the power distribution network. The example shown in Table 11.2 defines the following four noise components:

- On-chip IR drops, controlled by the on-chip power grid design.
- VRM/PCB low-frequency noise, controlled by the system/PCB-level supply design.
- Package/chip resonance (medium-frequency noise) dominated by package design, high frequency PCB decoupling, and on-chip decoupling.
- High-frequency noise, controlled by on-chip decoupling and the power-grid design.

The budget components shown in Table 11.2 are the minimum number of components for a power supply budget. Defining more budget components, dividing supply noise components into more detailed, smaller components, is possible. For example, the VRM/PCB low-frequency component can be divided into the following subcomponents:

- VRM DC offset
- VRM AC noise (load response, thermal drift, and so on)
- Self-induced noise on PCB
- Noise coupling on PCB from other supply rails

Each supply-noise budget component provides a design target for the corresponding power-distribution-network component that controls that specific noise component.

11.3.2 Assigning Noise Target Values to Budget Components

Specify DC offset, AC noise, and frequency-range target values for each supply budget component. These noise target values later become the specification for the design of the power distribution network.

11.3.3 Calculating the Supply Range for Circuit Operation

Calculate the total amount of noise, and the range of supply voltages expected at the system circuits, based on the noise target values of the supply budget components. Design the circuits to meet the system specification for any supply voltage in this range.

If the circuit design cannot meet the system specification over this supply voltage range (with a reasonable amount of effort), then the target values of the supply budget components is updated. Return to Step 2, in Figure 11.4, and reduce the maximum supply noise range of the system. The supply budget process may iterate several times between Steps 2 and 3 before a noise target assignment is achieved that results in an acceptable specification for the design of the power distribution system, and an acceptable supply voltage range for the circuit design.

11.3.4 Dividing the Supply-Noise Range into Noise Components

Determine the amount of timing and voltage margin loss (also known as margin loss, for the sake of brevity) due to supply noise in this range, for each supply budget component listed in Table 11.2. In most cases, timing loss (that is, power supply induced jitter), as listed in Table 11.2, is the dominating margin loss effect due to supply noise. Other margin loss mechanisms due to supply noise are possible and are tracked in a similar way.

Margin loss due to supply noise is often a function of the noise spectrum, and depends on system design decisions (for example, clocking architecture, partitioning of circuits to different supply rails, and circuit implementation). To estimate the margin impact of each supply budget component, each component's maximum margin sensitivity to supply noise is estimated, in the frequency range of the noise component, based on simulations or on experience with similar system implementations.

11.3.5 Calculating the Margin Impact of Supply Noise

Calculate the maximum margin loss due to each supply budget component, by using the component's margin sensitivity estimations and noise target values. Then, add each component's

maximum margin loss together to obtain the total maximum margin loss. If the supply-noise induced margin loss does not meet the margin budget expectation, either reduce the supply-noise induced margin loss or update the margin budget to reflect the increased margin loss. There are two ways to reduce the supply noise induced margin loss:

- Reduce the noise targets of the supply budget components (refer to Step 2 of the budgeting process in Figure 11.4).
- Reduce the margin sensitivity by changing the system implementation (refer to Step 4 in Figure 11.4).

Several iterations may be required before a power supply budget is achieved that meets all the design requirements regarding functionality, performance, and system cost.

11.4 Supply Noise Analysis Methodology

As stated earlier, the power distribution network of a typical digital system spans three hierarchies: silicon, package, and board. The power distribution network contains various geometric structures, like planes, vias, and traces, as well as circuit elements like decoupling capacitors, voltage regulator modules, and the on-chip power distribution. The issue of power integrity has to be addressed on each design level to achieve high-quality supply voltages. Traditionally, different design teams are responsible for the various design hierarchies of an electronic system. In most cases, separate teams design the silicon circuits, the package, and the PCB, and these components communicate through a limited set of boundary conditions at the interfaces between them. Such a separation is not advisable, particularly for high-performance systems, because the supply noise in a system depends largely on the interaction between the components of the power supply network on different system levels. Addressing each component individually does not account for these interactions and can often lead to incorrect noise predictions. It also ignores the possibilities of trade-offs in the design. In many cases, power integrity is the product of multiple solutions, originating from design decisions at different design levels. Understanding these trade-offs makes finding the best solution for the system, with regard to cost and implementation effort, possible. For these reasons, power integrity requires a systemic, or co-design, approach that combines the design of power distribution components and removes the borders of traditional design-team separation.

11.4.1 Components of Supply Noise Analysis

To model the impact of supply noise on system performance in a high-speed interface, we must first predict the supply noise generated in the system, and then understand the impact that the supply noise will have on system performance. To achieve this, the PDN analysis methodology must address the following aspects of supply-noise analysis:

- Modeling the impedance of the power distribution network
- Modeling the current waveforms exciting noise in the system

- Analyze the sensitivity of circuit to supply noise
- Verify supply noise and margin loss prediction in the implemented system

11.4.1.1 Modeling the Impedance of the Power Distribution Network

The impedance (Z_{PDN}) of the power distribution network describes the sensitivity of the supply system to current changes that can cause supply noise. Z_{PDN} is the impedance between the supply and the ground rail, as seen by the on-chip circuits. Z_{PDN} is, therefore, an important Figure-of-Merit for the power delivery network of a system. Figure 11.5 illustrates a system PDN, and the ports used to measure the supply impedance Z_{PDN}.

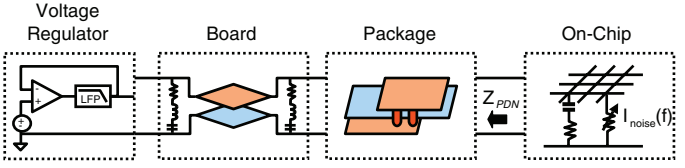


Figure 11.5 Power Supply Impedance (Z_{PDN})

Using Ohm's Law, the supply voltage $V_{circuit}\,(t)$, as seen by the on-chip circuits, is calculated as follows:

$$v_{circuit}\,(t) = V_{nom} - Z_{PDN}\,(t) * i_{circuit}\,(t) \tag{11.1}$$

where $i_{circuit}\,(t)$ is the supply current drawn by the on-chip circuits, and V_{nom} is the nominal (DC) voltage provided by the voltage regulator module.

The product of the supply impedance Z_{PDN}, and the circuit current $i_{circuit}\,(t)$, is the deviation of the voltage seen by the on-chip circuits from the ideal nominal (DC) voltage (that is, it is the supply voltage noise in the system):

$$v_{noise}\,(t) = Z_{PDN}\,(t) * i_{circuit}\,(t). \tag{11.2}$$

Because the PDN contains elements with inductive, as well as capacitive, properties, analyzing this relationship in the frequency domain is easier:

$$V_{noise}\,(f) = Z_{PDN}\,(f) \cdot I_{circuit}\,(f). \tag{11.3}$$

From equations (11.1)–(11.3), it is clear that the supply noise in a system decreases if the supply impedance decreases. Consequently, minimizing the impedance Z_{PDN} is a major goal in the design of a power distribution network.

A model of the supply impedance Z_{PDN} consists of a large number of component sub-models on different levels of the design hierarchy. Depending on the accuracy desired, some of these

components (for example, the power planes in the package and PCB and the on-chip power grid) can require a large number of model elements. The complexity of a PDN model that uses the highest accuracy of each component is usually too high to analyze the power distribution system of an entire I/O system. Therefore, the complexity of the individual component models is managed, based on the noise parameter that is simulated in a particular analysis step. For each analysis step the accuracy required for each component is adjusted, using reduced order models for those power-supply components whose higher order effects have little impact for this particular analysis.

11.4.1.2 Modeling the Noise Excitation Currents

Equation (11.3) describes the supply noise in a system in response to a circuit current excitation $I_{circuit}(f)$. In this relationship, both, Z_{PDN} and $I_{circuit}$, are functions of frequency. However, while $Z_{PDN}(f)$ is, in general, time-invariant (that is, it will not change over time during system operation), $I_{circuit}$ is dependent on the activity in the system, and can be time-variant and non-cyclo-stationary (that is, the frequency spectrum of $I_{circuit}(f)$ can vary over time as the activity—for example, the data pattern transmitted in the interface—changes).

Worst-case supply noise in the system is predicted by identifying and modeling the worst-case current excitation modes that result in the largest margin loss in the system. Figure 11.6 shows an example of $Z_{PDN}(f)$ and $I_{circuit}(f)$.

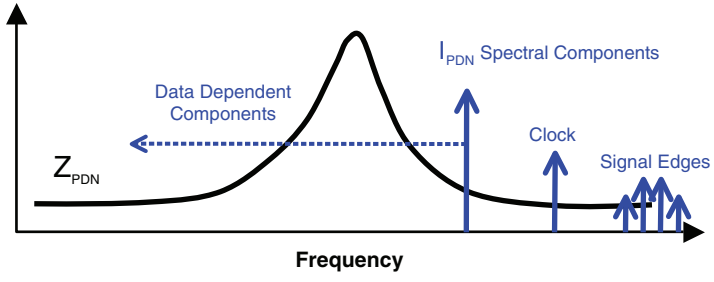


Figure 11.6 $Z_{PDN}(f)$ Profile and $I_{circuit}(f)$ Spectral Components

Expect to see worst-case supply noise amplitude when the current excitation $I_{circuit}(f)$ makes large spectral contributions in the frequency range of high supply impedance. Some of the $I_{circuit}$ frequency components are usually constant in frequency, like the clock current components, and the components associated with the edge rates of signal transitions. Other components are dependent on the activity of the system; for example, the data pattern transmitted in the interfaced system. Maximum supply noise amplitude is generated when the data-dependent components of $I_{circuit}$ excite the PDN in the frequency ranges where Z_{PDN} is large.

A general worst-case current profile for a given supply impedance (Z_{PDN}) can be calculated using the generalized peak distortion analysis described in Chapter 10, "Clock Models in Link BER Analysis," or the Reverse Pulse Technique described in by Drabkin et al. in reference [4]. However, for I/O systems, only a limited number of current profiles are possible, due to the interface signaling protocol, and a worst-case excitation profile is often constructed by hand.

11.4.1.3 Understanding Circuit Noise Sensitivity

The major design goal for the power distribution network is not to minimize supply noise itself, but to minimize the impact of supply noise on system performance and functionality. To achieve this goal, understanding the sensitivity of the system to supply noise at different frequencies is necessary. In many cases, circuits reject noise in some frequency ranges, but are sensitive to noise at other frequencies. Understanding these sensitivities makes it possible to focus on the optimization of supply noise in the sensitive ranges, define a suitable supply noise target for each operating range, and avoid over-constraining the design in regions with low sensitivity. A typical unit for the sensitivity of circuit timing to supply noise is "timing variation divided by noise amplitude [ps/mV]."

Figure 11.7 shows a typical supply noise sensitivity profile for a PLL circuit. At low and high frequencies, the sensitivity of the PLL jitter to supply noise is very low. At these frequencies, the phase-locked loop (PLL) can tolerate higher supply noise. However, the PLL shows a significant sensitivity to supply noise in the medium frequency range, so the supply noise must be held to lower levels at medium frequencies.

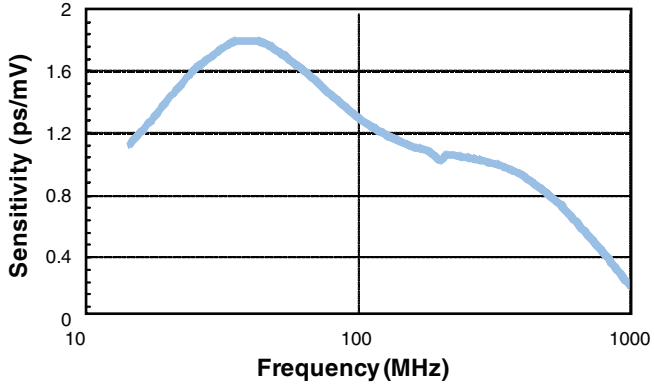


Figure 11.7 Supply-Noise Sensitivity Profile for a PLL Circuit

There are two approaches to accounting for supply-noise sensitivity when modeling the power supply induced margin loss. With the first approach, a full, detailed description of all the circuits in the system is maintained during the modeling and analysis of supply noise and noise impact. Although this is a very accurate methodology, it often results in a very complex simulation model, and is usually only feasible for the analysis of small systems. With the second approach, the supply noise sensitivity of the system is modeled separately, a sensitivity profile is

developed, and then this profile is used to identify the worst-case excitations. The supply noise for the worst-case excitations is analyzed, and the margin loss due to the resulting noise is predicted. Because this approach does not increase the complexity of the supply noise model used in the analysis, it is suitable for large systems. However, it requires a deeper insight into system behavior, and requires additional procedures to identify worst-case excitation and margin loss.

High-performance interface systems often use a combination of these two approaches. The impact of supply noise on the internal supply rails, which provide power to timing circuits like the PLL and the clock distribution, is accounted for using pre-simulated supply noise sensitivity profiles. Chapter 14 describes, in detail, how to model supply noise sensitivity. Supply noise, at the output driver supplies of an interface, is often included in the PDN simulation model, to account for the impact of supply noise during simultaneous switching noise (SSN) events. An SSN event occurs when multiple output drivers are switching at the same time, causing significant noise on their supply rails. Modeling this switching noise is quite challenging, because it involves both high-frequency signal modeling and medium frequency PDN modeling. (Chapter 12 covers this topic in-depth.)

11.4.1.4 Verifying the Supply Noise and Margin Loss Prediction

A final, important aspect of power supply analysis is correlating the supply noise and the noise impact measured in the system with the prediction of the model during the design stage. This correlation verifies the modeling methodology, as well as the implementation of the proposed solution. It also tests any assumptions made about the noise generation and sensitivity in the system during the design stage. Because each step in the analysis requires an adjustment of the component models, to reduce the total model complexity, the final correlation is necessary to confirm that the chosen methodology still maintains the required accuracy.

11.5 Steps in Power Supply Noise Analysis

The analysis of power supply noise in a system is broken down into separate steps; each step addresses different noise contributions. This concept was presented earlier in the definition of a power supply budget, and is used again here as a strategy for a PDN analysis flow.

11.5.1 DC Drop and Low-Frequency AC Noise Due to VRM, PCB, and Package

The resistance in the PCB board and package, as well as the accuracy and drift of the VRM, can cause DC offsets or low-frequency changes in the supply voltage. In some cases the supply voltage adjusts slowly during operation, reducing the voltage during high system activity, and increasing voltage during low system activity (droop control). Usually, I/O circuits are not sensitive to noise at this low frequency, as long as the minimum voltage necessary for proper operation of the circuits is maintained. Many timing-sensitive circuit blocks in high-speed I/O systems (for example, PLLs) are tracking low-frequency changes of the supply noise. Furthermore, I/O system

performance is primarily dependent on cycle-to-cycle timing distortions between consecutive clock cycles. Low-frequency supply noise only creates small distortions in this short time interval.

The VRM can cause additional AC noise during load changes in the system, due to the limited bandwidth of the regulator loop. Switching regulators can also create additional AC distortions at the regulator output, due to their switching characteristics. These noise contributions are controlled with a properly designed VRM regulator loop and PCB decoupling [5] [6], which can be easily verified later in the system. For this reason, this noise budget component is not a major issue for high-speed designs, and not covered here in detail.

11.5.2 Analyzing the On-Chip IR Drop

Due to the large resistance of on-chip wires, there is a noticeable resistive voltage drop (IR drop) in the on-chip power distribution. This voltage drop varies at different locations on the chip, and varies over time, as the activity distribution on the chip changes. Usually, package and PCB design do not affect the spatial distribution of IR drops in the chip, because the same voltage is provided to all chip bumps, because of the much lower resistance of the package and PCB traces and planes. In typical design flows, the IR drop is checked late in the design cycle, when most of the layout is available (post-layout). At this late stage in the design, only minor improvements are possible to address local IR problems.

When designing Multi-Gigabit I/O systems, analyzing the on-chip IR drop early in the design cycle, during the floor planning and bump assignment stages, is essential. The IR drop is dependent on the bump/pad placement, the power routing on the chip, and the current distribution over the chip area. Typically, signal escape routing requirements in the package restrict the placement of supply bumps/pads, and often, several different power rails are competing for bump/pad locations and routing resources. Optimized signal and supply bump/pad placement must be determined early on, together with routing guidelines for the various power rails and the floorplan of the circuit blocks on the chip to guarantee a high-quality power supply.

For IR analysis, the on-chip power distribution for each power rail is modeled as a resistive network. Static current sources are added between the power networks, which represent the maximum current dissipation averaged over one clock cycle at each location. Ideal voltage sources are placed at the positions of the power bumps/pads. Because only static currents are considered, the capacitance of the on-chip distribution can be omitted. Figure 11.8 shows the resulting model for a flip-chip design.

The spatial resolution of the power grid model is adjusted based on the spatial resolution of the current load information. In this early stage of the design, typically only the total current dissipation of the circuit blocks is known. Without detailed knowledge of the final power connections, this current is distributed homogeneously over the power routing in this region. Therefore, the large number of parallel supply wires in this region can be combined into a single “effective” wire, without loss of simulation accuracy. This reduces the complexity of the supply model drastically, allow-

ing the modeling and simulating of large areas, and of several power rails at the same time. Due to its small simulation runtime, it also allows several iterations of refinements in the bump assignment and power grid planning. As the design progresses and more detailed information about the floorplan and routing becomes available, the model can be refined, improving the spatial resolution of the model.

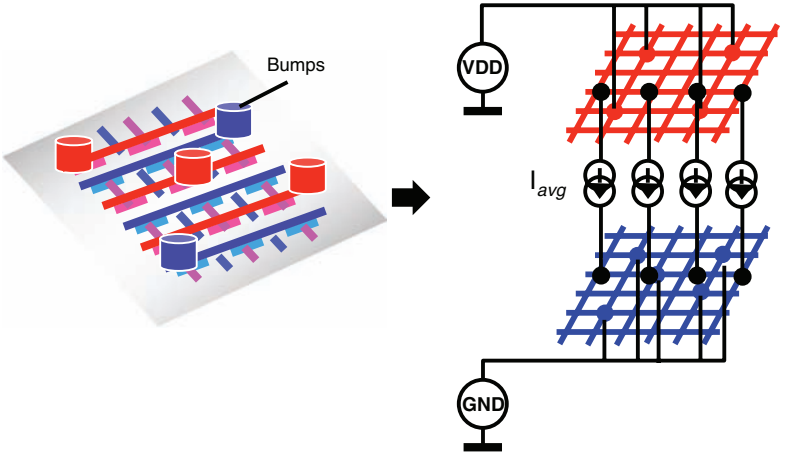


Figure 11.8 On-Chip IR Drop Model for Flip-Chip Design

Figure 11.9 shows example bump assignments, the original and optimized bump assignments. Figure 11.10 shows the IR drop map for various metal coverage settings (25%, 50%, 75%) used to reduce the maximum IR drop based on the original assignments. Due to the limited number of bumps, it fails to meet the target specification of 4%. Figure 11.10 also shows the final IR drop map after adding bumps at critical locations to meet the target specification with 50% metal coverage.

11.5.3 Analyzing the High-Frequency Switching Noise

The second supply noise component (in addition to IR drop) affecting the design of the on-chip power grid is high-frequency switching noise. Switching circuits in the I/O and core region can cause current peaks on the supply rails. Because the package typically is primarily inductive, the current provided by the package cannot change fast enough to provide the charge for this high frequency switching. Instead, on-chip decoupling capacitors are used to provide the switching current to prevent high-frequency noise on the power rails.

The inductive nature of the package impedance creates an effective low-pass filter, separating the PCB and chip at high frequencies. This filter significantly attenuates the high-frequency switching noise leaking from the chip into the PCB, where it could excite resonances in the PCB

supply planes. It also prevents high-frequency noise from any other source or device from leaking from the PCB into the chip. This filter makes analyzing high-frequency supply noise using only the on-chip supply distribution and circuits possible.

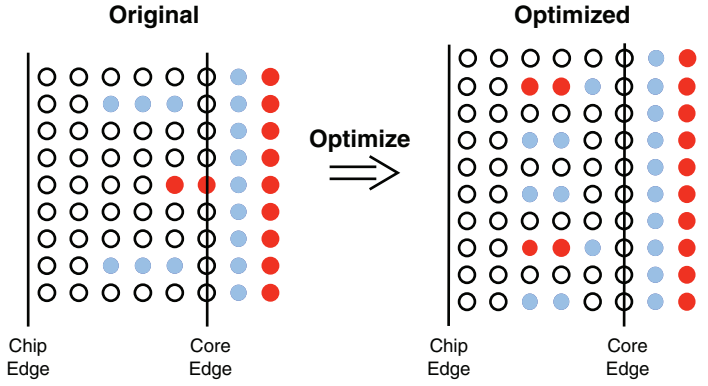


Figure 11.9 Example Bump Assignment Before and After Optimization

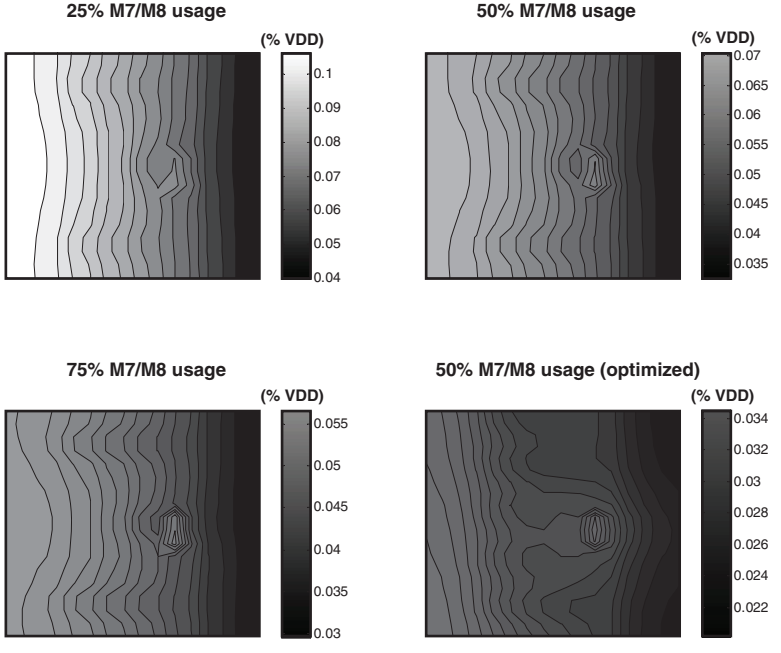


Figure 11.10 IR Drop Maps for 25%, 50%, and 75% Metal Coverage with the Original Design and 50% Coverage with the Optimized Design

The on-chip simulation model (presented earlier for static IR analysis) can be extended to simulate the effect of high-frequency switching noise to verify the amount and placement of the on-chip decoupling capacitors. For this analysis, equivalent circuits of decoupling capacitor cells are placed between the power rails, at the locations intended for decoupling capacitor placement. The static voltage sources from the IR analysis are replaced with equivalent models for the circuit blocks to create time-dependent current profiles of the circuits. These current profiles are derived from simple circuit simulations by measuring the current dissipation of a circuit block over time, as a piecewise-linear current profile. The resulting equivalent model also has to take the feedback of supply noise on the circuit itself into consideration. Fast-switching circuits create high current peaks. These current peaks can cause local collapses of the supply voltage, which in turn, slow down the circuits and reduce the height of the current peaks. This feedback effect must be reflected in the equivalent circuit models used to create the current profiles, because it noticeably reduces the amplitude of high-frequency noise in the system. Figure 11.11 shows the resulting on-chip model for high-frequency switching noise analysis.

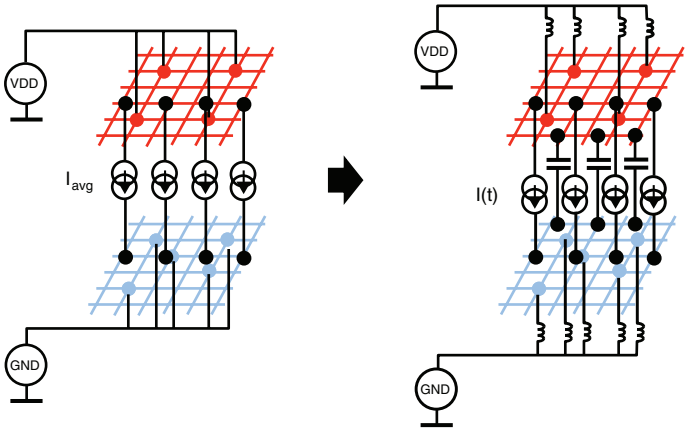


Figure 11.11 On-Chip Model for High-Frequency Switching Noise Analysis

As mentioned previously, the inductive nature of the package prevents the package and PCB supply network from reacting to high-frequency switching currents. Therefore, the package and PCB have little impact on the high-frequency noise, and their models are simplified, for this analysis, to an effective inductance connected to each pad/bump. This simple model works well, when a homogeneous high-frequency switching noise target is defined for the entire chip area. If, however, different regions of a chip are designed with different supply noise targets, then the supply noise can be coupled between these regions through the package. This case requires a more accurate, reduced-order package model that reflects the impedance between different package supply pads. A typical example is a design that separates the supply rail used for noise sensitive circuits from the main power rail to shield these sensitive circuits from switching noise. Even if

these power rails are routed separately on the chip, switching noise can be coupled to the sensitive circuits, if these power rails connect inside the package.

11.5.4 Medium-Frequency AC Noise

Medium frequency noise is often the dominating supply noise component in I/O systems. Power integrity analysis in I/O systems often focuses on simultaneous switching noise (SSN) on the supply rail of the output drivers, although this is only one contribution to this noise component. Supply noise on other supply rails at medium frequencies can also contribute to timing margin loss in the system.

Predicting the medium-frequency supply noise on the supply rails of the circuits requires the analysis of the supply network's frequency dependent impedance (Z_{PDN}), as seen by the circuits on the chip (see Figure 11.5). Additionally, it requires a model of the current changes causing the supply noise.

For medium frequency noise analysis the on-chip power distribution model can be simplified. At high frequencies, the position of a circuit block inside the power grid, and its proximity to the on-chip decoupling capacitors, determines the local supply noise for that circuit. However, at low and medium frequencies, the proximity of a circuit block to the on-chip decoupling cells has little impact, because the cycle time of the current excitation becomes much larger than the RC time constant of the on-chip power. As a result, the on-chip power grid and the decoupling capacitors in that grid can be combined into a lumped equivalent RC network. This equivalent on-chip RC network is modeled with a small number of passive elements, thereby drastically reducing the complexity of this model. The reduced-order model is derived from the frequency response of the more accurate RC on-chip model, used for high-frequency noise analysis.

The supply networks in package and PCB have to be modeled accurately for medium frequency analysis because they are the dominant components of the power delivery system. The traces and vias, as well as circuit elements in the package and PCB, are modeled using traditional equivalent models. Power planes require a more complex model to reflect the wave propagation inside plane pairs, and the two-dimensional current distribution effects at the ports of the planes. Several different modeling methodologies have been proposed [7–9]. Usually, the discussion of these models focuses primarily on the accurate modeling of plane resonances. Plane resonance itself usually occurs at very high frequencies, where package inductance and on-chip decoupling capacitance are dominating Z_{PDN}, and have little impact on system performance. These models, however, also accurately describe the two-dimensional current distribution over the ports of the power plane, which can have a significant impact on Z_{PDN}. Therefore, they are well suited to model planes in power delivery systems. Modeling planes with signal traces for SI/PI co-simulation is quite challenging. Section 12.4.3 discusses this topic in detail.

Figure 11.12 shows PDN profiles for different configurations. It compares four-and six-layer boards to observe the impact of plane inductance, and two different on-chip decoupling values are used. In this example, using a six-layer board to lower the plane inductance clearly

reduced the PDN impedance below 100MHz (package resonance). On the other hand, on-chip decoupling caps are used to reduce the PDN impedance above 100MHz.

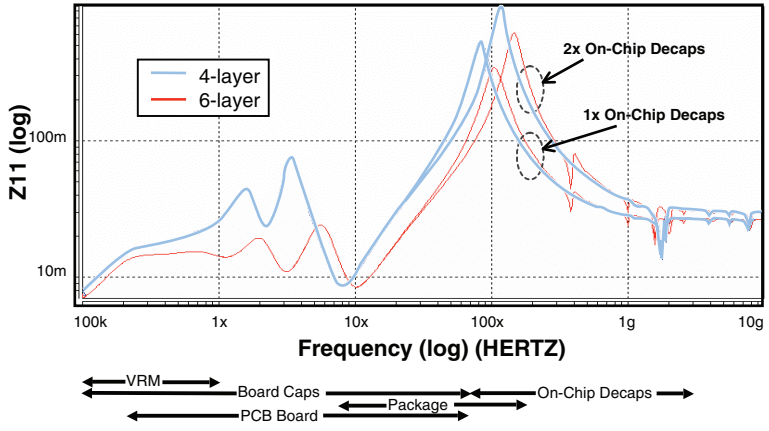


Figure 11.12 PDN Profiles for Four- and Six-Layer Boards with Different Decap Values

11.6 Summary

This chapter provides an overview of power integrity engineering and discusses the modeling methodologies and key concepts of power noise sources at different frequency contents. The low frequency noise of off-chip sources (such as VRM) is not critical for I/O interfaces, if it is low enough to be tracked by the PLL or clock. On the other hand, the on-chip DC IR drop is critical to circuit operation, and must be analyzed as early as the pre-layout stage to avoid late changes in the bump or pin assignments.

Because the package acts as a natural filter for high-frequency noise, off-chip high frequency noise on the power rails does not affect on-chip circuit operation. On the other hand, on-chip high-frequency noise is very critical to circuit operation, but it cannot be addressed by improving the off-chip decoupling capacitors, due to the package inductance. Typically, on-chip high-frequency noise has to be addressed allocating on-chip decoupling capacitors, so sufficient decap area must be budgeted for this purpose. An accurate estimation of this on-chip decap value is critical, as it can help avoid over-designing the package and system PDN.

Finally, the medium frequency near the package resonance is the most challenging to analyze, as it needs an overall PDN model that includes VRM, PCB, and on-chip PDN models. An accurate plane model, the equivalent series inductance (ESL) of the Surface Mount Capacitors (SMC), and the on-chip decap area, are all considered crucial to this analysis. Selection and placement of different SMCs is an important task, and accurate simulation can lead to a cost-effective optimized power distribution network.

References

1. A. V. Mezhiba and E. G. Friedman, *Power Distribution Networks in High Speed Integrated Circuits*, Kluwer Academic, 2004.
2. M. Swaminathan and A. E. Engin, *Power Integrity Modeling and Design for Semiconductors and Systems*, Prentice Hall, 2008.
3. I. Novak, Ed., *Power Distribution Network Design Methodologies*, IEC, 2008.
4. V. Drabkin, C. Houghton, I. Kantorovich, and M. Tsuk, "Aperiodic resonant excitation of microprocessor power distribution systems and the reverse pulse technique," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference,* Oct. 2002, pp. 175–178.
5. N. Pham, D. de Araujo, and E. Matoglu, "Design methodology for multiple domain power distribution systems," in *Proceedings of Electronic Components and Technology Conference*, Las Vegas, Jun. 2004, pp. 542–549.
6. O. Mandhana, "Optimizing the output impedance of a power delivery network for microprocessor systems," in *Proceedings of Electronic Components and Technology Conference*, Las Vegas, Jun. 2004, pp. 1975–1982.
7. N. Na, J. Choi, S. Chun, M. Swaminathan, and J. Srinavasan, "Modeling and transient simulation of planes in electronic packages," *IEEE Transactions on Advanced Packaging*, vol. 23, no. 3, pp. 340–352, 2000.
8. J. Kim and M. Swaminathan, "Modeling of irregular shaped power distribution planes using transmission matrix method," *IEEE Transactions on Advanced Packaging*, vol. 25, pp. 189–199, May 2002.
9. L. Smith, T. Roy, and R. Anderson, "Power plane SPICE models for frequency and time domains," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2001, pp. 51–54.

This page intentionally left blank

CHAPTER 12

SSN Modeling and Simulation

Dan Oh and Joong-Ho Kim

High-speed I/O interfaces commonly use differential signaling, because of its superior signal quality. However, mainstream memory interface designs still use single-ended signaling, such as *Stub-Series Terminated Logic* (SSTL) or *Pseudo Open Drain* (POD) I/O, because of its smaller pin count requirement and backward compatibility. Two of the most critical noise sources for single ended signaling are crosstalk and simultaneous switching noise (SSN). One can mitigate crosstalk by improving the physical design, such as using a flip-chip instead of wirebond, replacing microstrip with stripline, or placing ground guards. On the other hand, mitigating SSN is more challenging, because it is influenced by the physical constraints of the package design.

Figure 12.1 shows the impact of crosstalk and SSN for a typical high-end graphic channel at 5Gb/s [1]. Stripline routing is used in the controller package and motherboard to minimize crosstalk effects. Additionally, a state-of-the-art C4 DRAM package with a four-layer substrate is used to minimize power supply noise. The first eye diagram in Figure 12.1 shows a case with no crosstalk (the aggressor lines are kept quiet), and no SSN (an ideal power distribution network is used). The second diagram in Figure 12.1 shows a case with only crosstalk, and the third diagram shows a case with both crosstalk and SSN. These examples demonstrate that SSN is one of the dominant noise components in next-generation memory interface design.

Traditionally, SSN is minimized by increasing the number of power and ground pins in order to reduce the power and ground inductance. Figure 12.2 illustrates the trend towards higher pin count in graphics memory systems. The figure plots the total pin count and the ratio of power and ground pins to signal pins. As shown in this figure, the number of power and ground pins have continuously increased compared to signal pins. Although some of the new pins are attributed to increased core power requirements, the majority are needed to reduce the power supply noise for transceivers. Unfortunately, increasing power and ground pins does not reduce the over-

all inductance, after it reaches a certain level, as explained by the following formula for net inductance:

$$L_{net} = \frac{1}{\frac{1}{L_{pin}} + \frac{1}{L_{pin}} + \cdots + \frac{1}{L_{pin}}}. \tag{12.1}$$

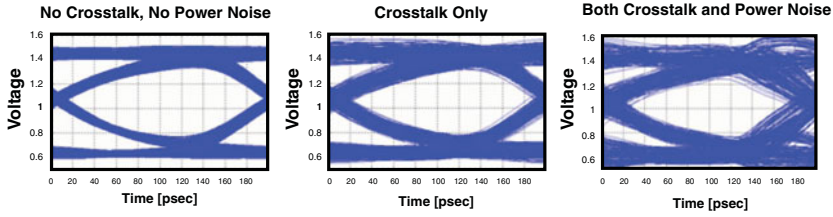


Figure 12.1 Impact of Crosstalk and Power Noise on a Graphic Memory System at 5Gb/s [1] (© 2008 IEEE)

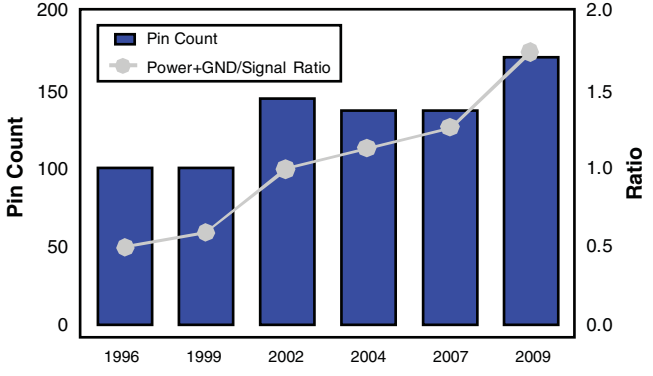


Figure 12.2 Pin Count Trends in Graphic Memory Systems

To demonstrate this fact more clearly, the net inductance is plotted against the number of pins, in Figure 12.3. In this example, the inductance of a single pin is arbitrarily assumed to be 10nH. (Real cases can be modeled by simply scaling Figure 12.3.) As shown in the figure, the number of pins needs to be doubled to reduce the inductance by half. This means that a much larger number of pins are required to reduce a constant amount of inductance. Although power demand has continuously increased due to core speed increase and the number of cores, the package design and size have not been able to catch up with the power demands. As a result, a limited

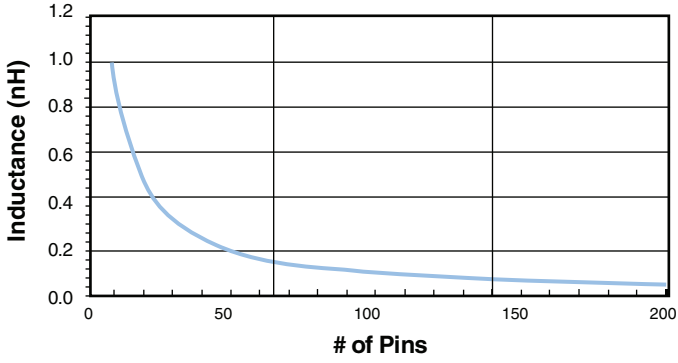


Figure 12.3 Inductance Variation vs. Pin Count

number of power and ground pins are available for an I/O interface, making SSN unavoidable in single-ended high-speed I/O designs.

This chapter discusses the challenges and methodologies used to model and simulate simultaneous switching noise. It focuses on high-level strategies and issues, rather than the details of how to generate an SSN model. Interested readers are encouraged to study other references [16] and [17]. Section 12.1 discusses the challenges of SSN modeling. Section 12.2 describes the SI and PI co-simulation flow. Analyzing SSN requires a thorough understanding of signal return loops. Section 12.3 reviews the signal loops for common single-ended signals. Section 12.4 discusses some practical tips and potential pitfalls in SSN modeling and simulation. Finally, Section 12.5 demonstrates the SSN simulation flow, using a DDR2 system with a wire-bond package.

12.1 SSN Modeling Challenges

Most single-ended signaling systems contain some degree of SSN, because the power delivery network can never be ideal, due to the physical limitation of the packages. An accurate characterization of SSN's impact on link margin is crucial when designing a cost-effective and robust I/O interface. This section pinpoints a few of the difficulties associated with modeling and simulating a channel with SSN noise.

12.1.1 Interaction between Signals

SSN is generated when multiple single-ended output drivers are switching at the same time, causing large AC current spikes on the supply rails of the device. Simulating this supply noise on power rails is not too difficult, and results have been presented in the past with good correlations to the noise measurements. However, characterizing the impact of this noise on the voltage and timing margin of the overall link is much more difficult. The channel margin depends on the interactions between different signals in the interface, such as the DQ (data) signals, the DQS (strobe) signal, and the VREF (reference) signal. Signal integrity, as well as supply noise, affects

each of these signals in various ways. Depending on the particular clocking topology, some of the noise can be tracked out. Therefore, the relative net noise is more important than the absolute noise. Consequently, the channel simulation requires that all the signals and noise sources be modeled at the same time. Figure 12.4 shows common noise sources for single-ended signaling: intersymbol interference (ISI), crosstalk, the reference voltage noise, and power and ground noise. Many of these noise sources are strongly interrelated; for instance, noise in the reference voltage is related to the power and ground noise. Additionally, their impact on signal and aggressor are also tightly correlated. Modeling and analyzing SI and PI separately cannot predict an accurate system margin.

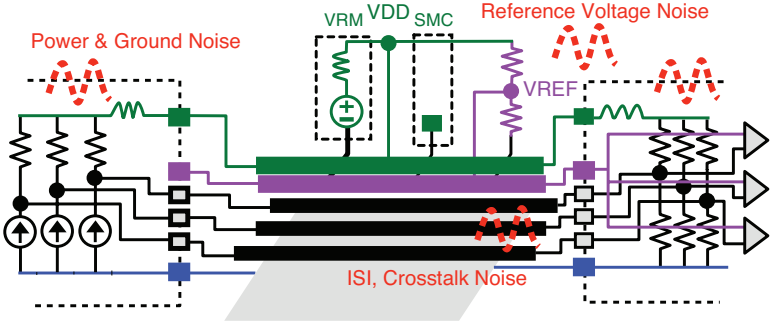


Figure 12.4 Various Noise Sources in a Single-Ended Signaling System

Typically, high-speed single-ended signaling is implemented using a pseudo differential receiver. A pseudo differential receiver uses a DC reference signal (VREF) to compare with the input signal, which makes the noise on VREF as important as the noise on the signal. Figure 12.5 shows the impact of noise on the VREF signal. In the past, VREF lines were typically designed to be quiet. A better approach is to optimize the bypassing capacitance, in order to track the noise on the power rail. Figure 12.5 shows the data strobe signal (DQS) with both a quiet VREF, and an optimally bypassed VREF, designed to track out some of the noise on the strobe signal.

The strobe signal is widely used as a clock in many memory interface systems, such as DDR1/2/3 and GDDR1/2/3/4. The main difference between a conventional clock and the strobe is that the clock signal is typically free running; that is, it always operates in a toggling pattern. On the other hand, the strobe signal only sends the toggling pattern during READ and WRITE transitions, as necessary. Because the strobe is used to sample data, ensuring that it is as free of jitter as possible is important. Figure 12.5 illustrates how a bypassing scheme reduces strobe jitter.

Figure 12.6 shows data signal (DQ) eye diagrams with both quiet and optimized VREF. The quality of the optimized VREF eye is better than the quiet case. Medium-frequency power noise is very hard to reduce due to package inductance. Hence, the best options are to either cancel it or track it out with the common noise on the reference signal. Tracking is best for low- and

medium-frequency noise as it is not location dependent due to relatively long electrical wavelength compare to high-frequency noise.

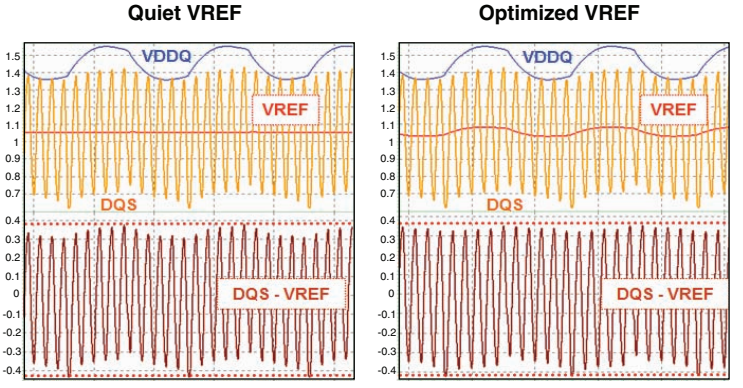


Figure 12.5 Data Strobe Signal with Quiet VREF and Optimized VREF

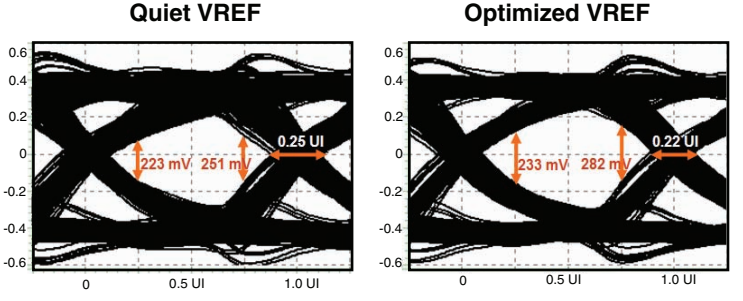


Figure 12.6 Data Signal with Quiet VREF and Optimized VREF

12.1.2 Full Circuit Driver Modeling

Traditionally, SI engineers studied I/O channel using a simple behavior model, such as IBIS, or linear voltage-controlled drivers. Because SSN is closely related to power noise and its impact on drivers, a behavior model no longer meets the accuracy requirements for SSN analysis [2] [3]. High levels of noise in the power supply or on the ground rails can cause significant changes in the driver's state. For instance, if noise grows beyond the target range, it can alter driver behavior drastically, making the drivers non-linear. As a consequence, a full circuit transistor model is desirable for SSN analysis. Unfortunately, simulating simultaneous switching noise for a wide (x16 or x32) memory interface requires many drivers, and it is not practical to use a full transistor model due to the large simulation time. Section 12.2.1 presents an efficient method for addressing this issue.

12.1.3 Complexity of Package Modeling

The dominant cause of SSN is the package parasitic inductance. This is particularly true for wirebond packages, as wires have a significantly larger inductance, as compared to the solder balls used in C4 packages. Wirebond packages are still widely used for consumer applications, such as HDTV, digital cameras, and so on. Mobile applications use package-on-package (PoP) systems, which also have significant inductance. Figure 12.7 shows an example of a PoP system. PoP systems use wirebonds to stack memory dies. The controller device can be either soldered down or wirebonded. Modeling 3D wirebonds in a PoP package is quite challenging due to its complexity. Hundreds of wires must be modeled together due to the tight coupling. Even with modern macromodeling techniques, the complexity of wirebond package models is still quite high. Section 12.2.2 discusses the reduction of wirebond model complexity.

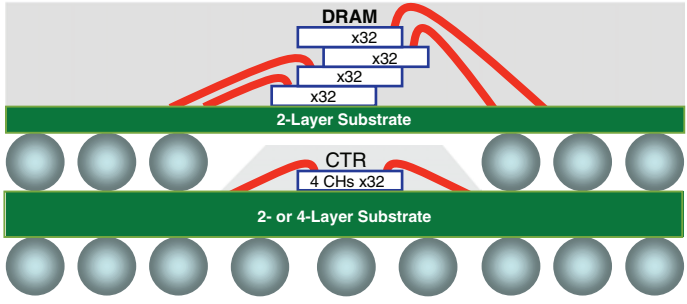


Figure 12.7 Sample Memory System in Package-on-Package (PoP)

12.1.4 Source Excitation and Simulation

Yet another difficult challenge in SSN analysis is exciting and simulating the proper data pattern. Intersymbol interference (ISI) and crosstalk are dominated by high-frequency switching activity. In contrast, power supply noise peaks at the package resonant frequency, caused by the PDN inductance and on-chip capacitance. This resonance frequency (50MHz~300MHz) is typically much lower than signal data rate. Therefore, the worst-case channel analysis requires considering both medium and high frequencies at the same time. Finding this worst-case excitation requires a good understanding of PDN and channel characteristics. This process can be even more complicated when the pre-driver is also included in analysis. As the pre-driver often uses a separate power supply, the worst-case excitations for the pre-driver and output driver are, in general, different due to different PDN resonance frequencies.

The strategy presented here is suitable for determining worst-case SSN and crosstalk. First, one of the signal lines is chosen to be the victim line. Then, the immediate neighboring signals are modeled as separate independent aggressors, in order to generate the worst-case crosstalk. The remaining signals are modeled as SSN generating aggressors, and excited using the same data pattern, which has a strong PDN resonant frequency content.

Even after the worst-case pattern has been determined, the simulation itself is still challenging, due to the long simulation time required to simultaneously analyze both the medium and high frequency effects. To avoid this large simulation time, a new simulation methodology is proposed by Kim, et al. [6–8]. This new approach captures the impact of simultaneously switching outputs at the package resonance frequency by independent simulation. The voltage margin loss obtained from this simulation is modeled as an additional input voltage requirement at the receiver (in addition to the conventional voltage requirement due to sampler sensitivity and power noise). Consequently, the effective input voltage requirement is now a function of system configurations. Figure 12.8 shows the excitation example for medium and high frequency analysis. The resulting eye diagrams for the victim data signal and strobe signal are shown in Figure 12.9. If a fast-time domain tool (such as the one described in Chapter 9, "Fast Time-Domain Channel Simulation Techniques") is available, then one can directly perform a simulation that includes both medium and high frequency data patterns.

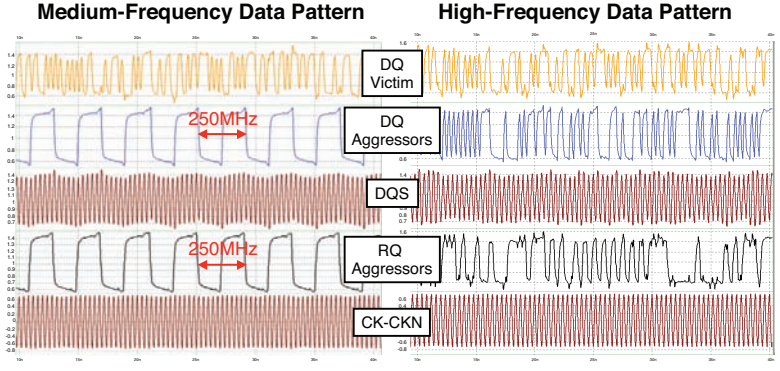


Figure 12.8 Data Patterns for Medium- and High-Frequency SSN Analysis

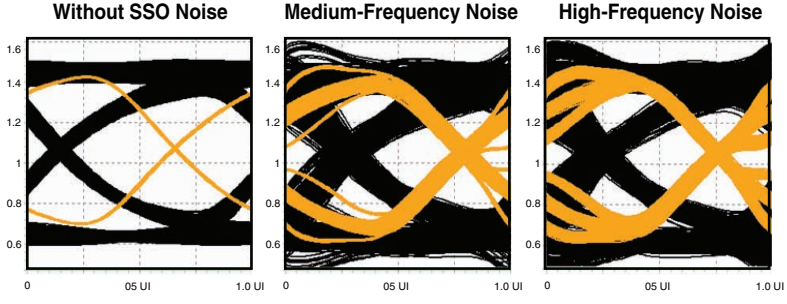


Figure 12.9 Eye Diagrams with No SSN, Medium-, and High-Frequency SSN

12.2 SI and PI Co-Simulation Methodology

As mentioned in the previous section, accurate transistor-level transmitter and receiver models are desirable for SSN simulations, because they provide accurate current profiles on the supply rails, including the potential feedback effects due to rail collapse. Typical behavior driver models do not capture complex transistor effects, such as the impact of the pre-driver and any nonlinearity of output driver [2] [3]. To analyze SSN, both the signal and power nets must be modeled at the same time with all the interface drivers and receivers. Unfortunately, analyzing an entire I/O system with transistor-level driver and receiver models requires vast computing resources.

This section describes the SI and PI co-simulation methodology proposed by Ralf, et al. [4–9]. Figure 12.10 shows the flow of the overall methodology.

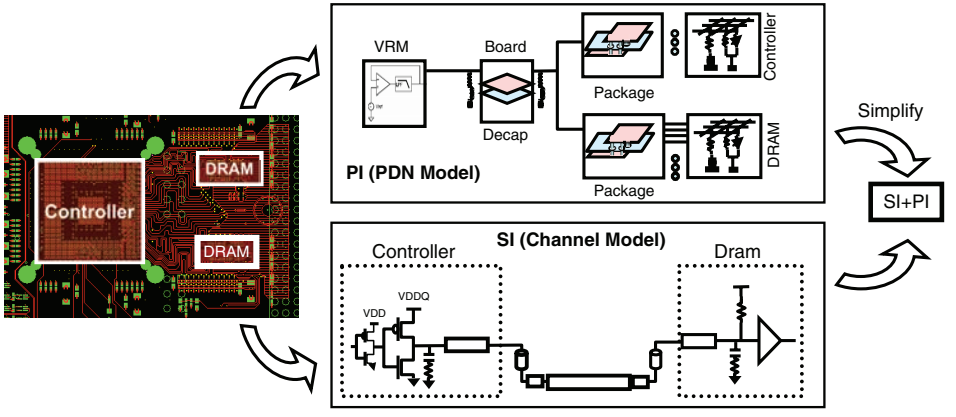


Figure 12.10 SI and PI Co-Simulation Flow

First, the PDN model for the PCB and packages is generated. A distributed RLGC model is used to model the power and ground planes, including the non-ideal ground return paths. Next, a simplified equivalent S-parameter model is calculated by eliminating the internal nodes and merging the same power nets at the external nodes. Modern SPICE simulators support S-parameter modeling and simulation, and can be used for this S-parameter model generation without the need for additional macro-modeling tools. For wirebond packages, where inductance dominates the package model, this S-parameter model can be further simplified by computing an equivalent inductance model. The equivalent inductance model can be calculated by approximating the S-parameter, using a transmission line RLGC parameter (as described in Section 5.4.1). Section 12.2.2 describes a simpler, but less rigorous, approach.

Now, an on-chip power distribution model is combined with the package and motherboard PDN model. The resulting overall PDN model is merged with a channel model for signals. For

wirebond packages, the PDN and signal models cannot be separated due to tight coupling, and so are modeled together. On the other hand, for the motherboard and on-chip trace portions, the PDN and channel models can be prepared separately. The signal traces on the PCB are typically modeled as transmission lines, using a 2D field solver. Other components, such as vias and escape lines, are characterized using a 3D EM solver. Finally, the simulation deck is completed by adding the driver and receiver models.

12.2.1 Reducing the Complexity of the Driver Model

A novel scheme to model a full circuit driver and receiver for SSN analysis is presented in references [4–9]. This scheme combines a small number of accurate transistor-level driver models with current-controlled current sources, in order to lower the complexity of the overall system model. Figure 12.11 illustrates the proposed circuit topology. The basic idea is that only one of the drivers is modeled accurately, and the remaining drivers are duplicated using current mirror drivers.

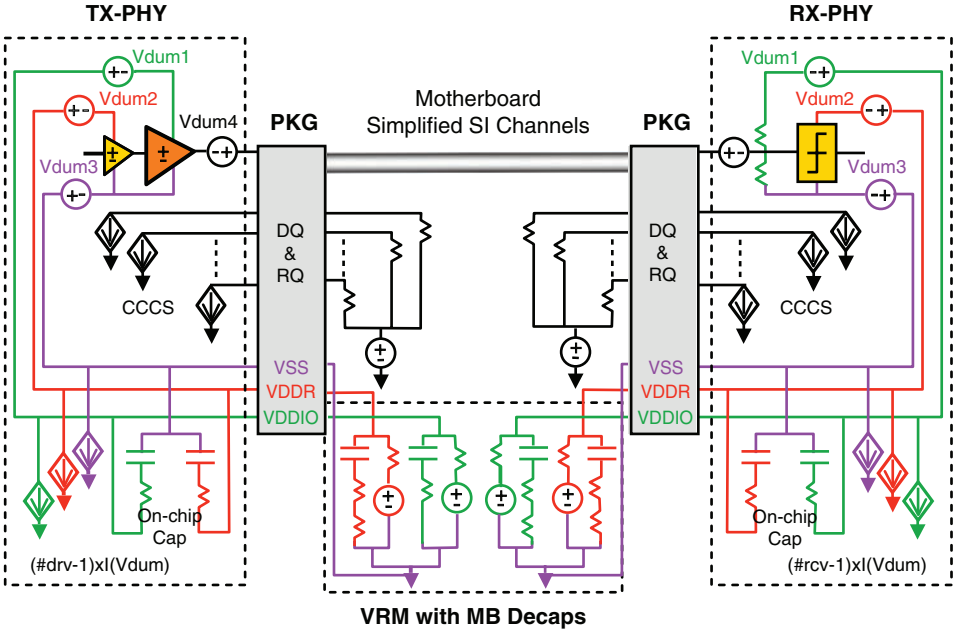


Figure 12.11 Final Channel Model Combining Both Signal and Power Integrity Models

The underlying assumption behind this scheme is that all drivers share the same data pattern, which is a reasonable assumption for SSN study. Drivers with different data patterns must

be modeled separately, using a full-circuit model. Consequently, the required number of full-transistor models is a function of the different data patterns (and power supply domains). A typical channel simulation requires the following independent drivers:

- A victim line is always required. The data pattern of the victim line consists of high frequency content, in order to maximize the ISI impact. Furthermore, the high-frequency content must repeat many times, to account for any low- and medium-frequency impact from the other aggressor lines.
- Close neighbor lines that cause significant crosstalk must be modeled independently. Crosstalk is a function of distance so the worst-case data pattern could be different for each aggressor, requiring an independent transistor model. For single-ended drivers, two neighbors from each side of the victim line are typically modeled as crosstalk aggressor. These neighbor lines contribute both crosstalk and SSN, so calculating the worst-case data pattern for these drivers can be quite challenging. If the number of aggressors for SSN is large, the impact of SSN due to these crosstalk neighbor lines can be ignored. Such cases, the data pattern can contain only a high-frequency component, in order to maximize the crosstalk impact.
- The remaining aggressor drivers can be modeled using one full circuit model, as long as they share the same data pattern and common supply voltage. In this situation, all the drivers switch at the same time to excite the worst-case SSN.
- Typical driver models consist of a pre-driver and output driver blocks and often use different power supply rails. For each full-circuit driver, both the pre-driver and the output driver must be mirrored. Some designs use separate ground rails, which must be modeled individually.

Ideal voltage sources, with 0V, are inserted at the power, ground, and signal nodes of a full-circuit driver and receiver. Then, the driver and receiver currents for individual aggressors are mirrored as shown in Figure 12.11. Note that all the mirrored currents are dumped to an ideal global ground. The package model in Figure 12.11 contains a full netlist, including signal lines and individual power and ground wires. When the package model is built using a loop quantity instead of partial elements, the ground current is not mirrored.

To reduce the modeling complexity, one can simplify the channel on the motherboard by using a simple termination for all the mirrored signal nets. This approximation assumes that there are sufficient decoupling capacitors near the package pins, so that there is negligible power coupling between the transmitter and receiver. This simple approximation also assumes that most interactions with the victim signal are covered with the crosstalk lines, and the mirrored signal lines are only used to excite SSN.

For the power and ground rails, the aggressor and receiver mirrors can be combined into one effective mirror, if it is tightly shorted at the on-chip side. Based on this assumption, one can also merge the package nodes for power and ground into one equivalent net. The following

section describes this terminal reduction technique. Note that this technique can be extended to combine even driver and receiver nodes.

Figure 12.12 compares the supply voltage waveforms (at both the controller and DRAM) from a full system with those from an equivalent mirrored system. The example system is the LPDDR2 x32b PoP system, shown in Figure 12.7. The overall comparison shows a surprisingly good correlation.

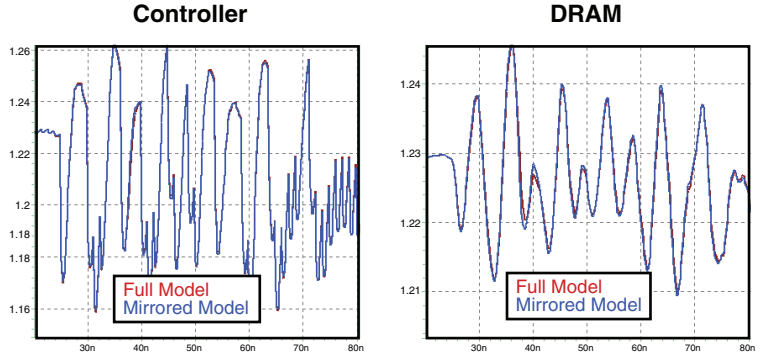


Figure 12.12 Comparison of a Full Circuit Model and Mirrored Current Model at the Controller and DRAM for LPDDR2 PoP System

12.2.2 Generating Simplified Equivalent Circuit Models

A complete package model consists of numerous conductors, wires, and planes for signal, power, and ground. One can use macro-modeling techniques to simplify the complexity of the package model by reducing the complexity of the internal circuit representation. However, the resulting model is still quite complex, due to the large number of external nodes. This section presents several options that reduce the number of external nodes for SSN analysis [9]. First, the section covers the reduction of power and ground nodes. Then, it is extended to signal lines under a simultaneous switching condition.

12.2.2.1 Merging Common Power and Ground Nodes

When power and ground nodes are well shorted (connected) at the package terminals, these terminals can be combined (as shown in Figure 12.13). One way to obtain a reduced terminal model is by running a circuit simulator with the connected package model to generate the S-parameter. As explained in later sections, the S-parameter is not suitable for representing PDN due to its low impedance nature. The dynamic part of PDN impedance is primarily located near the package resonance, which is significantly lower than the signal's Nyquist rate. Consequently, it can be effectively modeled using a few lumped elements. In addition, the lumped-element representation is more intuitive and may give insight into optimizing the physical design. This section

describes a simple, but effective, way to generate a reduced circuit model based on a distributed RLGC model.

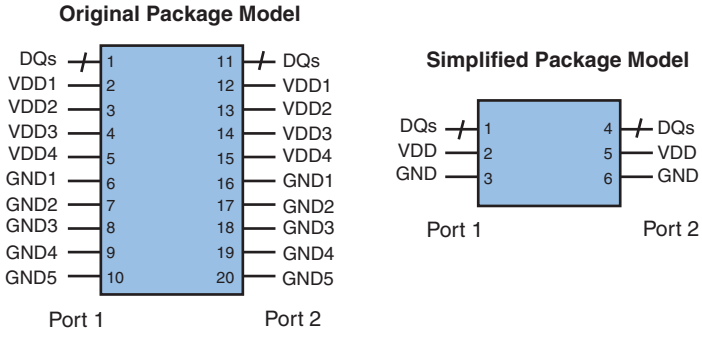


Figure 12.13 Original Package Model with Individual Power and Ground Nodes and Reduced Model after Merging Power and Grounds

Figure 12.14 shows an equivalent circuit for a single power and ground case, based on a π-network. This model uses a partial element representation. The capacitance and conductance are connected to an ideal global ground. Note that all the mutual terms between power and ground are also included in the model. (Alternatively, one could use a T-network, or multiple segment models.)

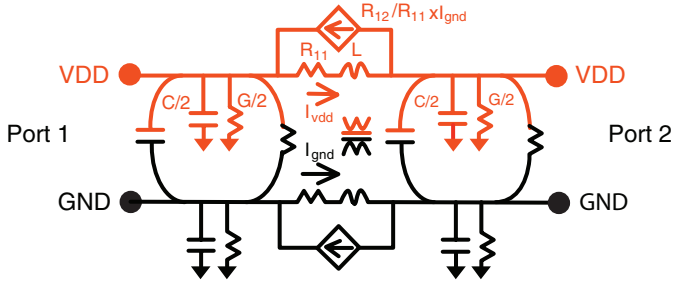


Figure 12.14 Sample Equivalent Circuit Based on RLGC Elements

To derive the lumped equivalent circuit, first consider the input impedance of a loaded transmission line (as described in Chapter 5, "Transmission Lines"):

$$Z_{in} = Z_c \frac{R_L + Z_c \tanh(\gamma l)}{Z_c + R_L \tanh(\gamma l)}. \tag{12.2}$$

For a short transmission line with an open load, the preceding equation is simplified as follows:

$$Z_{in} \cong \frac{Z_c}{\tanh(\gamma l)} \Rightarrow Y_{in} \cong G + j\omega C. \tag{12.3}$$

For a short transmission line with a short load, (12.2) becomes:

$$Z_{in} \cong Z_c \tanh(\gamma l) \Rightarrow Z_{in} \cong R + j\omega L. \tag{12.4}$$

Although the previous two equations are derived from the transmission line, one can also apply them to a power distribution network. With open termination, the admittance characteristic of the power and ground planes is dominant, rather than the series inductance, as described in (12.3). On the other hand, with short termination, the admittance characteristic of the power and ground planes is shorted out, and the impedance characteristic dominates the measurement, as described in (12.4). To summarize: the G and C values are calculated at the input terminals by opening the output terminals, whereas the R and L values are calculated at the input terminals by shorting the outputs.

Now, the simplified equivalent circuit model is constructed by combining (12.3) and (12.4). Figure 12.15 compares the original and the approximated models. The power nets and ground nets belong to the same domains are first connected together at both the input and output terminals. Then, a circuit simulation is performed using both open and short terminations, to obtain Z_{in} and Y_{in}. R, L, G, and C are estimated by line fitting Z_{in} and Y_{in}. For example, the admittance plot in Figure 12.15 illustrates the original Y-parameter, and its approximation for a typical PoP system. Both the partial power net admittance and the mutual admittance between the power and ground nets are shown. The figure shows an excellent match. The ground net comparison is omitted, because the response is virtually identical to the power net case. The impedance plot in Figure 12.15 shows the original Z-parameter and its approximation. Again, the results of the approximation perfectly match both the partial power net and mutual terms.

Because the loop impedance, looking from the driver circuit, is the most important parameter in a PDN model, the loop impedances of the original and approximate models are compared in Figure 12.16. Note that the loop impedance is much smaller than the partial impedance compared to Figure 12.15.

Finally, Figure 12.17 illustrates the S-parameter model of the package. The S21 response has a very wide frequency response, which can cause the simulation challenges described in Section 4.2. In the case of the power and ground net models, lumped circuits with a few segments are often sufficient.

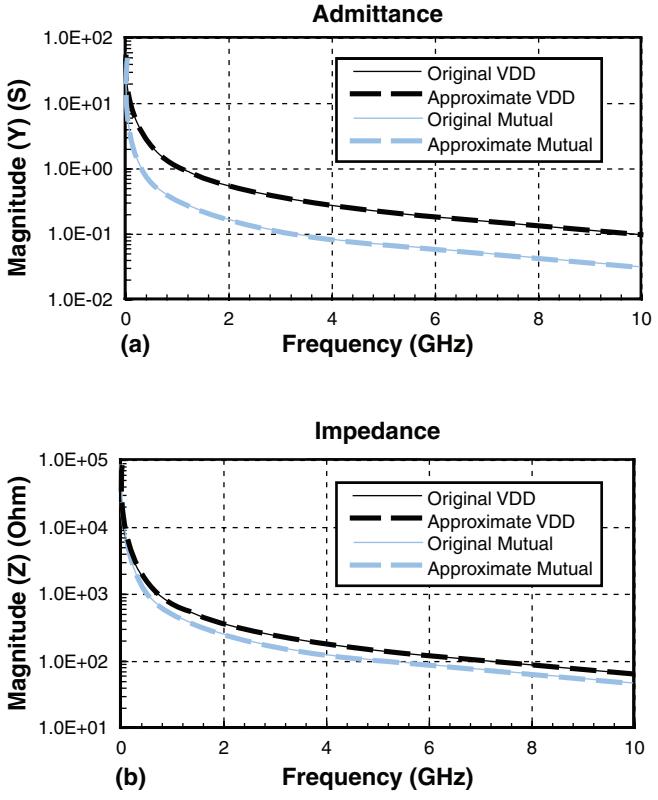


Figure 12.15 Partial Admittance and Impedance Comparison of Power and Ground Nets

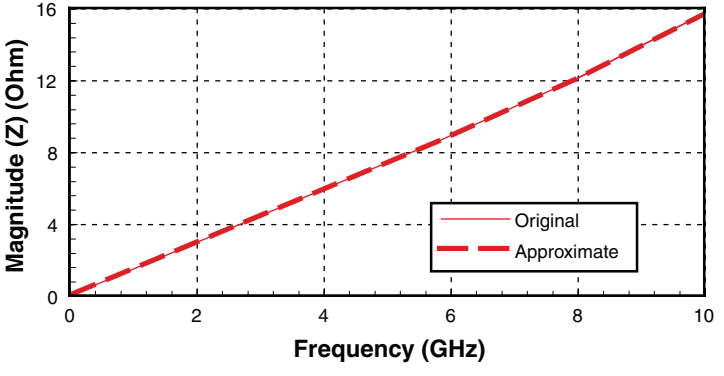


Figure 12.16 Comparison of Loop Impedance (Looking from the Driver Side)

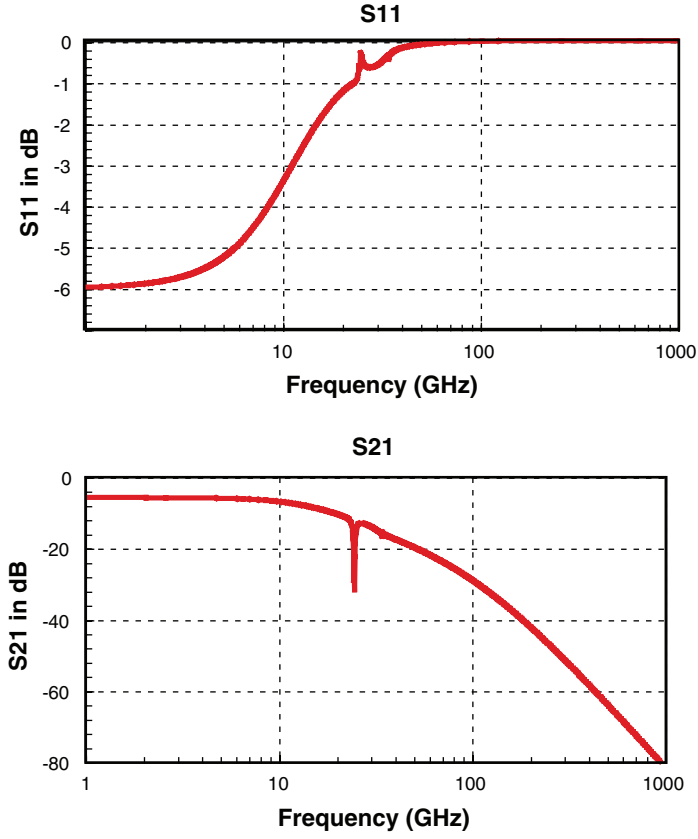


Figure 12.17 S-Parameter for Package PDN: S11 and S21

12.2.2.2 Merging Common SSN Aggressor Lines

Section 12.2.1 discusses a novel way to reduce the driver modeling complexity based on current mirroring. The method assumes simultaneous switching activity, where all aggressor lines are switching at the same time. Under this condition, one can also reduce the package model for the aggressor signals. The concept of merging equipotential signal lines is described in references [10–13]. All of these methods have been applied to crosstalk analysis. In addition to modeling the accurate aggregated crosstalk response, they also attempt to maintain an aggressor self response that is identical to that of an independent single-line case. Although these methods are effective when coupling is relatively small, they cannot be used when coupling is severe.

In this section, the approach used in the previous section is applied to the signal lines. The merged net of aggressors now becomes a *supernet*, which generates equivalent power and ground noise due to multiple aggressor lines. However, this supernet does not model accurate aggressor

self response. Because the signal quality of the supernet itself is not of interest for SSN analysis, this approach provides a simple, yet accurate, way to model the effects of aggressor SSN impact. This method, however, is not applicable to lines with strong direct coupling.

Signal nets differ from power and ground nets in several respects; these differences are as follows:

- Both the source and termination networks must be adjusted properly. For instance, when *N* lines are merged into one supernet, the amount of current driven into the supernet must be also be increased by *N*, and the source and load termination values must be reduced by *N*. Figure 12.18 illustrates a three-line case.

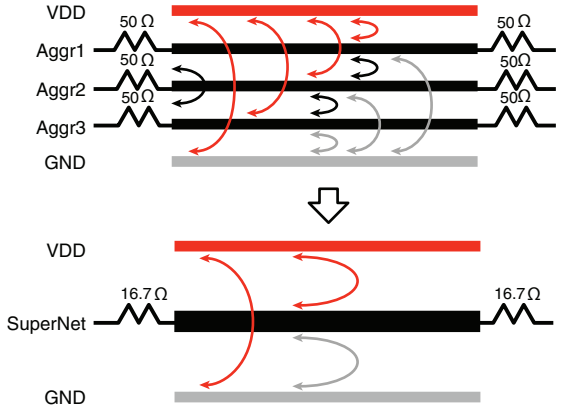


Figure 12.18 Three-Line Example of Merging Signal Lines into One Supernet (The Arcs Represent Mutual Coupling between the Two Nets)

- Due to all the changes to the source and load conditions, the waveform at the supernet has little physical meaning. Only the impacts on the victim, power, and ground nets are real.
- Because the supernet is a strong function of coupling to power and ground nets, its self-characteristics also change, based on this coupling.
- In general, two supernets from two different blocks cannot be cascaded.

Even with these limitations, the supernet concept is still useful, and can be applied to a wide variety of applications (however, keeping the preceding limitations in mind is important).

One might consider removing the mirrored signal nets, or the supernet, because all the currents in the power and ground nets are mirrored (as shown in Figure 12.11). This is feasible for packages where there is no coupling between the signal and power nets. For instance, there is an additional power plane below the ground plane associated with a microstrip line. However, in general, and especially for wirebond packages, the coupling between the signal and power nets is significant, and cannot be ignored.

To demonstrate the advantages of the supernet in model reduction, a low power DDR2 (LPDDR2) system is again considered (as shown in Figure 12.7). There are 61 signal, 14 power, and 15 ground lines to be modeled. This example uses a 216-ball JEDEC PoP package (shown in Figure 12.19). Only one-half of the package is considered to model an x32 DRAM system. Both controller and DRAM use a wirebond package. To simplify the analysis, only the wirebond portions of the packages are modeled, while the package and PCB traces are assumed ideal.

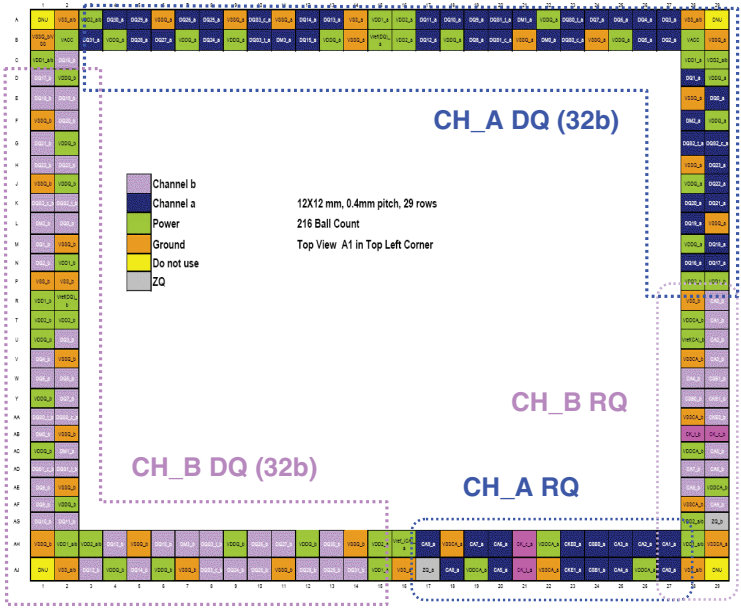


Figure 12.19 216-Ball JEDEC PoP Package Ball Assignment

Figure 12.20 compares the eye diagram from a full model to eye diagrams from a simplified model with and without supernet. As the figure illustrates, a good match exists for both cases even for the case without the supernet. Figure 12.21 shows the power noise waveforms at the controller and DRAM. Again, they all match nicely. In this package design, the PDN network is optimized so that the mutual inductance between the supernet's signal loop (or any aggressor nets) and the power current loop is almost negligible. Figure 12.22 illustrates two cases with small and large mutual inductances between the signal and power loops. In the left figure, the VDD and GND traces are tightly coupled, so that the inductive coupling from the signal loop to the power loop is almost zero. In the right figure, the power current loop is wide, and is susceptible to noise from the signal loop.

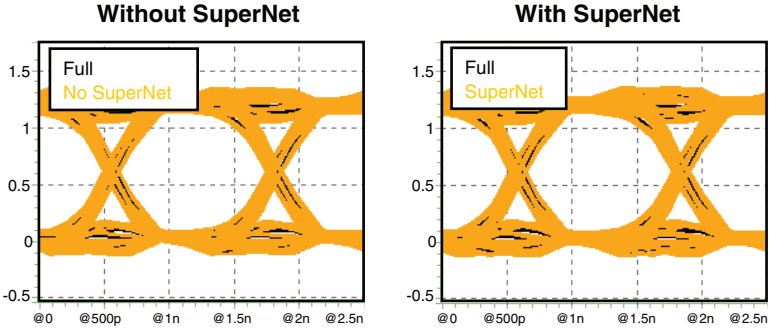


Figure 12.20 Comparison of LPDDR2 Signal Eye Diagrams, Using Full Terminal and Reduced Models, with Supernet and without Supernet

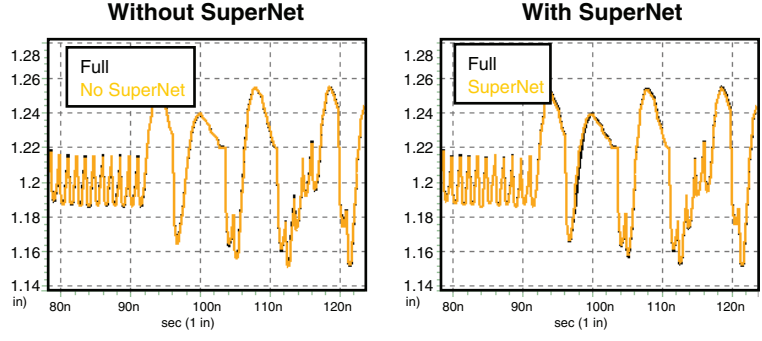


Figure 12.21 Comparison of LPDDR2 Power Noise, Using Full Terminal and Reduced Models, with Supernet and without Supernet

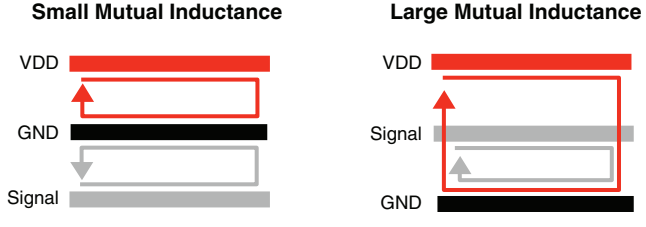


Figure 12.22 Small and Large Mutual Inductances between Signal and Power Current Loops

Next, a suboptimized design is considered, with non-zero mutual inductance between the aggressor current loop and the power current loop. Eye diagrams and noise waveforms are shown in Figures 12.23 and 12.24. As expected, without the supernet to account for the mutual terms, the eye looks significantly better than the full model case, because it underestimates the signal to

power coupling. On the other hand, the supernet example shows a good correlation with the full-model example.

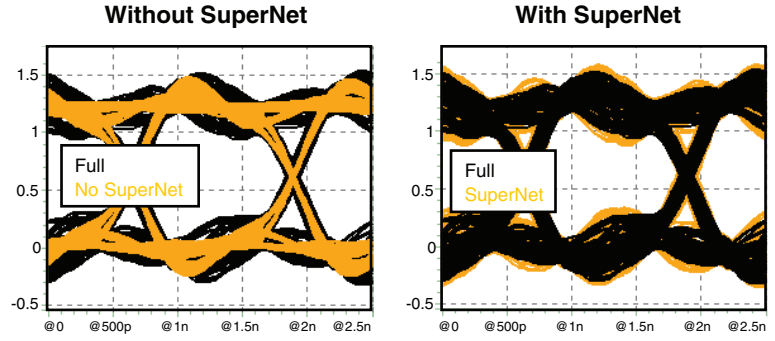


Figure 12.23 Comparison of LPDDR2 Signal Eye Diagrams for Sub-Optimal Package Design, Using Full Terminal and Reduced Models, with Supernet and without Supernet

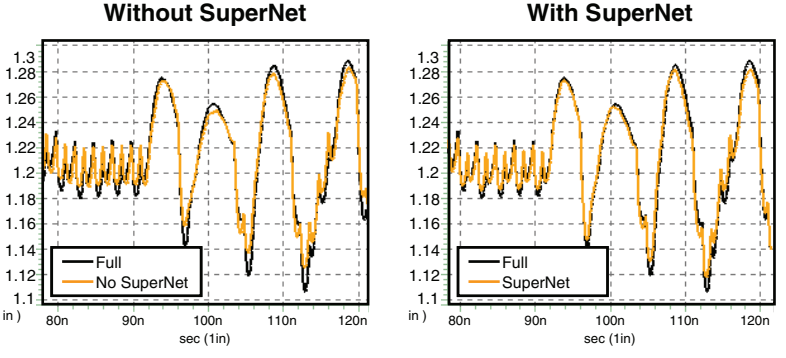


Figure 12.24 Comparison of LPDDR2 Power Noise for Sub-Optimal Package Design, Using Full Terminal and Reduced Models, with Supernet and without Supernet

12.3 Signal Current Loop and Supply Noise

Because simultaneous switching noise is mainly due to PDN inductance, the signal current loop plays an important role in SSN analysis. Simulating the correct responses and improving channel performance depends on a thorough understanding of the physics behind how supply noise couples to a signal, and how a signal generates supply noise. SSN is a strong function of the signal current loop, and the coupled supply noise is highly dependent on the signal return path. As a result, SSN is inherently a function of driver and termination schemes. To reduce SSN's impact on signals, one must consider the entire signal current path. Two common single-ended signaling schemes, SSTL and POD, are considered to illustrate this mechanism.

12.3.1 SSTL Signal Current Flow

Stub-Series Terminated Logic (SSTL) signaling uses a symmetric push-pull driver and termination, which is commonly used in main memory systems for PCs. Figure 12.25 shows the SSTL channel topology, along with the signal current loop for the high-frequency components. Due to the symmetric nature of SSTL signaling, only the pull-down operation is addressed in this section. The package self-inductance (in Figure 12.25) represents the loop inductance for the supply current loop between the power and ground nets. The mutual inductance represents the inductance between the signal current loop and the supply current loop. As Figure 12.25 illustrates, the on-chip decoupling capacitance provides a return path for the high-frequency currents. VRM and on-board surface-mount capacitors (SMC) do not provide significant current, due to package inductance.

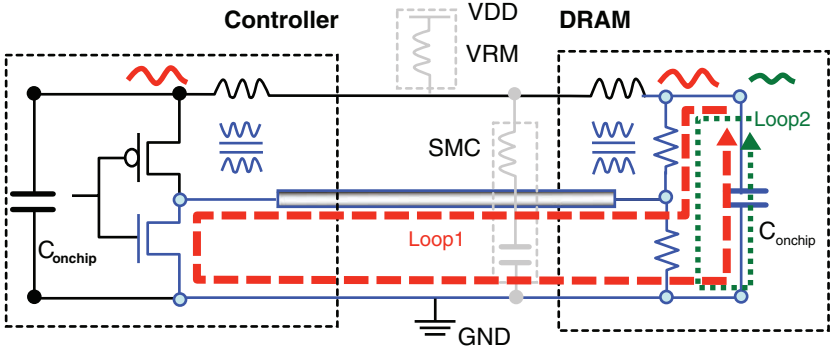


Figure 12.25 Current Loops for High-Frequency SSTL Signal During Pull-Down Operation

Two major current loops exist in SSTL signaling. Loop 2 is an internal on-chip current loop, which generates very little power noise, because the current does not go through the package inductance. Loop 1 is an external current loop, and its return current goes through both the controller and the DRAM package inductances, creating power noise. The induced power noise is small, because the majority of the power noise is generated through the mutual inductance of the package. Flip-chip packages have even less mutual inductance, and power noise for this high-frequency current is minimal. Note that Figure 12.25 uses a loop inductance representation, instead of partial inductance. The partial representation used in [6–8] could lead to spurious ground-bounce noise. (Section 12.4.2 covers this topic.)

For the low- to medium-frequency signal components, the on-chip decoupling capacitance no longer provides a low-impedance return path, and the majority of the current is provided either by an on-board voltage regulator module (VRM) or a surface mounted capacitor (SMC); see Figure 12.26. Now, both Loops 1 and 2 are external current loops, and cause power noise on the DRAM side. On the controller side, only Loop 1 causes power noise. Because the majority of the

current is through the DRAM package power rail, the induced supply noise is large on the DRAM side for both Loops 1 and 2. On the controller side, Loop 1 generates the supply noise through the mutual inductance, so the induced supply noise is relatively small.

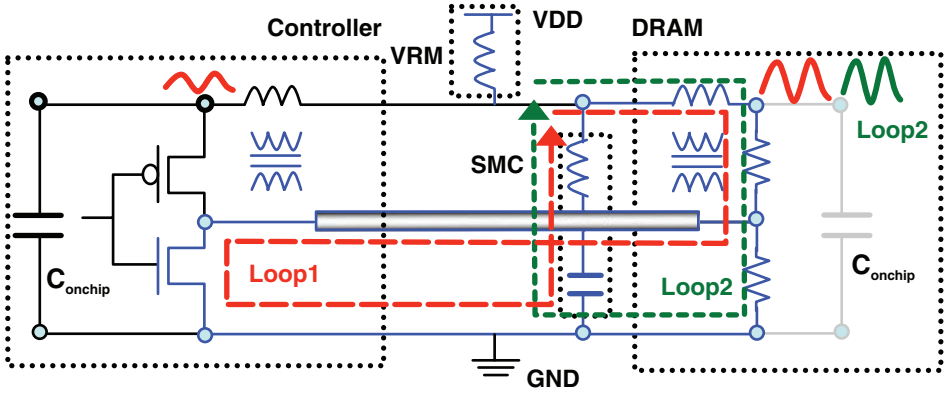


Figure 12.26 Current Loops for Medium-Frequency SSTL Signal During Pull-Down Operation

In practice, the driver-side power noise is often larger than the receiver side, due to the supply noise generated by the crowbar current. During the transition from the pull-down operation to the pull-up operation (or vice versa), there is a significant time where both pull-up and pull-down drivers are active, which causes crowbar current to flow directly from power to ground. This crowbar current is high frequency and dynamic in nature, but can lead to medium- or low-frequency supply noise, if its high-frequency content is modulated by low-frequency events; for instance, when the high-frequency toggling data pattern is sent in burst mode.

12.3.2 POD Signal Current Flow

Pseudo Open Drain (POD) signaling also uses a push-pull driver, but it only uses pull-up termination. POD is commonly applied to high-performance graphic systems and is being considered for next-generation main memory systems. Because POD signaling uses only a pull-up termination, it behaves similarly to open-drain signaling, which uses a pull-up termination with only a pull-down driver. This significantly reduces power consumption by eliminating current flow during the high state. By employing suitable data coding to minimize the low state, using POD can save significant power (see Chapter 13, "SSN Reduction Codes and Signaling"). Unlike the open-drain example, POD signaling uses a pull-up driver, which is typically weaker than the pull-down driver, to aid in the pull-up process.

Figure 12.27 illustrates POD signaling with the high-frequency current loop during a pull-down operation. There is only one major current loop in this case, so all the current goes through the package inductance. The noise mechanism is similar to the Loop 1 example for SSTL signaling. The analysis of low- and medium-frequency signal components is straightforward, and so it

is omitted. The pull-up operation is a more interesting case, because it provides two current loops in the high-frequency example (see Figure 12.28). The high-frequency loop can be decomposed into two loops: one at the controller, and the other at the DRAM. Compared to the pull-down example, noise generation and susceptibility are lower, because the opposite current directions cancel the two loops. However, there is no cancellation of the low- or medium-frequency signal components, because Loop 1 does not exist.

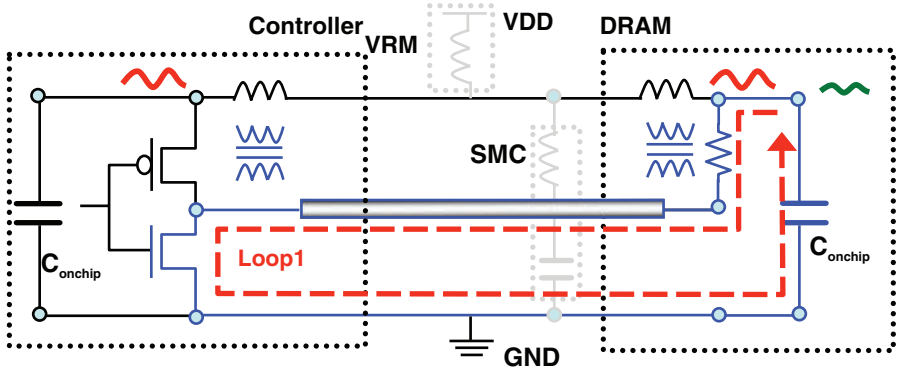


Figure 12.27 Current Loop for High-Frequency POD Signal During Pull-Down Operation

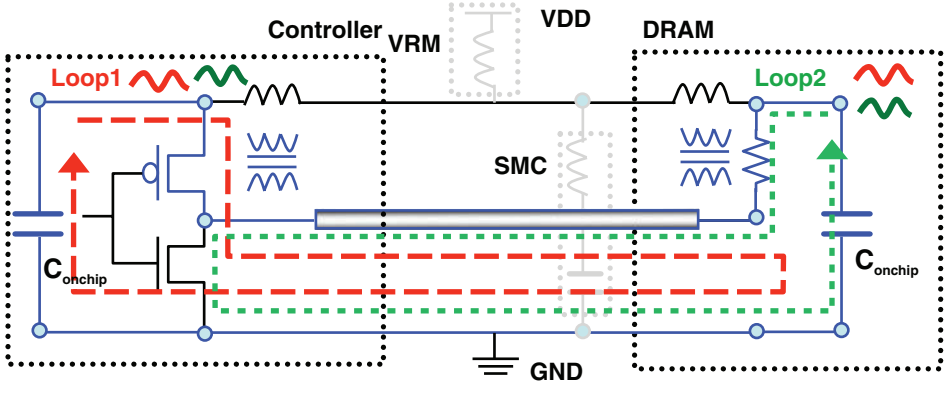


Figure 12.28 Current Loops for High-Frequency POD Signal During Pull-Up Operation

The previous two examples only considered static termination. Realistically, the parasitic capacitances, for both the output driver and input receiver, must be considered during a switching event. Additionally, the crowbar current, caused by the overlap period when both the pull-down and pull-up transistors are active, is also significant. Often, the crowbar current generates a large

SSN component at the driver, because it provides a direct current path between power and ground.

12.4 Additional SSN Modeling Topics

The previous section reviewed a general methodology to prepare and simulate an SSN analysis model. Detail modeling procedures are quite complex and they are beyond the scope of this book. This section discusses a few selected topics and issues in modeling and simulation that are important but not covered in other publications.

12.4.1 On-Chip PDN Modeling

Typical SSN analysis uses a lumped circuit to represent the on-chip PDN network. Although SSN analysis often deals with low- or medium-frequency data patterns, even these low-frequency data patterns still have significant high-frequency components, due to fast edge transitions. With such fast edge rates, a lumped PDN model may cause an erroneous high-frequency response.

This point is illustrated using a high-speed graphic memory interface. First, the on-chip PDN model is divided into partitions, as shown in Figure 12.29. The on-chip resistances for the controller and DRAM are 20.6 mΩ and 46 mΩ respectively. For the on-chip power distribution, 50% metal coverage is used for the power and ground rails. Finally, a data rate of 10 Gb/s, with a 30ps rising edge, is used in the simulation.

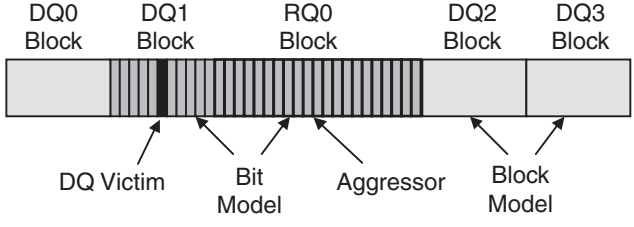


Figure 12.29 Allocation of On-Chip PDN Model and Victim Signal Locations for DQ (Data Line) and RQ (Address/Command Line)

The simulation is performed using both distributed and lumped versions of the PDN model. The first noise plot in Figure 12.30 provides an example of a single aggressor. The aggressor is located in the middle of the RQ block, and the waveform is observed at the victim's driver. The slight difference in the DC levels is due to the difference in the on-chip IR drop, between the two models. As the figure illustrates, the lumped PDN model produces significant high frequency undershooting.

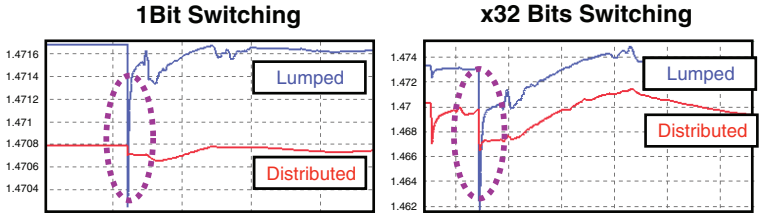


Figure 12.30 Comparison of Lumped and Distributed PDN Models for Single Aggressor and 32 Aggressors

The second noise plot in Figure 12.30 shows an example where all 32 aggressors switch at the same time. As with the single aggressor example, the high frequency undershooting is significant in the lumped model. Note that both examples captured accurate medium frequency response. Figure 12.31 shows the coupled noise due to different aggressor locations. The figure demonstrates that noise from the more distant aggressors is significantly attenuated. This attenuation could not be captured using the lumped version of the PDN model.

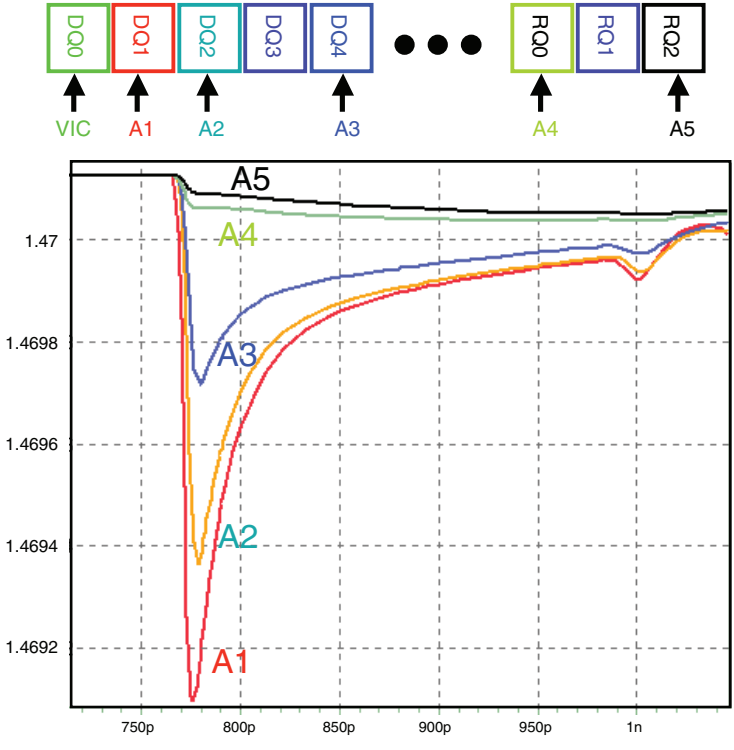


Figure 12.31 SSN Noise Due to Different Aggressors

12.4.2 Partial Inductance Model

Because inductance is defined only for a current loop, physical inductance is always a loop quantity. Although loop inductance always results in accurate simulation data, it is rather inconvenient, because the current loop must be defined a priori. For instance, with a component-level model, the return path may not be known until all the components are assembled. This is particularly true for wirebond, via, or connector models, where there is no solid return plane. A partial inductance is often used in such cases.

Partial inductance is a mathematical quantity that can be quite useful in SSN modeling. A package model, based on the partial elements, can be used directly in a circuit simulator without converting to the loop quantity. A circuit simulator automatically handles the return path. However, partial inductance can be badly misused if one does not pay close attention to details or underlying assumptions used in partial inductance:

- The partial inductance representation of a current loop consists of at least two partial self-inductance elements and one partial mutual-inductance element. These elements must be dealt as a group at all times. One cannot adjust one element without changing the other elements.
- The partial mutual inductances are often very large when compared to the mutual loop inductance, and they decay very slowly over distance compared to the loop case.
- Only the voltage difference between two nodes of the partial element is real, and the two nodes must be at the same side (see Figure 12.32).

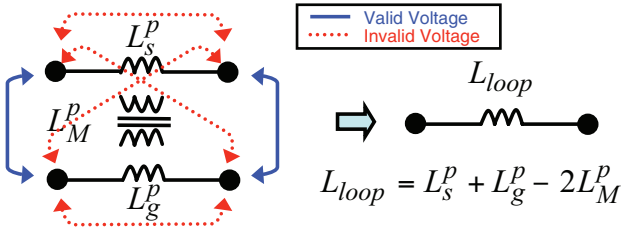


Figure 12.32 Correct Voltage Measurements for Partial Inductance and Corresponding Loop Inductance

- When a new partial inductance is added to an existing model, one must correctly add a self term and mutual terms to all the other partial inductances. Incorrectly adding or combining the two models results in artificial signal or power noise [14].
- All currents must be returned through the partial inductance elements. This implies that global ground nodes cannot be located at the two sides of partial elements.

The most common mistake when using a partial inductance is observing the node voltage without measuring relative to the corresponding return node. Often, the voltage of the on-chip ground is mistakenly measured globally, or relative to an on-board node. Such measurements

show a spurious voltage noise, which is referred to as *ground bounce*. *Ground bounce is purely artificial; do not treat it as real noise* [15]. (The same is true for observing the power node without a proper reference node.)

Figure 12.33 illustrates this mistake. The package model is generated using partial and loop elements. The victim line is observed globally. The partial element case has spurious glitches, and these same glitches are shown at the ground. On the other hand, the loop element model shows no glitches at the signal. After subtracting the victim line voltage from local ground, the net voltage shows a clean response (see the bottom of the figure).

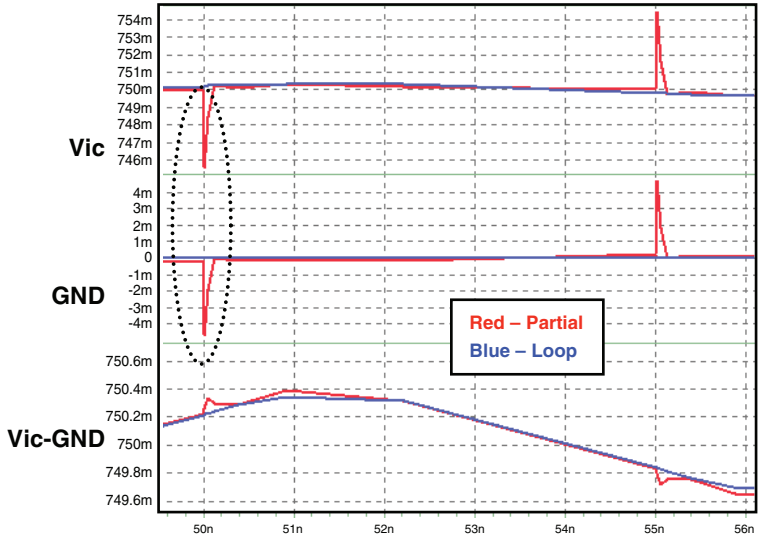


Figure 12.33 Artificial Voltage Glitches Caused by Partial Element Model

The LPDDR2 PoP system is considered again to observe the difference in voltage waveforms using the global and local references. The resulting eye diagrams are shown in Figure 12.34. Again, the globally referenced signal shows significant noise due to artificial noise associated with the partial element, whereas the locally referenced signal has much smaller noise.

12.4.3 Co-Simulation of Signal Net and Power Planes

During channel modeling for signal analysis, planes are often treated as ideal grounds with equipotential. This assumption is valid under the condition that there is sufficient decoupling capacitance, or stitching vias, between planes. For co-simulation of signal and power noise, the channel model (based on the ideal plane assumption) is combined with the power plane model. Figure 12.35 illustrates this modeling approach with a microstrip. However, generalizing this approach with a stripline case, where the signal trace is sandwiched between the power and ground planes, is difficult. This is because the stripline model assumes that the two planes are equipotential, and no separate current is modeled for the power plane. The brute-force method of

modeling the power plane as an additional signal node is numerically unstable. A new model based on modal decomposition is proposed by Engin, et al. [16] [17]. This section only provides a brief description of the model, and interested readers are encouraged to review references [16] and [17].

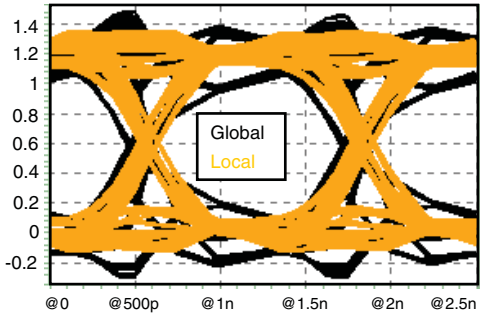


Figure 12.34 Artificial Voltage Illustration Using Signal Eye Diagrams

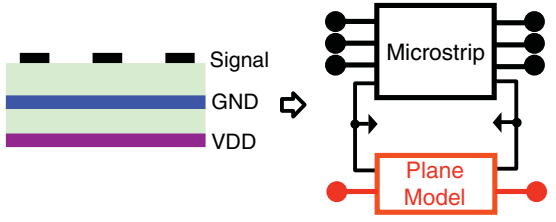


Figure 12.35 Signal and Power Co-Simulation Model for a Microstrip

Figure 12.36 shows a general model that can be used for various transmission line types, such as microstrip line, stripline, and co-planar waveguide. The coupling between the plane propagation mode and the signal transmission line mode is controlled by using an extra current-controlled current source and voltage-controlled voltage source. The model captures this coupling mechanism using a single factor k. The value of k depends on the different plane options. For a microstrip above a ground, or power plane (assuming a signal is still referenced to ground), $k = 0$ and -1, respectively. For a stripline in a homogeneous medium, $k = -h_1/h_1 + h_2$ where h_1 and h_2 are the heights of the strip and power plane from the ground plane, respectively. For an inhomogeneous stripline case, additional capacitive loading must be included to compensate for the conversion between the plane and transmission line modes. However, in most cases, the dielectric constants for the different layers are not so different, and the mode conversion can be ignored. For the coplanar lines, the coupling factor values need to be computed using a field

solver. A quasi-static solver can be used to compute the ratio of currents on the signal traces and power plane [18] [19].

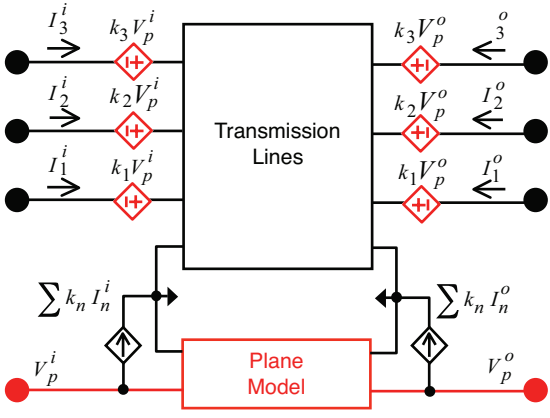


Figure 12.36 General Signal and Power Co-Simulation Model

12.5 Case Study: DDR2 SSN Analysis for Consumer Applications

This section discusses the SSN analysis of a DDR2 memory system, and addresses the issues of both self-induced and coupled noise. DDR2 memories are widely used in consumer applications, as well as in PC main memory. Designing memory systems for consumer applications is particularly challenging, because the system must be designed using low-cost technologies, such as two-layer boards and wirebond packages. The data rate of a DDR2 system is currently at 667MHz or 800MHz. Although these data rates are significantly lower than the rates of other high-speed links, SSN can still significantly reduce the timing margin in these systems due to single-ended signaling and low-cost physical designs. Many DDR2 memory interfaces lack on-chip bypass capacitance, as the result of limited silicon area, and use a highly inductive wirebond package.

Compared to a flip-chip design with C4s, wirebond packages typically add a substantial amount of inductance to the power distribution network (PDN). This results in large impedance of the power distribution system, at around 50MHz–300MHz, causing a resonance peak. Typical I/O drivers draw significant current over the supply rails, and this generates a large amount of supply noise. To reduce the impact of this supply noise, noise-sensitive circuits (for example, DLLs and PLLs) can use their own dedicated power rails. However, this requires additional package pins. Alternatively, power can be shared at the board level, if sufficient bypass caps are installed. However, noise can still be generated on these rails through wirebond coupling. Placing an inductive filter (such as ferrite beads) helps to reduce noise coupling through wirebond.

The system under study is comprised of a test board with two x16 DDR2 DRAMs, running at a 667-Mbps data rate (see Figure 12.37). The address and data net topologies appear in Figure 12.38. The two DRAMs share the address lines, and they are on-board terminated. The data lines are point-to-point signaling, and on-die terminated (ODT). The DDR2 interface consists of 32 data signals, operating at a data rate of 667 MHz, and 21 address and control signals, operating at half the data signal rate. The system is implemented on a six-layer PCB board. The controller uses a wire-bond package.

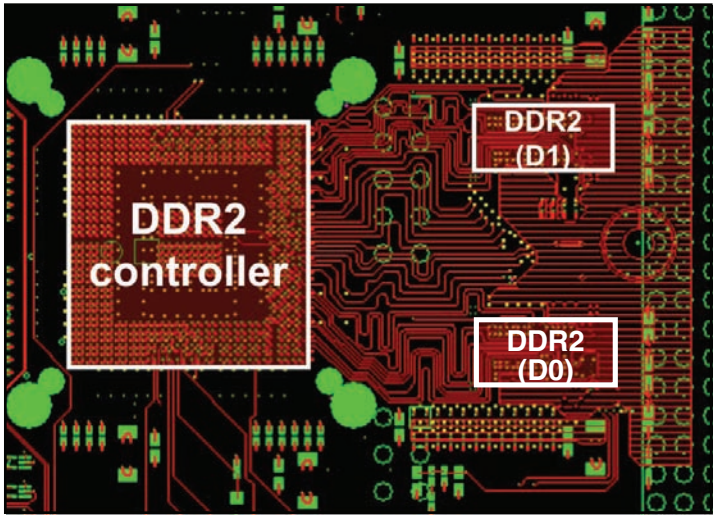


Figure 12.37 DDR2 Test System Board

The interface in the controller chip uses power supply rails that are separated from the power supply of the ASIC core. All power supply rails use individual voltage regulators to prevent noise coupling on the PCB board. The DDR2 controller interface uses a 1.8-V supply rail (VddIO) for the output drivers and a 1.2-V supply rail (Vddr) for the other circuits in the interface. Both supply rails share a common ground node (VSS).

Wirebond packages add a substantial amount of inductance to the supply network. In many cases, the impedance of the supply loop is dominated by the inductance of the wirebonds in the package. Furthermore, coupling between wirebonds is a major source of signal-to-supply and supply-to-supply coupling in the system. Therefore, a 3-D field solver is used to extract a package model from a three-dimensional picture of the entire wirebond section of the package, based on the partial inductance concept. This model preserves the inductive coupling between all bond wires.

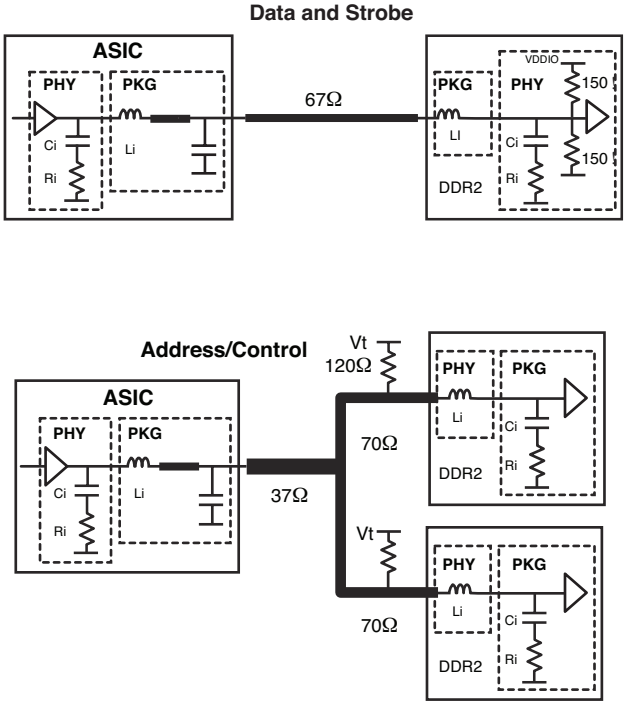


Figure 12.38 Address- and Data-Line Topologies

12.5.1 Self-Generated Power Noise and Worst-Case Switching Pattern

An impedance profile $Z_{PDN}(f)$ can be used to find the worst-case data pattern. A DDR2 push-pull driver has two return current paths, depending on the driver's input status. Figure 12.25 shows the current path during the pull-down operation. A pull-up operation creates a different current loop through the power traces and wires. The crowbar current, which flows between the pull-up and pull-down transistors during overlapping transition times, creates an additional current loop. The crowbar current path goes through the power and ground traces. Each of these currents is associated with its own PDN profiles, and all three current loops contribute to the total supply noise in the system. To identify the worst-case activity pattern in the system, one must analyze all three impedances.

Figure 12.39 shows the impedance profiles of $Z_{up}(f)$, $Z_{down}(f)$, and $Z_{PDN}(f)$ for the test system. As shown in the figure, all three of the supply impedances have peak resonance at approximately 150MHz. For data signals, an 1100... pattern (at 667MHz) is used to excite the current near the resonance frequency, which generates 167-MHz noise. Because the address lines operate at half the data rate, 0101... is used for the worst-case pattern.

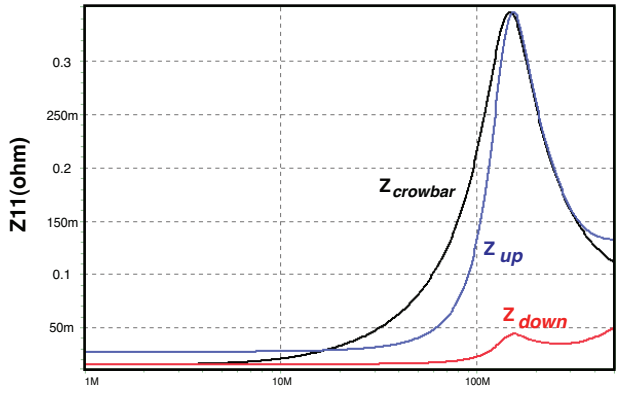


Figure 12.39 Three Driver-Impedance Profiles

Figure 12.40 shows the impact of increasing the on-chip decoupling capacitance value for $Z_{PDN}(f)$. Increasing the decoupling capacitance value reduces the PDN impedance and shifts the resonant frequency toward a lower frequency. However, its effectiveness decreases eventually.

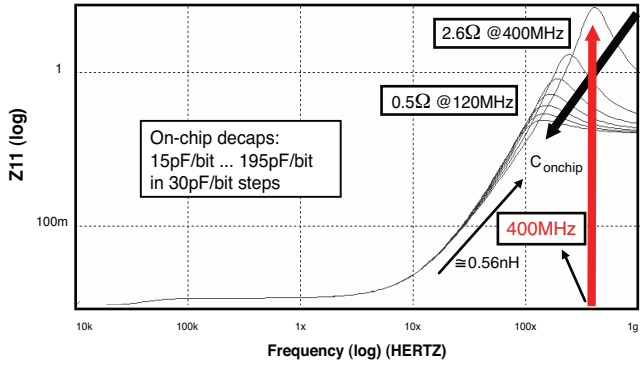


Figure 12.40 Impact of Increasing On-Chip Decoupling Capacitance

12.5.2 Coupled Power Noise and Worst-Case Switching Pattern

Self-generated noise is often well defined, and the target PDN impedance is tightly controlled. However, coupled noise is often neglected, and this can lead to unexpected system failures. There are several possible contributions to noise coupling, and each of these contributions requires a different worst-case access pattern in the system. Additionally, the self-induced noise on the internal supply rails must be considered, especially if it is correlated to the activity of the output drivers (for example, a pre-driver could be operating from the internal supply rail, rather than the output I/O supply).

To characterize coupled noise, the transfer impedance, seen between the current driver using one supply rail and the coupled noise at another supply rail, is analyzed. Figure 12.41 shows the different transfer impedances for noise coupling in the Vddr supply rail, as well as the impedance of Vddr-Vss, for self-induced noise.

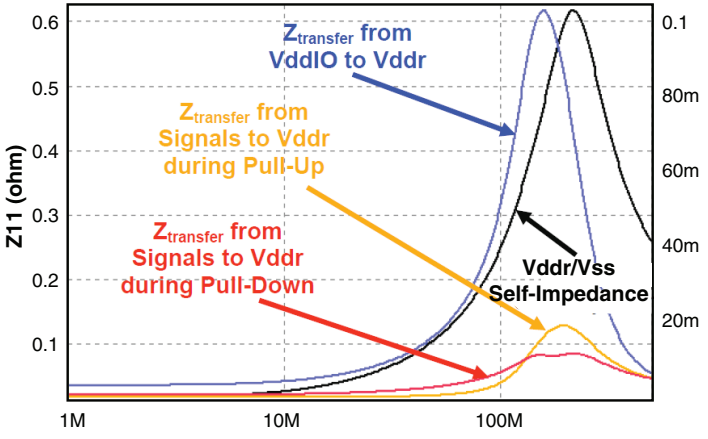


Figure 12.41 Self Impedance and Transfer Impedance for Supply Noise Generation on Vddr

The worst-case switching pattern is hard to find because multiple outputs must be considered. Furthermore, as the current loop changes, depending on the operation status (for example, pull-up, pull-down, and crowbar cases), it can no longer be calculated using AC analysis alone. One way to handle this problem is to generate a few patterns, based on AC analysis, and then perform a time-domain simulation to verify them. This is even more difficult for SSN analysis, because the PDN worst-case pattern must be combined with the worst-case ISI pattern. For general cases, one can use a time-domain numerical approach, such as the generalized peak distortion analysis described in Chapter 9, or the reverse pulse technique [20].

12.5.3 Noise Measurements and Correlation

Supply noise is difficult to observe at the package pin due to the package inductance filtering. Consequently, one often uses a sense line, which connects the power supply rail to an external pin through a transmission line, to monitor on-chip power noise. Alternatively, an on-chip noise monitor (as described in Chapter 16, "On-Chip Link Measurement Techniques") can be designed to measure supply noise directly. These approaches may not be possible for product design, except for test chips.

However, for SSTL drivers, transmitter on-chip supply and ground noise can be observed directly at the output of a driver. This method was initially developed for another memory system [21]. During this measurement, one driver transmits a constant value (either 1 or 0), depending on the target power or ground noise, while all the other drivers transmit the same switching data pattern. As long as the driver transmits a constant 1, the driver's supply rail (VddIO) is connected to

the output signal through the on-die termination resistor of the driver. Any noise on the VddIO supply rail is reflected as noise on the signal line. If the on-chip driver's impedance is known, the supply noise can be determined. Figure 12.42 illustrates this concept with the power rail noise.

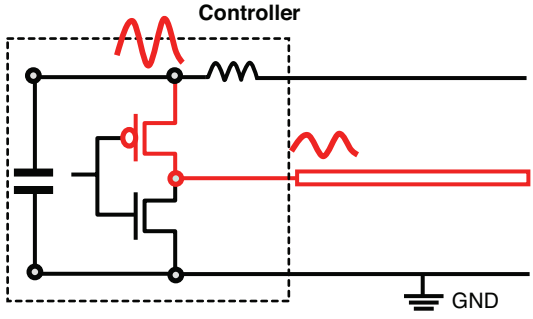


Figure 12.42 Supply Noise Reflected at the Signal Line

Figure 12.43 shows the signal, measured at a data line of an output driver that transmitted a long string of 0s, followed by a long string of 1s, while all the other drivers transmitted the worst-case pattern. The measurement shows the supply noise on both the VddIO and Gnd rails. During a pull-up operation, the output is connected to the VddIO rail, while during a pull-down operation, the signal is connected to the ground node. Note that the amplitude of the noise on both rails is comparable.

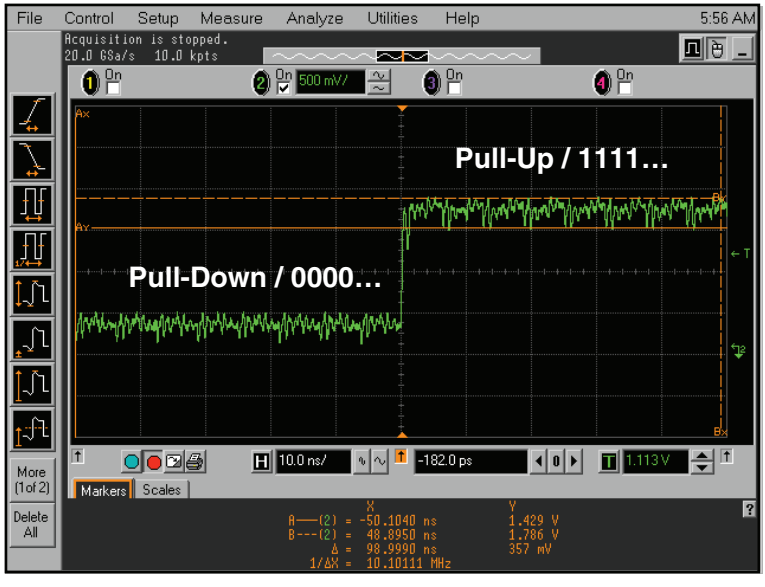


Figure 12.43 Supply and Ground Noise Signal-Line Measurements

Performing a measurement of one of the worst-case patterns (generated by AC analysis) and comparing it to the simulation shows a very good correlation, as shown in Figure 12.44. The simulated and measured waveforms not only correlate well in noise amplitude, but they also correlate reasonably well in noise signature. This confirms the accuracy of the modeling methodology described in this chapter.

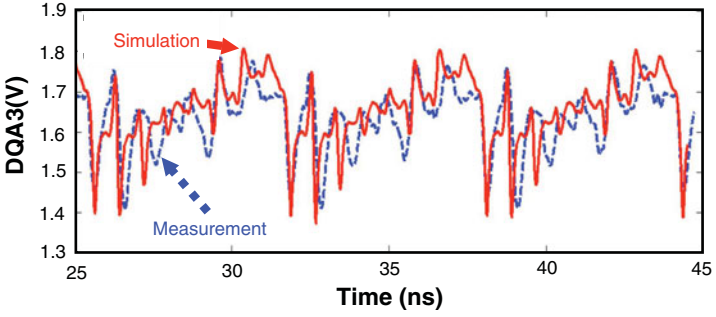


Figure 12.44 Signal-Line Supply Noise Measurement

12.6 Summary

SSN analysis requires modeling of both PDN and signals. Covering all the aspects of these models is beyond the scope of this book; this chapter presents only an overview. From a design perspective, the only way to reduce core power noise is through a good PDN design, which often increases the system design cost. Power noise is more critical for single-ended I/O interfaces than for differential interfaces, because they are highly susceptible to SSN. Fortunately, data coding is an effective way to reduce SSN (see Chapter 13).

The key points covered in this chapter are:

- SSN modeling requires complex channel and PDN models and excitations.
- Current mirrors can be used to reduce the channel model complexity without accuracy degradation.
- Modeling reduction, based on port reduction, is very helpful in reducing the simulation burden.
- Partial element representation requires careful attention to avoid misuse.

References

1. D. Oh, F. Ware, W. Kim, J.-H. Kim, J. Wilson, L. Luo, J. Kizer, R. Schmitt, C. Yuan, and J. Eble, "Pseudo-differential signaling scheme based on 4b/6b multiwire code," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2008, pp. 29–32.
2. A. Varma, M. Steer, and P. Franzon, "SSN issues with IBIS models," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2004, pp. 87–90.
3. Z. Yang, S. Hug, V. Arumugham, and I. Park, "Enhancement of IBIS modeling capability in simultaneous switching noise (SSN) and other power integrity related simulations—proposal, implementation, and validation," in *Proceedings of International Symposium on Electromagnetic Compatibility*, Aug. 2005, pp. 672–677.
4. B. Young, *Digital signal integrity: Modeling and Simulation with Interconnects and packages*, Prentice Hall, 2000.
5. R. Schmitt, J.-H. Kim, C. Yuan, J. Feng, W. Kim, and D. Oh, "Power integrity analysis of DDR2 memory systems during simultaneous switching events," presented at the IEC DesignCon, Santa Clara, CA, 2006.
6. J.-H. Kim, W. Kim, D. Oh, R. Schmitt, J. Feng, C. Yuan, L. Luo, and J. Wilson, "Performance impact of simultaneous switching output noise on graphic memory systems," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2007, pp. 197–200.
7. D. Oh, W. Kim, J.-H. Kim, J. Wilson, R. Schmitt, C. Yuan, L. Luo, J. Kizer, J. Eble, and F. Ware, "Study of signal and power integrity challenges in high-speed memory I/O designs using single-ended signaling schemes," presented at the IEC DesignCon, Santa Clara, CA, 2008.
8. R. Schmitt, J.-H. Kim, W. Kim, D. Oh, J. Feng, C. Yuan, L. Luo, and J. Wilson, "Analyzing the impact of simultaneous switching noise on system margin in gigabit single-ended memory systems," presented at the IEC DesignCon, Santa Clara, CA, 2008.
9. M. Ha, J.-H. Kim, D. Oh, and M. Swaminathan, "A study of reduced-terminal models for system-level SSO noise analysis," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2010, pp. 49–52.
10. Q. Qi, D. Quint, M. Frank, T. Michalka, and K. Bois, "Optimizing the package design with electrical modeling and simulation," in *Proceedings of Electronic Components and Technology Conference*, Orland, FL, May 2001, pp. 311–318.
11. Q. Qi, D. Quint, M. Frank, and T. Michalka, "Simulation of a coupled signal and power delivery system in an electronics package," in *Proceedings of Electronic Components and Technology Conference*, San Diego, CA, May 2002, pp. 111–117.

12. D. Oh and C.-C. Huang, "Efficient representation of multi-bit data bus structures by symmetric two-line models," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2002, pp. 141–144.
13. Z. Chen, "Crosstalk superposition of multiple aggressors in electronic package system pre-PD signal integrity simulations," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2006, pp. 115–118.
14. R. Oikawa, D. Gope, and V. Jandhyala, "Broadband SSO modeling for a weak signal return-path system based on the large-scale signal-power combined three-dimensional full-wave BEM solver model," in *Proceedings of Electronic Components and Technology Conference*, Las Vegas, NV, Jun. 2010, pp. 638–645.
15. M. Tsuk, "The use of loop inductances in signal integrity modeling," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2001, pp. 81–84.
16. M. Swaminathan and A. E. Engin, *Power Integrity Modeling and Design for Semiconductors and Systems*, Prentice Hall, 2008.
17. A. E. Engin, W. John, G. Sommer, W. Mathis, and H. Reichl, "Modeling of striplines between a power and a ground plane," *IEEE Transactions on Advanced Packaging*, vol. 29, no. 3, pp. 415–426, Aug. 2006.
18. K. (Dan) Oh, "Efficient modeling of interconnections and capacitive discontinuities in high-speed digital circuits," Ph.D. dissertation, University of Illinois at Urbana-Champaign, May 1996.
19. C. Wei, R. F. Harrington, J. R. Mautz, and T. K. Sarkar, "Multiconductor transmission lines in multilayered dielectric media," *IEEE Transactions on Microwave Theory and Techniques*, vol. 32, no. 4, pp. 439–450, Apr. 1984.
20. V. Drabkin, C. Houghton, I. Kantorovich, and M. Tsuk, "Aperiodic resonant excitation of microprocessor power distribution systems and the reverse pulse technique," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2002, pp. 175–178.
21. R. Schmitt, C. Yuan, and W. Kim, "Modeling and Correlation of Supply Noise for a 3.2GHz Bidirectional Differential Memory Bus," presented at the IEC DesignCon, Santa Clara, CA, 2005.

CHAPTER 13

SSN Reduction Codes and Signaling

Dan Oh

As discussed in the previous chapter, simultaneous switching noise (SSN) is one of the major bottlenecks facing single-ended signaling interfaces. SSN is caused by current change over the inductance of a power distribution network (PDN). Because the PDN's inductance is mainly due to the package's parasitic inductance, SSN is conventionally reduced by increasing the package ball or pin counts (see Figure 12.2, in Chapter 12, "SSN Modeling and Simulation," for the pin-count trend in graphics memory systems). However, increasing the pin count has limited effectiveness, because inductance decays slowly as the number of pins increases, causing physical design improvements to be difficult and expensive.

Fortunately, SSN can be minimized using alternative approaches. Because SSN is generated by current changing, on the power or ground rails, one can code the data in a way that minimizes the amount of current changes. Conventional data coding is used in temporal space (sequence in time) to achieve DC balance, or to guarantee a finite number of switching activities for Clock-to-Data Recovery (CDR) circuitry. Although the concept of using data coding to reduce power supply noise seems new, it has already been used in modern high-speed graphics memory systems (as will be shown later).

The simplest type of data coding to reduce SSN is a differential code. Each bit (0 or 1) is sent over two wires as 1 and –1, or –1 and 1. This keeps the total current on the bus constant, regardless of the transmitted data pattern, which results in zero supply noise (however, supply noise due to the pre-driver still exists). An additional advantage of this code is that it can be sent and received differentially, resulting in true differential signaling. Besides SSN reduction, differential signaling has many other advantages over single-ended signaling. For example, it is immune to crosstalk, and needs no explicit reference voltage. In addition, receiving the data differentially effectively doubles the input voltage swing as compared to the single-ended case. Unfortunately, the downside of differential signaling is that it requires twice the number of signal

pins. An additional, minor drawback of a differential interface is that the wires for a differential pair must be routed tightly, and intra pair skew must be minimized to maximize common-mode noise rejection and to minimize differential to common mode conversion.

Due to the additional pin requirements, differential signaling is often expected to run at least two times faster than the single-ended case. Using differential signaling is justified, if the device can support the 2x speed with reasonable power consumption. If the goal of coding is only to reduce power supply noise, then more efficient signaling or coding schemes are available. This chapter explores some of these coding options. Section 13.1 begins with data bus inversion (DBI) code, used in graphics memory systems [1–5]. Although DBI code reduces supply noise, its effectiveness is not optimal in terms of noise reduction. Section 13.2 discusses a pseudo differential signaling scheme that minimizes SSN more effectively. This signaling scheme uses a 4b6b code, which allows signals to be received differentially, but at the cost of some receiver complexity.

13.1 Data Bus Inversion Code

Data bus inversion is designed to reduce the power consumption used in POD signaling [1] (POD signaling was reviewed in Section 12.3.2). As Figure 12.27 in Chapter 12 shows, the channel is only terminated at the power rail for POD signaling, so it only consumes DC power during low states, which draws a current from termination to ground. The main goal of DBI code is to minimize the number of low states (0s) on the data bus to reduce the power consumption. DBI was first used in GDDR4 memory systems. GDDR4 offers two encoding methods for data bus inversion, referred to as DBI-AC and DBI-DC. Both methods encode the polarity of the bus, and the coding (inversion) information is sent over using an extra signal, but they differ in terms of the coding criterion. To avoid an extra pin for this code signal, GDDR4 uses the WRITE data mask as a DBI flag during READ operations, and the RDQS as a DBI flag during WRITE operations.

DBI-AC reduces the total number of bit changes between consecutive states, by inverting the polarity of the bus values [1] [2]. Proper implementation of this encoding method, which should also consider the state of the DBI bit, inverts the bus polarity to limit the maximum number of state changes for an N-bit data bus and DBI flag to $N/2$. As a result, this coding scheme minimizes AC power consumption, but it does not consider DC power consumption. As it minimizes the switching current, it reduces high-frequency power noise. However, it cannot reduce a medium-frequency power noise near a PDN resonant frequency. In terms of power consumption, DBI-AC makes more sense for unterminated channels (such as LP-DDR1/2), because DC power is zero for these channels.

DBI-DC reduces the "weight" (or the total number of 1 or 0 states) of the bus by inverting the polarity of the bus values [3]. Proper implementation of this encoding method will limit the "weight" of an N-bit bus to $N/2$. The symbol to be minimized depends on the type of bus termination. With GDDR4, which uses VDDQ-referenced POD signaling, power is saved by reducing the total number of 0 states, because they consume static current. Table 13.1 shows sample code for both the DBI-DC and DBI-AC cases. The code is applied to eight bits, as is done in typical

graphics systems. Due to the additional DBI pin requirement, DBI coding is unlikely to be used for buses less than eight bits.

Table 13.1 DBI Coding Example

Non-Coded	DBI-AC		DBI-DC	
	Data	DBI Bit	Data	DBI Bit
11000...	11111...	00111...	11001...	00101...
10000...	10111...		10001...	
11111...	11000...		11110...	
11000...	11111...		11001...	
11010...	11101...		11011...	
11010...	11101...		11011...	
11000...	11111...		11001...	
11011...	11100...		11010...	

To study the effectiveness of DBI codes, in terms of power saving, a simple Fourier analysis is performed using MATLAB. To create a system that can be quickly evaluated in MATLAB, the channel effect is excluded, the PDN impedance is assumed ideal, and only the output signaling currents are considered. The influence of DBI on the aggregate behavior of the signaling currents is evaluated for an 8-bit GDDR4 bus; using 1.8V VDDQ referenced termination, a 60Ω pull-up and driver, and a driver with a 40Ω pull-down impedance. The current of interest (IVDDQ) is the sum of the individual signaling currents.

Figure 13.1 shows the results of Fourier analysis, used to compare the magnitude spectrum for the 8-bit bus, with and without DBI encoding, at a signaling rate of 4Gb/s [6]. (The results for the DBI encoded buses include the current for the DBI signal.) The DBI-AC encoding actually causes an increase in spectral terms and power consumption, due to signaling current, when compared to the non-coded bus. This is not surprising, because the DBI-AC encoding algorithm was originally intended to reduce the switching activity of non-terminated buses, and is "unaware" of the static current consumption of a terminated bus.

However, the bus using DBI-DC encoding shows a decrease in signaling current across the entire spectrum, when compared to the non-coded bus. In theory, the DBI-DC algorithm will reduce the maximum di/dt of the encoded bus by 50%, and decrease power consumption due to signaling current by 18%. A point of interest is that the Fourier analysis indicates a reduction by 39% in the RMS current of the spectral terms (excluding DC) when comparing DBI-DC to the non-coded bus. Table 13.2 shows these results for the ideal 8-bit bus [7]. The table compares the

worst-case DC current, the spectral terms from the Fourier analysis, and the average current consumption for the non-coded and encoded methods.

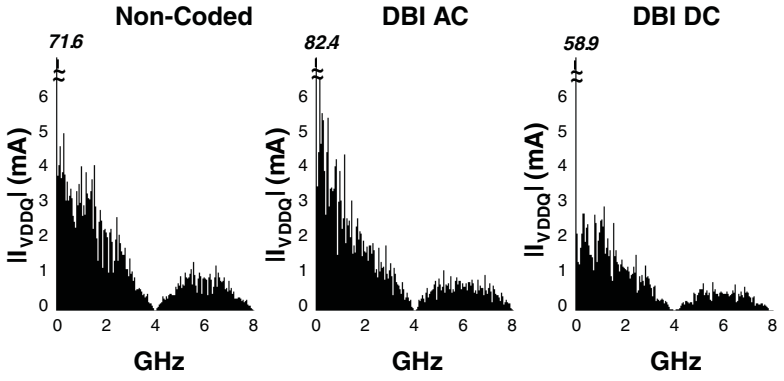


Figure 13.1 Spectrum of Aggregate Signaling Current for "Ideal" 8-Bit Bus

Table 13.2 Current Amplitudes for Non-Coded, DBI-AC, and DBI-DC Coded Busses

Aggregate Bus Current	Non-Coded	DBI-AC	DBI-DC
Worst-case DC (relative %)	144 mA (100%)	162 mA (113%)	72 mA (50%)
AC terms only (relative %)	25.7 mA (100%)	27.1 mA (105%)	15.7 mA (61%)
Average output (relative %)	71.6 mA (100%)	82.4 mA (115%)	58.9 mA (82%)

The next example uses the GDDR5 channel, shown in Figure 13.2. First, five drivers are modeled, including the victim lines, for full crosstalk impact. Next, 35 additional drivers are modeled using the current mirrors described in Chapter 12. A fast-time domain simulation (see Chapter 10, "Clock Models in Link BER Analysis") simulates the eye diagrams. Figure 13.3 shows the eye diagrams and bathtub curves at 5Gb/s for the case with and without DBI-DC coding. The worst-case data pattern with all the aggressors switching at the same time is used for input. About 10% UI margin is gained at BER $= 10^{-15}$. Now, a PRBS data pattern is used for each aggressor, instead of the worst-case data pattern. Figure 13.4 shows the resulting eye diagrams. Unlike the previous case, DBI-DC does not improve the timing margin much. The latest GDDR5 uses data scrambling [5], and DBI code is less effective when data scrambling is applied.

In summary, although DBI reduces the power noise less than 40%, it is a very effective and simple way to mitigate worst-case power noise issues, and offers the added benefit of power saving. The latest GDDR5 devices extend the bus inversion feature to their address and command lines (ABI). As I/O data rates continue to increase, the performance advantage of using DBI or

other codes will be more significant. Note that, when extra pins are used for coding, the effective bandwidth of the overall system must be carefully calculated by considering the pin overhead.

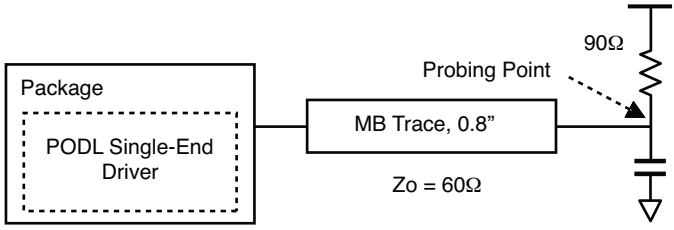


Figure 13.2 SSN Simulation Setup for GDDR5 Graphics Channel

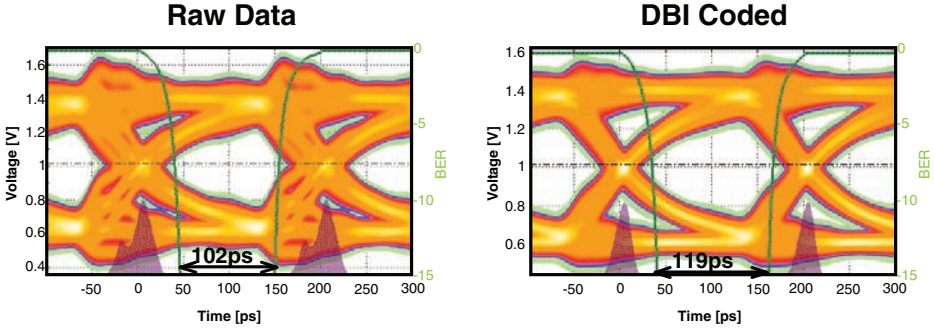


Figure 13.3 Worst-Case Data Pattern Eye Diagrams, without DBI and with DBI-DC

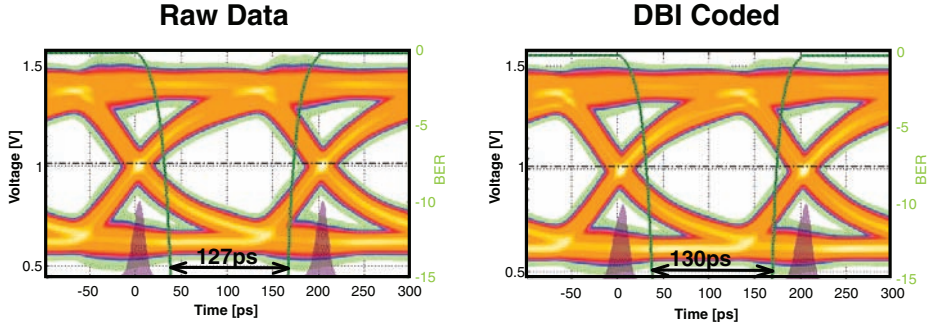


Figure 13.4 PRBS Data Pattern Eye Diagrams, without DBI and with DBI-DC

13.1.1 Noise Shaping using DBI

DBI-DC reduces the worst-case power noise by limiting the number of drivers having the same state, whereas DBI-AC reduces the high-frequency switching noise by limiting the number of drivers to be switched. However, supply noise peaks at the medium frequency range near PDN resonance, and neither of these methods accounts for the frequency response behavior of the PDN. If we treat the aggregated bus signal as a single signal, this signal can be shaped (coded) to avoid the peak PDN resonance frequency, thereby reducing the worst-case supply noise.

An advanced DBI coding technique, called DBI-SS, encodes the activity of a bus in a manner that shapes the spectrum of the aggregate bus signaling current [7]. The coding shapes the spectral content of the bus signal, in order to avoid the PDN resonance frequency. The spectrum of the signal is shaped using equalization, similar to that used with conventional data signals. Because the equalization is performed using the existing DBI signal, DBI-SS can be easily implemented in current devices. The simplest form of bus equalization is to create a notch around the PDN resonance frequency. The bus signal excites the minimum spectrum content at the worst-case PDN impedance location. A digital filter can be used in the encoder to generate a general frequency shape. This filter can operate at the bit-rate, or at a sub-rate (decimation). Figure 13.5 shows a sample implementation of the DBI-SS encoding technique. The encoder adds a "detection filter," and a threshold-based decision making block, to the traditional DBI encoder. No additional circuitry is required for a decoding block.

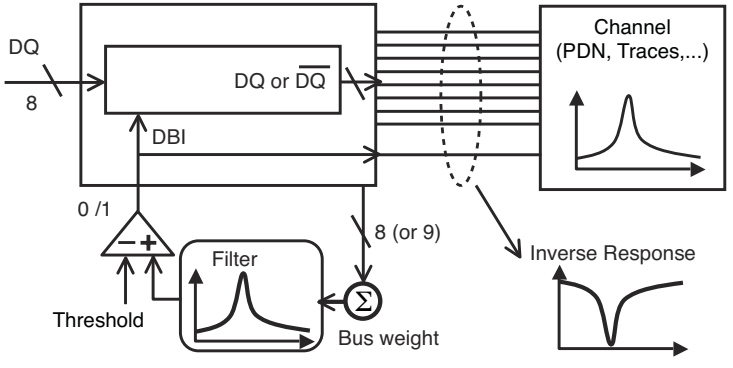


Figure 13.5 Block Diagram of DBI-SS Encoder

13.1.2 Hybrid DBI-SS Implementations for Reducing Both Pre-Driver and Output Driver Noise

A high-speed I/O driver consists of both a pre-driver and an output driver. Pre-drivers are often implemented as a CMOS signal and use a separate supply from the output driver. Because a typical clock tree shares the same power supply used by the pre-driver, any supply voltage noise that the pre-driver generates not only adds jitter to the pre-driver, but it also affects the CMOS clock

buffers, causing additional jitter. Unlike output supply noise (which occurs only with a single-ended signaling interface), the pre-driver noise induced jitter occurs even in differential signaling interfaces, because the pre-driver is still single-ended.

Figure 13.6 shows various waveforms for a DDR2 channel. The voltage and current waveforms of the data signal are shown on the top. The currents on the output supply rail and pre-driver supply rail are shown on the bottom. As the figure indicates, the output supply noise current is significantly different from the pre-driver supply noise current. In the pre-driver's case, the current is excited whenever there is a logic switch, and it does not have a steady term. Additionally, the PDN for the output supply is different from the pre-driver's case. Transmitting a burst from a toggling data pattern, at the resonant frequency of the pre-driver PDN, will generate significant power noise and errors could start occurring [8]. As shown in Figure 13.6, the pre-driver noise is a function of the number of bit changes, instead of the bus weight, so neither DBI-DC nor DBI-SS are effective against pre-driver noise.

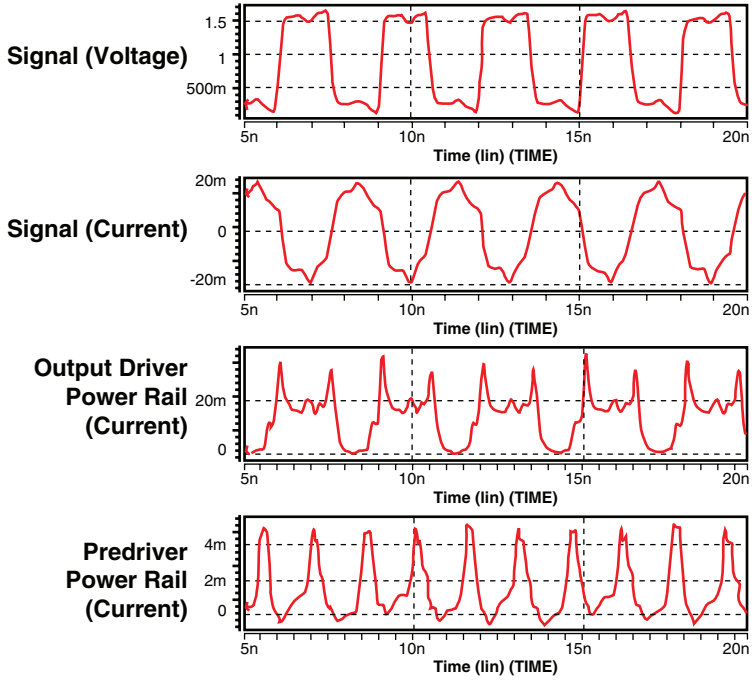


Figure 13.6 Waveforms of DDR2 Pre-Driver and Output Driver Currents on Power Rails

Fortunately, DBI-SS can be easily modified to utilize the pre-driver PDN frequency response by changing the input signal to the number of switching bits, instead of the bus weight.

The DBI-SS method can be further modified to handle both pre-driver and output driver power noise by using the weighted average of the two input signals: one input using the number of 1s, and the other input using the number of switching bits. Figure 13.7 illustrates this hybrid scheme; the bus coding automatically reduces either the output driver or pre-driver power noise, based on the current noise levels.

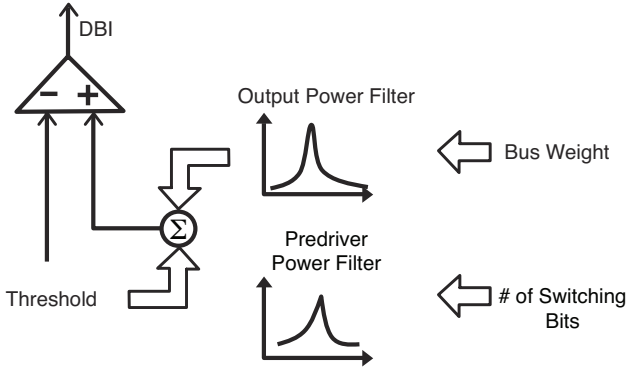


Figure 13.7 Hybrid Filter Scheme for DBI-SS that Addresses Both Pre-Driver and Output Driver Power Noise

13.2 Pseudo Differential Signaling Based on 4b6b Code

The examples in Section 13.1 demonstrate that DBI-DC coding eliminates worst-case power noise by limiting the magnitude of the current changes. This section examines a new coding scheme that not only removes the current changes, but also maintains most of the desirable properties of differential signaling [9]. This coding, referred to as *vector signaling*, provides pseudo-differential signaling. From the transmitter's point of view, vector signaling is simply coded single-ended signaling. The receiver can be implemented using either a single-ended receiver or a differential receiver with additional circuitry. However, the benefits of using vector signaling come at the cost of additional samplers (>x2.5 larger than the case of the single-ended receiver). Therefore, this section also covers an alternative receiver design, which requires fewer samplers [10]. Vector signaling is an alternative signaling interface for both single-ended memory applications and differential parallel interfaces.

13.2.1 Generalized Vector Signaling

The code space for vector signaling is chosen to have zero SSO noise. This means that the number of 1s (Hamming weight) is fixed for all codewords to maintain constant total power and ground currents. In general, vector signaling encodes n-bits of data (where n may not be an integer) into an M-bit codeword, which is sent over M wires. The pin efficiency is n/M. (The pin efficiency for differential signaling and single-ended signaling are 0.5 and 1, respectively.)

Table 13.3 shows valid sets of codewords for the *M*=4 case. The code's efficiency is maximized when the disparity between the number of 1s and 0s is the smallest.

Table 13.3 Valid Sets of Codewords for *M*=4

Hamming Weight	1	2	3
Codewords	0001	0011	0111
	0010	0101	1011
	0100	1001	1101
	1000	0110	1110
		1010	
		1100	
Pin Efficiency	0.5	0.65	0.5

Vector signaling also supports pair-wise differential samplers. These differential samplers eliminate the reference voltage and achieve 2x larger input swing than a sampler using single-ended signaling. However, unlike differential signaling, where there are two signaling voltage levels (–1 and 1), the differential sampler sees three voltage levels (–1, 0, and 1). Based on the previous code selection, the decoder can ignore the 0 level, which is a metastable state for a differential receiver. In other words, the output of the 0 level can be treated as a "don't care."

Figure 13.8 illustrates a four-wire vector signaling system. The four-wire system requires six differential samplers. Table 13.4 is the decoder table for a Hamming weight of 2. The 0 level can be treated as a "don't care" because all the symbols have unique locations of –1 and 1.

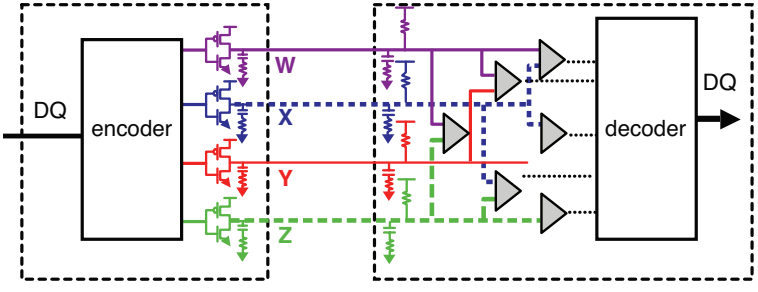


Figure 13.8 Sample Four-Wire Vector Signaling System

Table 13.4 Decoding Table for Vector Signaling, with M=4 and Hamming Weight of 2

Symbol	Transmitter	Receiver					
	(w x y z)	w-x	w-y	w-z	x-y	x-z	y-z
A	0011	δ	–1	–1	–1	–1	δ
B	0101	–1	δ	–1	1	δ	–1
C	1001	1	1	δ	δ	–1	–1
D	0110	–1	–1	δ	δ	1	1
E	1010	1	δ	1	–1	δ	1
F	1100	δ	1	1	1	1	δ

Figure 13.9 shows two eye diagrams, based on six-wire vector signaling with 15 differential samplers. The first eye diagram shows the raw eye diagram of the received signal. The second eye diagram shows the eye diagram after filtering the metastable states. Even after filtering, some extra bands still remain. These are due to the transitions to and from the metastable level to valid levels. Fortunately, these bands are located outside the inner eye opening, and do not degrade the final performance. Significant intersymbol interference (ISI) is observed due to the extra capacitance loading associated with the large number of samplers.

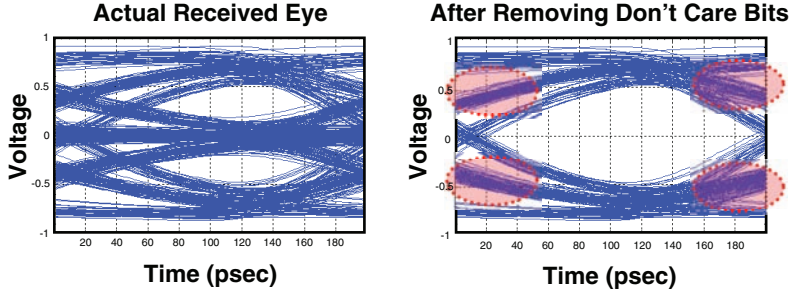


Figure 13.9 Raw Eye Diagram Received by a Vector Signaling Differential Sampler and Effective Eye Diagram after Filtering Metastable Inputs [9] (© 2008 IEEE)

Table 13.5 lists the pin efficiency and the required number of samplers for various M values. The pin efficiency improves as the wire number increases, but at the expense of an increased number of samplers. In general, odd numbers of M provide poor pin efficiency. The $M = 6$ case (six-wire vector signaling) provides a good balance between pin efficiency and receiver design complexity, and is studied in more detail in the following section.

Table 13.5 Pin Efficiency and Required Number of Samplers for Various Codes

M	*n*	# of Symbols	Efficiency	# of Samplers	Hamming Weight
3	1	3	0.53	3	1
4	2	6	0.65	6	2
5	3	10	0.66	10	2
6	**4**	**20**	**0.72**	**15**	**3**
7	5	35	0.73	21	3
8	5	70	0.77	28	4
9	6	126	0.78	36	4
10	7	252	0.80	45	5

13.2.2 4b/6b Vector Signaling

As illustrated in Table 13.5, six-wire vector signaling (which maps four-bit data to six-bit codewords) appears to be the most practical in terms of both pin efficiency and receiver complexity. Figure 13.10 illustrates a six-wire vector signaling system. As the figure shows, the 15 samplers make the design quite complex, because the signals must be routed with good trace-length matching. It also adds significant capacitance loading to the channel (five samplers per line). As shown in Figure 13.9, there is significant eye closure, due to the ISI caused by the increased capacitance loading. To mitigate this issue, one can use a preamplifier, or linear equalizer, to shield the extra capacitance due to the multiple samplers. Table 13.6 provides the encoding and decoding table.

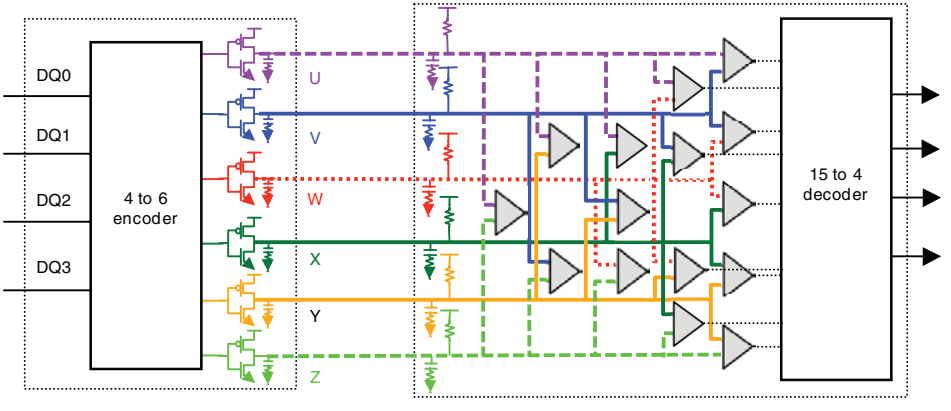


Figure 13.10 Sample Six-Wire Vector Signaling System

Table 13.6 Encoding and Decoding for Six-Wire Vector Signaling

Symbol	Transmitter						Receiver													
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	U	V	W	X	Y	Z	U-V	U-W	U-X	U-Y	U-Z	V-W	V-X	V-Y	V-Z	W-X	W-Y	W-Z	X-Y	X-Z	Y-Z
A	1	1	1	0	0	0	δ	δ	1	1	1	δ	1	1	1	1	1	1	δ	δ	δ
B	1	1	0	1	0	0	δ	1	δ	1	1	1	δ	1	1	–1	δ	δ	1	1	δ
C	1	1	0	0	1	0	δ	1	1	δ	1	1	1	δ	1	δ	–1	δ	–1	δ	1
D	1	1	0	0	0	1	δ	1	1	1	δ	1	1	1	δ	δ	δ	–1	δ	–1	–1
E	1	0	1	1	0	0	1	δ	δ	1	1	–1	–1	δ	δ	δ	1	1	1	1	δ
F	1	0	1	0	1	0	1	δ	1	δ	1	–1	δ	–1	δ	1	δ	1	–1	δ	1
G	1	0	1	0	0	1	1	δ	1	1	δ	–1	δ	δ	–1	1	1	δ	δ	–1	–1
H	1	0	0	1	1	0	1	1	δ	δ	1	δ	–1	–1	δ	–1	–1	δ	δ	1	1
I	1	0	0	1	0	1	1	1	δ	1	δ	δ	–1	δ	–1	–1	δ	–1	1	δ	–1
J	1	0	0	0	1	1	1	1	1	δ	δ	δ	δ	–1	–1	δ	–1	–1	–1	–1	δ
K	0	1	1	1	0	0	–1	–1	–1	δ	δ	δ	δ	1	1	δ	1	1	1	1	δ
L	0	1	1	0	1	0	–1	–1	δ	–1	δ	δ	1	δ	1	1	δ	1	–1	δ	1
M	0	1	1	0	0	1	–1	–1	δ	δ	–1	δ	1	1	δ	1	1	δ	δ	–1	–1
N	0	1	0	1	1	0	–1	δ	–1	–1	δ	1	δ	δ	1	–1	–1	δ	δ	1	1
O	0	1	0	1	0	1	–1	δ	–1	δ	–1	1	δ	1	δ	–1	δ	–1	1	δ	–1
P	0	1	0	0	1	1	–1	δ	δ	–1	–1	1	1	δ	δ	δ	–1	–1	–1	–1	δ
Q	0	0	1	1	1	0	δ	–1	–1	–1	0	–1	–1	–1	δ	δ	δ	1	δ	1	1
R	0	0	1	1	0	1	δ	–1	–1	δ	–1	–1	–1	δ	–1	δ	1	δ	1	δ	–1
S	0	0	1	0	1	1	δ	–1	δ	–1	–1	–1	δ	–1	–1	1	δ	δ	–1	–1	δ
T	0	0	0	1	1	1	δ	δ	–1	–1	–1	δ	–1	–1	–1	–1	–1	–1	δ	δ	δ

Table 13.6 is similar to the decoding table for the four-wire case: the 0 level can be still considered as a "don't care" by observing that all the symbols have unique locations of –1 and 1. Also, notice that all the vector signal codes have a symmetric decoding pattern. Although there are twenty codewords (symbols), only sixteen symbols are needed to map the four-bit information. The remaining four symbols can be used for other purposes, such as error detection or data masking.

The major challenge in implementing six-wire vector signaling is routing the traces to the 15 samplers. All the traces must be length matched, just as was done for the four-wire example, because any mismatch could induce noise at the differential samplers. This is especially hard with DRAM processes, as the number of metal layers is limited.

13.2.3 Alternative Receiver Design with a Lower Number of Samplers

The previous implementation of six-wire vector signaling requires a large number of samplers, which adds significant complexity to the receiver design. In fact, the number of required samplers is a quadratic function of the number of wires. This section offers a simpler implementation, which uses a smaller number of samplers [10] and allows for easier trace routing.

A new decoding table is constructed as follows. First, the wires are divided into two subgroups: $\{U,V,W\}$ and $\{X,Y,Z\}$. Then, we compare the wires within each subgroup. This clearly does not result in enough information to distinguish all the symbols. To aid the decoding, an additional six-terminal comparator is introduced. Table 13.7 shows the completely decoded information. Note that the symbols A and T are not valid codewords in this implementation, and are crossed out in the table. Similar to the previous case, the new decoding table guarantees unique locations of –1s and 1s, and has a symmetric decoding pattern. Figure 13.11 illustrates this special implementation of six-wire vector signaling. The major performance limiter, in this scheme, is the six-terminal comparator design.

Now, consider a more general formulation for a larger number of wires. Divide the total of M wires into P subgroups, with sizes of $\{N_1, N_2,..., N_P\}$. P must be an even number, and all the sizes of the subgroups must be odd numbers. Each subgroup is treated as an independent vector signaling system, thereby reducing the complexity of the receiver design. For each subgroup, assume that there is the minimum disparity between 1s and 0s, which means that, depending on the subgroup, there will be either one more 1 or 0 for all the symbols within a subgroup. By inverting the symbols in each subgroup, an additional P symbols are now available. Because the assumption is that P is an even number, the vector signaling can be further applied for these additional signals.

In the six-wire system, P is 2, N_1 and N_2 are 3, and the number of codewords for each group is 3. We force the two inverting signals to be of opposite signs, in order to use the six-terminal differential comparator. Therefore, the final number of available codewords is $3 \times 3 \times 2 = 18$.

Table 13.7 Encoding and Decoding for 6-Wire Vector Signaling with Simplified Receiver Design

Symbol	Transmitter						Receiver						
	1	2	3	4	5	6	1	2	3	4	5	6	7
	U	V	W	X	Y	Z	U-V	U-W	W-U	X-Y	Y-Z	Z-X	(U+V+W)-(X+Y+Z)
~~A~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~δ~~	~~δ~~	~~1~~	~~1~~	~~1~~	~~δ~~	~~1~~
B	1	1	0	1	0	0	δ	1	δ	1	1	1	1
C	1	1	0	0	1	0	δ	1	1	δ	1	1	1
D	1	1	0	0	0	1	δ	1	1	1	δ	1	1
E	1	0	1	1	0	0	1	δ	δ	1	1	-1	1
F	1	0	1	0	1	0	1	δ	1	δ	1	-1	1
G	1	0	1	0	0	1	1	δ	1	1	δ	-1	1
H	1	0	0	1	1	0	1	1	δ	δ	1	δ	-1
I	1	0	0	1	0	1	1	1	δ	1	δ	δ	-1
J	1	0	0	0	1	1	1	1	1	δ	δ	δ	-1
K	0	1	1	1	0	0	-1	-1	-1	δ	δ	δ	1
L	0	1	1	0	1	0	-1	-1	δ	-1	δ	δ	1
M	0	1	1	0	0	1	-1	-1	δ	δ	-1	δ	1

N	0	1	0	1	1	0	-1	δ	-1	-1	δ	1	-1
O	0	1	0	1	0	1	-1	δ	-1	δ	-1	1	-1
P	0	1	0	0	1	1	-1	δ	δ	-1	-1	1	-1
Q	0	0	1	1	1	0	δ	-1	-1	-1	0	-1	-1
R	0	0	1	1	0	1	δ	-1	-1	δ	-1	-1	-1
S	0	0	1	0	1	1	δ	-1	δ	-1	-1	-1	-1
~~T~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~δ~~	~~δ~~	~~-1~~	~~-1~~	~~-1~~	~~δ~~	~~-1~~

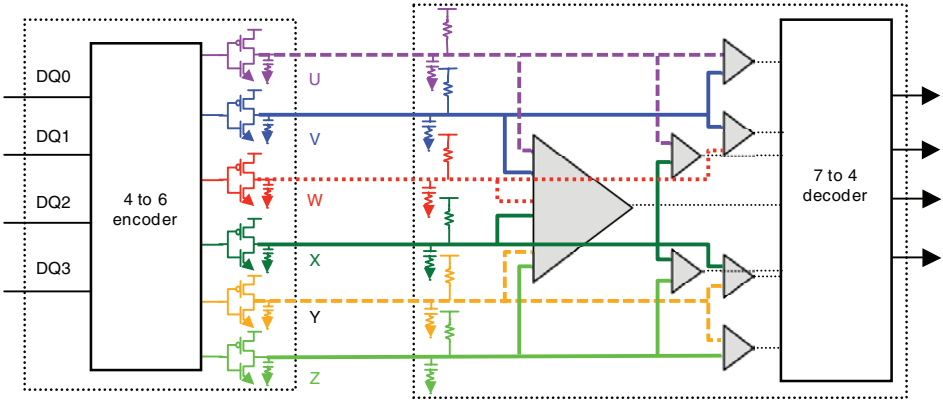


Figure 13.11 Six-Wire Vector Signaling System with Six-Terminal Comparator

13.2.4 Performance Comparison

In this section, the performance of the vector signaling system is evaluated. A GDDR graphics memory channel is considered as the test platform. Figure 13.12 illustrates the channel setup. This setup represents a high-end graphics card system, which uses a four-layer DRAM package, and whose traces are routed using stripline to minimize the crosstalk impact. (The simulation plots presented in the previous sections used the same setup.)

Figure 13.13 compares a single-ended eye diagram with the one from the six-wire vector signaling system with fifteen samplers. Notice the difference in scale between the two plots in Figure 13.13. The eye opening improvement, due to vector signaling, is almost doubled in this example.

Although vector signaling enlarged the eye opening significantly, the eye diagram showed significant ISI due to extra capacitive loading from the multiple samplers. Figure 13.14 shows eye diagrams with different Ci values and equalization options. The first case with extra capacitance shows a closed eye due to ISI at 9 Gb/s. The second case without the extra capacitive loading shows a fairly decent eye opening. The third case when using a preamplifier with a 10GHz corner frequency results in similar performance to that of the second case. Therefore, a preamplifier can shield the samplers' capacitance very effectively. Further eye improvement can be achieved by applying a linear equalizer with 3dB gain at 4 GHz. The resulting eye diagram is shown in the fourth diagram.

The remainder of this section compares the performance of various system configurations. The single-ended graphics system, shown in Figure 13.2, represents a high-end graphics system with a four-layer DRAM package. The baseline will be the performance of this system (HE Baseline). A low-cost version of the system, which uses microstrip instead of stripline for the motherboard traces, and a two-layer DRAM package instead of four-layer, represents a Low-Cost System. The high-end system with DBI, vector signaling, and differential signaling are also evaluated.

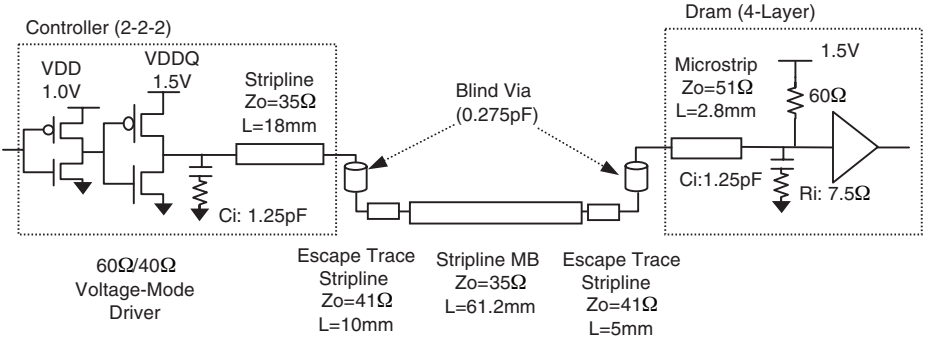


Figure 13.12 Graphics Memory Channel Setup

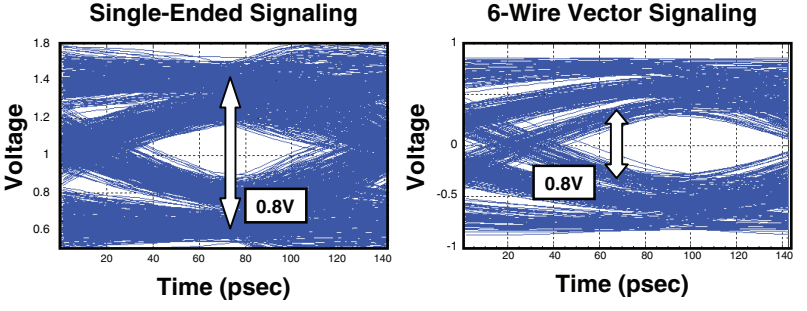


Figure 13.13 Comparison of 7Gb/s Eye Diagrams, for Single-Ended Signaling and 6-Wire Vector Signaling

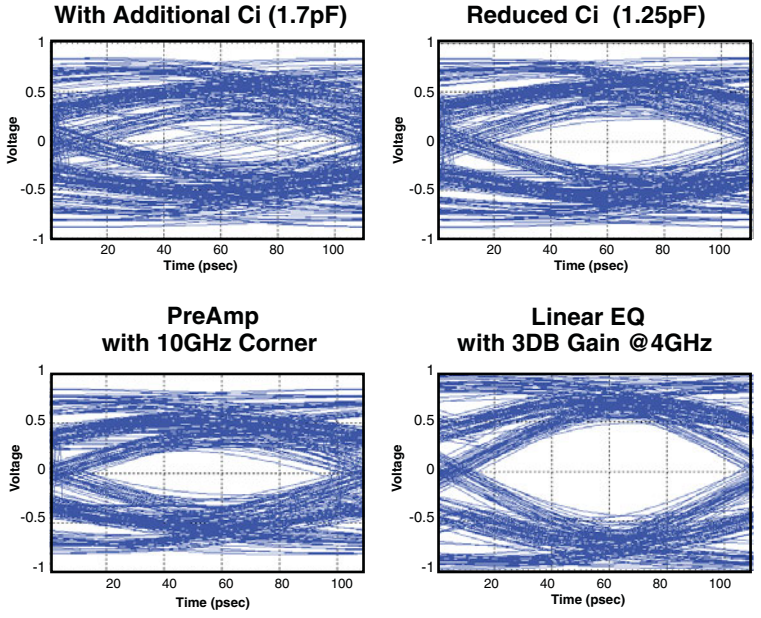


Figure 13.14 Six-Wire Vector Signaling Eye Diagrams at 9Gb/s: with Extra Capacitance Due to the 15 Samplers, without the Extra Capacitance, with a Preamplifier with 10GHz Corner Frequency, and with a Linear Equalizer with a 3dB Gain at 4GHz [9] (© 2008 IEEE)

For this comparative analysis, the target data rate is determined by using an eye mask 100mV in height, and ½UI in width. Design of experiments, similar to those described in Chapter 7, "Manufacturing Variation Modeling," is performed to account for potential manufacturing variations. Figure 13.14 illustrates the results, after comparing the performance of each configuration to the high-end baseline system. No pin overhead was considered in this comparison.

Significant performance was lost in the low-cost system, due to crosstalk, when compared to the baseline. Both DBI and vector signaling show reasonable performance improvement. However, if the pin overhead is accounted for due to coding, both cases do not show any significant performance improvement. DBI uses one additional pin per byte, whereas vector signaling uses an additional four pins per byte. On the other hand, both vector signaling with a linear equalizer and differential signaling show excellent performance improvement, even after accounting for the pin overhead.

Finally, the potential drawbacks of vector signaling are susceptibility to crosstalk, and tight length-match requirements, over a large number of wires. Although vector-signaling performance is not as great as in the differential signaling case, it is an interesting alternative to differential signaling. This is particularly true for slow DRAM process, which would require

significant power consumption and area to support high data rates required by the differential signaling.

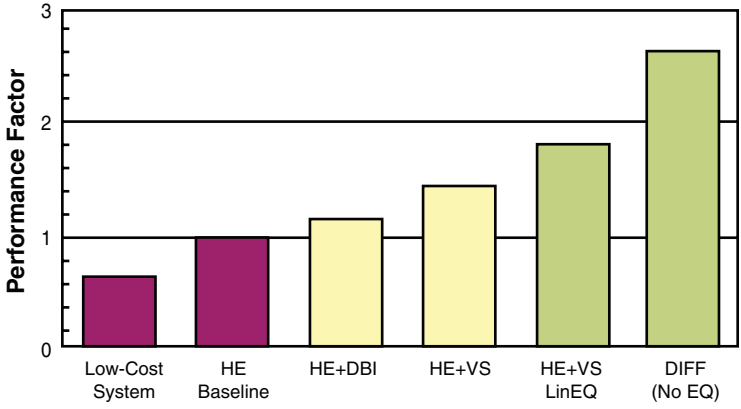


Figure 13.15 Performance Comparison of Various Single-Ended Technologies vs. Differential Signaling

13.3 Summary

This chapter discusses two bus-coding classes used to reduce supply noise. The first class of coding uses a data bus inversion (DBI) signal to reduce the worst-case supply noise. The chapter also discusses an advanced version of this method, which can "equalize" the bus activity. The second class of coding uses a balanced code to "completely" eliminate SSN. The method results in pseudo differential signaling.

Both of these coding classes offer benefits in addition to SSN reduction, such as lower power consumption or better signal quality due to the pseudo differential signaling nature. One can derive other noise reduction codes, based on either minimizing the disparity between 1s and 0s over the bus, or fixing the bus weight. However, they do not offer any benefits besides noise reduction. In fact, the existing 8b10b code can be applied over pin space, instead of temporal space to minimize supply noise [11]. It is a very efficient code, in terms of supply noise reduction.

References

1. R. J. Fletcher, "Integrated circuit having outputs configured for reduced state changes," United States Patent 4667337, 1987.
2. M. R. Stan and W. P Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on VLSI Systems*, vol. 3, No. 1, pp. 49–58, Mar. 1995.
3. M. R. Stan and W. P Burleson, "Coding a terminated bus for low-power," in *Proceedings of Great Lakes Symposium on VLSI*, Mar. 1995, pp. 70–73.

4. S. Bae, et al., "An 80 nm 4Gb/s/pin 32 bit 512 Mb GDDR4 graphics DRAM with low power and low noise data bus inversion," *IEEE Journal of Solid-State Circuits*, vol. 43, pp.121–131, January 2008.
5. S.-J. Bae, et al., "A 60nm 6Gb/s/pin GDDR5 graphics DRAM with multifaceted clocking and ISI/SSN-reduction techniques," in *International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2008, pp. 278–279.
6. D. Oh, W. Kim, J.-H. Kim, J. Wilson, R. Schmitt, C. Yuan, L. Luo, J. Kizer, J. Eble, and F. Ware, "Study of signal and power integrity challenges in high-speed memory I/O designs using single-ended signaling schemes," presented at the IEC DesignCon, Santa Clara, CA, 2008.
7. J. Wilson, A. Abbasfar, T. Greer, L. Luo, J.-H. Kim, D. Oh, C. Werner, J. Ren, J. Kizer and J. Eble, "Equalization of mid-frequency power supply noise via a spectrum-shaping encoder for parallel buses," in *Proceedings of Electronic Components and Technology Conference*, San Diego, CA, May 2009, pp.1122–1126.
8. H. Lui, H. Shi, X. Jiang, and Z. Li, "Pre-driver PDN SSN, OPD, data encoding, and their impact on SSJ," in *Proceedings of Electronic Components and Technology Conference,* San Diego, CA, May 2009, pp.1127–1131.
9. D. Oh, F. Ware, W. Kim, J.-H. Kim, J. Wilson, L. Luo, J. Kizer, R. Schmitt, C. Yuan, and J. Eble, "Pseudo-differential signaling scheme based on 4b/6b multiwire code," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2008, pp. 29–32.
10. A. Abbafar, "Generalized differential vector signaling," *IEEE International Conference on Communications*, Jun. 2009, pp. 1–5.
11. A. X. Widmer and P. A. Franaszek, "A DC-balanced, partitioned-block, 8b/10b transmission code," *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 440–451, Sep. 1983.

CHAPTER 14

Supply Noise and Jitter Characterization

Hai Lan and Ralf Schmitt

Today's high-speed I/O interfaces, operating at multi-gigabit per second data rates, present unprecedented design challenges [1]. Of all these challenges, achieving very low jitter, in order to meet increasingly tighter timing budgets, is one of the most difficult tasks. Timing jitter can be attributed to several different error sources. The most significant source of timing jitter is power supply noise. Power-integrity engineering has invested considerable effort to provide a stable power distribution network (PDN) that minimizes power noise. However, designing a PDN that makes power noise negligible in a high-speed interface is almost impossible, because the package exhibits a more inductive nature in high-speed systems. Power supply noise directly contributes to the jitter of the system's internal timing sources (for example, voltage-controlled oscillator [VCO], phase-locked loop [PLL], and delay-locked loop [DLL] circuits). It also affects the timing of other circuits, such as the clock distribution and the output driver circuits. Moreover, different circuit components exhibit different jitter sensitivities, or responses, to the supply noise. Therefore, a comprehensive approach to characterizing *power supply noise induced jitter* (PSIJ) is of the utmost importance when designing and optimizing high-speed I/O interfaces. The goal is to establish a methodology that models and verifies the impact of the supply noise on the system timing jitter. To accomplish this, the following questions must be addressed:

- How do we model and simulate power supply noise?
- What is the best metric to characterize the jitter response to supply noise impact?
- How do we analyze and predict the impact of supply noise on jitter?
- Can we correlate the simulation data with the results of the measurements?

This chapter focuses on comprehensive and systematic approaches to the analysis of PSIJ. Section 14.1 reviews the importance of characterizing power noise and its associated jitter.

Section 14.2 provides an overview of the modeling methodology for supply noise induced jitter. Section 14.3 addresses the simulation aspects of the modeling approach (including pre- and post-layout supply-noise-simulation techniques), followed by a methodology for jitter sensitivity extraction, by both conventional and faster means. Section 14.4 presents the correlations between simulation and measurement, for several examples of high-speed I/O interfaces.

14.1 Importance of Supply Noise Induced Jitter

High-speed interfaces require very low random and deterministic jitter to meet ever-shrinking timing budgets. As the data rate increases, the acceptable timing loss due to supply distortion decreases. In general, timing jitter can be reduced in either of two ways: reduce the circuit's sensitivity to supply noise, or reduce the amplitude of the supply noise itself.

A common approach to this problem is to scale the noise budget of the system's supply rails, so that they are inversely proportional to the system's data rate. Assuming that the circuits have a constant sensitivity to supply noise, scaling the supply noise in this way assures that the relative contribution of PSIJ to the total bit time (UI) remains constant. However, this approach makes designing the power distribution systems more difficult. For Gigabit I/O systems, pure linear scaling of the supply noise budget to the data rate would soon lead to unrealistic supply budgets, which cannot be achieved in a system with reasonable package and decoupling resources.

With high-speed interfaces, this problem can be addressed by introducing several independent power supply rails, each with a different noise budget. A separate power rail, with a very tight noise budget, supplies the circuits that control the internal timing of the interface (for example, PLLs and DLLs). This separate power rail is often called an analog supply (VDDA), due to the analog nature of the circuits that it supplies. The tight noise budget reflects the high jitter sensitivity of these circuits to supply noise. Another supply rail (VDDIO) supplies the output drivers and on-chip terminations for the signaling bus. Often, this additional, higher-voltage output circuit supply is needed to drive an external channel. It also helps to mitigate the supply noise problem in the system. The supply noise requirement for this supply rail is usually more relaxed than VDDA, because the jitter sensitivity of output drivers to supply noise is typically small. Finally, a third rail (VDD) with a moderate supply noise budget supplies the remaining circuits in the interface.

There are two major advantages to separating the interface circuits into different groups, supplied by independent supplies: The first is that one can focus the resources of the power distribution system on the supply rails with the tighter supply noise budgets. For example, a larger number of on-chip decoupling capacitors can be assigned to the supply VDDA to achieve a low supply noise level on that rail. The second advantage is that this separation isolates sensitive circuits from other noise sources. For instance, output drivers often excite significant power noise; therefore, allocating separate power rails to the PLL or clock buffers is better.

In the past, many research efforts have focused on deriving power-delivery design requirements, in order to achieve a pre-defined supply noise budget [2] [3]. These activities have

provided many insights into the generation of supply noise, as well as the optimization of a power delivery system in the frequency domain. However, these analyses primarily focus on the supply noise in the system, not on the jitter generated by this supply noise. Little information is available about the exact process used to derive the supply noise budget. With many designs, the supply noise budget is derived from negotiations between the circuit team and the system designer responsible for the power delivery design; they define supply noise levels that are both achievable (with reasonable system resources), and acceptable as a basis for the design and optimization of the circuits. (Chapter 11, "Overview of Power Integrity Engineering," reviews the systematic process of defining a power supply budget.)

For Gigabit I/O systems, the PSIJ requirements are so demanding that they require a more detailed understanding of how jitter is generated in the system. PSIJ is the result of the interaction of two separate and largely independent parameters: the supply noise spectrum generated on each supply rail, and the sensitivity of the circuits supplied by these supply rails to noise at different frequencies. Together, these two parameters define the final jitter impact on the system due to supply noise. Understanding both parameters independently, as well as in combination, provides the insight needed to optimize the jitter performance of the system.

Comparing the total jitter impact created by supply noise on different rails helps identify which supply rails are contributing most to the system jitter, and which should be the subjects of further supply-delivery optimizations. A supply rail with lower noise sensitivity, but with higher supply noise levels, can contribute as much or more jitter to the system than a supply rail with higher noise sensitivity, but with very low supply noise levels. Analyzing both the noise-sensitivity profile and the supply-noise spectrum provides insight into the interaction between both parameters, and directs the optimization of these parameters for optimized system performance. For example, the power delivery system design can provide higher attenuation for noise in a frequency range where the noise sensitivity is known to be great, or the circuit design can be optimized to reduce sensitivity at the frequency where the supply noise is high.

14.2 Overview of PSIJ Modeling Methodology

As mentioned in the previous section, two factors determine the overall jitter impact introduced by the power supply noise. The first factor is the characteristics of the supply noise itself, and the second factor is how the system timing jitter responds to the noise present on its supplies. Depending on the circuit implementation, operation mode, data activity pattern, and power delivery network, the supply noise may exhibit different characteristics in terms of its time-domain waveform signature and its frequency-domain spectral content.

The second factor is solely determined by the circuit implementation and clocking architecture choice; it is independent of the data activity or operation mode. From the system point of view, this process can be modeled with a linear time invariant (LTI) system responding to an input stimulus. Naturally, the most convenient way to characterize such a system is to model both the input stimulus and system transfer function in the frequency domain. A comprehensive and

systematic methodology for analyzing the supply noise impact on jitter can be developed by combining the frequency spectrum of the supply noise and the frequency-dependent jitter sensitivity profile.

Figure 14.1 shows the overall concept of modeling power supply noise induced jitter (PSIJ). The goal is to establish a systematic approach to predict the supply noise impact on jitter. The supply noise spectrum ($V(f)$) is circuit and activity dependent. The jitter sensitivity profile ($S(f)$) is circuit dependent, but data independent. These two factors are largely decoupled, and can be characterized independently. By combining the noise spectrum and jitter sensitivity profile, we can compute the jitter spectrum, as denoted by $J(f)$ in the figure. From the jitter spectrum, the time-domain jitter sequence can be reconstructed using an inverse Fourier transform.

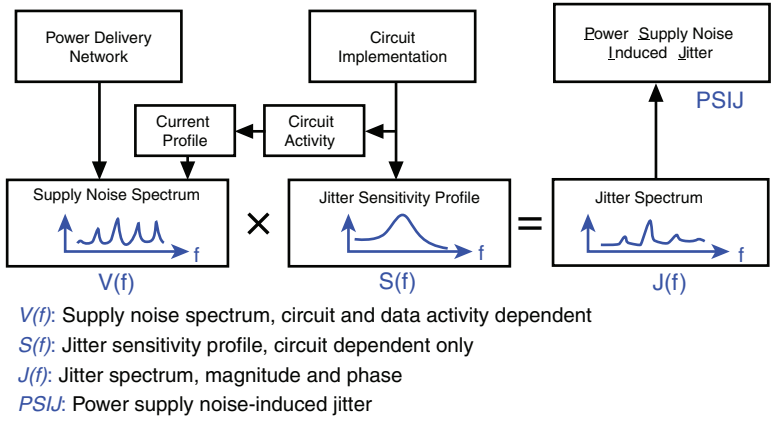


Figure 14.1 Power Supply Noise Induced Jitter (PSIJ) Modeling Methodology

This methodology has two fundamental assumptions: First, the supply noise is sufficiently small that it warrants normal circuit operation. Second, the system jitter response is a linear function of the supply noise. The second assumption is hard to prove in general, and it can be validated in simulation. Although the linearity assumption may not hold true in general, it is still useful for qualitative or first-order analysis.

The noise on the power supply rail of the system is the product of the current spectrum $I(f)$ and the impedance of the power supply network $Z_{PDN}(f)$:

$$V(f) = Z_{PDN}(f) \cdot I(f) \tag{14.1}$$

The simulated current profile, and the supply impedance profile (from the PDN model of the test system), are shown in Figure 14.2. One expects to see the supply noise spectrum feature the same spectral peaks as the original current spectrum, but the background level of the supply noise spectrum follows the profile of the supply impedance $Z_{PDN}(f)$. This effect emphasizes any

noise contributions at medium-frequency, where the supply impedance $Z_{PDN}(f)$ shows large amplitude, due to package-chip resonance.

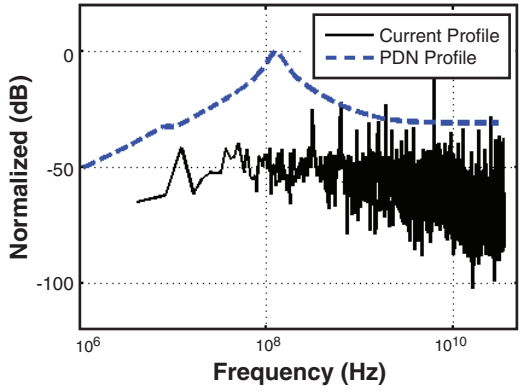


Figure 14.2 Sample Multi-Gigabit I/O System Current and Impedance Profiles

Figure 14.3 shows the validation of the linearity assumption, based on simulation. The linearity assumption is also supported with two measurement observations [4] [5]. First, exciting single-frequency supply noise generates single frequency jitter at the same frequency. Second, jitter amplitude is a linear function of noise amplitude, given that the circuits are operating in the linear region under normal system operating conditions. Based on these observations, one can define the jitter sensitivity parameter as the ratio of the resulting jitter to a 1-mVpp supply noise disturbance at frequency *f*:

$$S(f) = \frac{J(f)}{V(f)} [ps, pp/mV, pp] \tag{14.2}$$

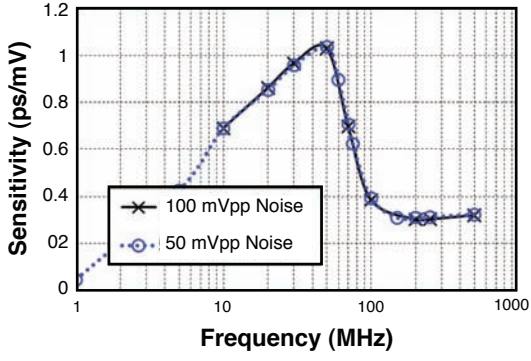


Figure 14.3 Simulation Validation of Linearity of Jitter Sensitivity Concept

The jitter spectrum $J(f)$, as stated in (14.2), serves as a key parameter in characterizing the jitter impact due to supply noise. It can be used in many ways to understand the various aspects of the supply noise induced jitter. First, the spectral profile provides the jitter spectrum frequency contents, and how they are related to the system's operating conditions (for example, reference clock frequency, data rate, operation mode, transaction data pattern, and so on). Second, because the spectral profile contains both magnitude and phase information, they can be used to derive the time domain counterpart of jitter. The worst peak-to-peak jitter J_{pp} is derived as follows:

$$J_{pp} = 2 \times \int_0^\infty J(f) \cdot df \tag{14.3}$$

Often, using the accumulated jitter percentage is also constructive to help identify the biggest jitter contribution components in the frequency domain. The accumulated jitter percentage is defined as follows:

$$\eta = \frac{\int_0^f J(f) \cdot df}{\int_0^\infty J(f) \cdot df} \tag{14.4}$$

The time-domain jitter sequence ($J(f)$) can be reconstructed by performing an inverse Fourier transform of $J(f)$:

$$J(t) = \int_0^\infty J(f) e^{j2\pi f} df \tag{14.5}$$

14.3 Noise and Jitter Simulation Methodology

As discussed earlier, the prediction of power supply induced jitter requires both the supply noise spectrum and jitter sensitivity profile. This section addresses how to simulate supply noise and how to extract jitter sensitivity.

14.3.1 Supply Noise Simulation

Simulating on-chip supply noise requires the modeling and extraction of three components: off-chip power delivery network, on-chip power delivery network, and supply current profiles. Typically, lumped models are used in pre-layout noise analysis to access early stage supply noise and the total inductance and decap budgets. In post-layout noise verification, a distributed on-chip power grid model and distributed temporal current profiles are used.

Figure 14.4 illustrates a typical pre-layout supply-noise-simulation setup, using lumped PDN and current profile components. Typically, the off-chip PDN is modeled with passive RLC components, resulting from voltage regulator, PCB, and package parasitics, as well as low- and medium-frequency decoupling capacitances. The on-chip PDN represents the physical power grids, from the die pads to all over the chip, and typically includes RC parasitics and, very

importantly, on-chip decaps. (Well-established methods exist to extract all the parasitic parameters.) The third component is the current profile, which is the stimulus for the complete PDN.

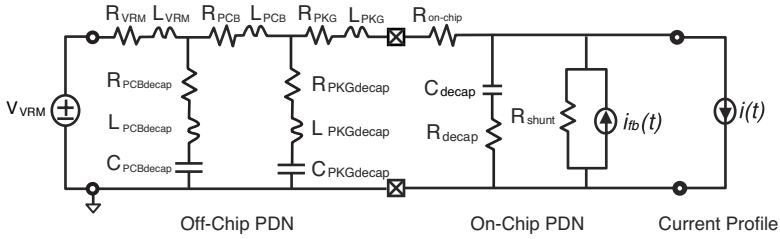


Figure 14.4 Pre-Layout Supply Noise Simulation

The primary design target of a PDN is to deliver an off-chip power supply to on-chip grids, with maximum delivery efficiency and minimal degradation and noise. A complete PDN includes both off-chip and on-chip PDNs, which require different modeling approaches. Typically, an off-chip PDN consists of VRM, PCB, and package parasitics, which are modeled with resistances and inductances. Decoupling capacitances (ESC) on the PCB and package are used to filter the low- and medium-frequency noise in the power supply. ESC parasitics are modeled with ESR and ESL. Figure 14.4 illustrates a typical off-chip PDN.

The on-chip PDN can be modeled in two ways: lumped modeling or distributed modeling. Lumped modeling is suitable for pre-layout supply noise estimation. Although it assumes that the on-chip supply has insignificant spatial dependence, this method can largely capture the overall supply noise characteristics when the circuit die size is small. The middle portion of Figure 14.4 illustrates a simplified, lumped, on-chip model. On-chip power grids are simplified and lumped into $R_{on\text{-}chip}$, which can be estimated using the static IR drop specification and the nominal DC power supply current. All the on-chip decaps are lumped into C_{decap}, which represents the overall capacitive decoupling effects from intentional decaps (MOS or MIM caps), parasitic capacitances (due to non-switching gates in digital circuits), and interconnect parasitic capacitances. The loss associated with C_{decap} is represented by R_{decap}, which is determined by the fabrication process-dependent relaxation time constant as follows:

$$R_{decap} = \tau_{process} / C_{decap} \tag{14.6}$$

To model the effect of supply voltage collapse to the first order, two additional elements, R_{shunt} and $I_{feedback}$, are introduced into the on-chip lumped model, as follows:

$$R_{shunt} = \frac{V_{nom}}{\alpha \cdot I_{DC}} \tag{14.7}$$

$$I_{feedback} = \alpha \cdot I_{DC} \tag{14.8}$$

where α is an empirical factor representing how much current should be deducted from the ideal DC current, in order to model the effective decrease in total current draw demand to be supplied by the PDN. In a real application, the value of α generally ranges from 0.05 to 0.3, depending on the circuit type. The smaller α is, the smaller the effect of the voltage collapse.

In the supply-noise-simulation environment, simulating the entire circuit with a fully extracted on-chip PDN model is usually prohibitively expensive. Therefore, the transient supply currents are usually extracted first, by simulating the circuit under an ideal supply environment. The extracted current profiles then replace the active devices, and equivalently represent the current draw due to circuit activities. Extracting the lumped, top-level current profiles for all the supply domains is often straightforward, and typically involves only probing the current draw from the voltage sources. This type of current profile suits the needs of pre-layout simulation. It can also be used in post-layout simulation, if it is properly distributed among all the current sink nodes.

Although the current profiles are considered to be the sources of the supply noise, note that the supply voltage noise is determined by both the current profile and the impedance profile. Figure 14.2 shows a typical normalized impedance profile, overlaid with a typical current profile from the I/O test system operating at 6.4Gb/s [4]. To minimize the dynamic range of the resulting supply voltage, the design effort should try to avoid overlapping the impedance resonance frequency with any strong components in the current profile.

Despite the ease of use and acceptable accuracy of lumped on-chip models, the distributed on-chip model is preferable for post-layout supply noise simulation. Figure 14.5 shows the post-layout supply-noise-simulation flow. At this stage, full chip power grids are extracted to generate detailed, distributed 2-D RC models. The on-chip decap distribution should also be correctly identified and accurately estimated. Current sink nodes connecting the circuit blocks or transistor terminals must be back-annotated, so that they can be tied to the provided current profiles. To achieve better accuracy in post-layout simulation, detailed current profiles are extracted at the sub-block circuit-level, and even down to the device-level. This is done by first generating current probes from the circuit netlist, which recognize all the nodes connected to the supply or ground. These nodes are then included in the circuit simulations. The resulting current profiles are connected to the back-annotated on-chip PDN at their corresponding physical locations.

14.3.2 Jitter Sensitivity Extraction

A conventional approach to determining the jitter sensitivity profile is to use brute-force transient simulation with any standard circuit simulator (for example, HSPICE). Because we are concerned with sensitive jitter analysis here, using a post-layout SPICE netlist that incorporates RC parasitics and distributed RLC models is preferable (if long, multi-GHz clock wires are used). Usually, a single-tone sinusoidal signal, with small but sufficient amplitude, is injected into the supply node as the stimulus. After the circuit simulation stabilizes, the resulting jitter transient is recorded by observing the output clock TIE (time interval error) measurement.

Figure 14.6 shows an example of the single-tone test for extracting the jitter sensitivity at 50MHz. As shown in the figure, 50MHz noise, of 50mVpp, is excited at the VDD supply after the

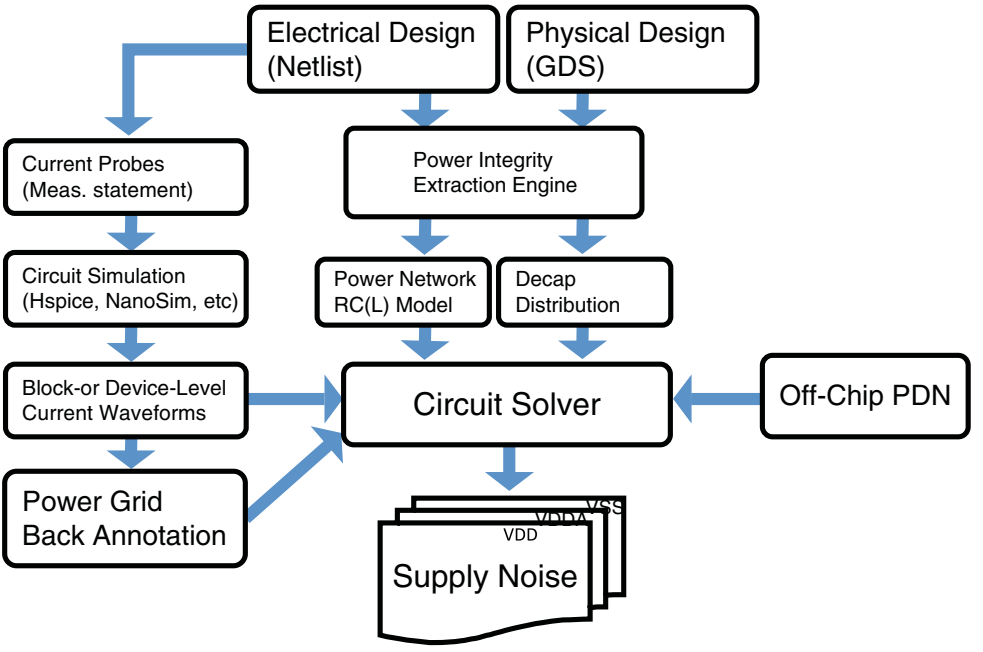


Figure 14.5 Post-Layout Supply Noise Simulation Flow

PLL settles. Because of this noise disturbance, the clock sees a 50MHz jitter, induced by the supply noise. Figure 14.7 shows the spectrum of the supply noise and its induced jitter. The supply noise spectrum shows $V(t) = 25\text{mV} \cdot \cos(2\pi \cdot 5 \cdot 10^7 \cdot t - 90°)$. The jitter spectrum shows that the jitter is $J(t) = 10.96\text{ps} \cdot \cos(2\pi \cdot 5 \cdot 10^7 \cdot t - 57°)$. Therefore, both the magnitude and phase of the jitter sensitivity, at 50MHz, can be determined from the preceding information—in this example, $|S(50\text{MHz})| = 10.96\text{ps}/25\text{mV} = 0.44\text{ps}/\text{mV}$ and $\varphi|S(50\text{MHz})| = -57° - (-90°) = 33°$. The same procedure is repeated at multiple frequency points to obtain the jitter sensitivity profile over frequency.

However, a faster alternative approach exists. Circuits such as PLL and DLL are non-linear in the voltage-domain, but inherently linear in other domains (like the phase/delay). Mapping the voltage variables to/from the variables in linear domain allows one to characterize the circuits by using linear analysis techniques, rather than transient simulations. An efficient approach to extracting the jitter sensitivity profile was recently introduced that exploits a variable domain transformation to map harder non-linear voltage domain problems to easier linear phase domain problems [7]. Figure 14.8 shows the phase transformation application in a PLL characterization.

To apply this method to simulate jitter sensitivity, two variable domain translators were implemented as Verilog-A modules [7]. One is a phase-to-voltage translator, and the other is a

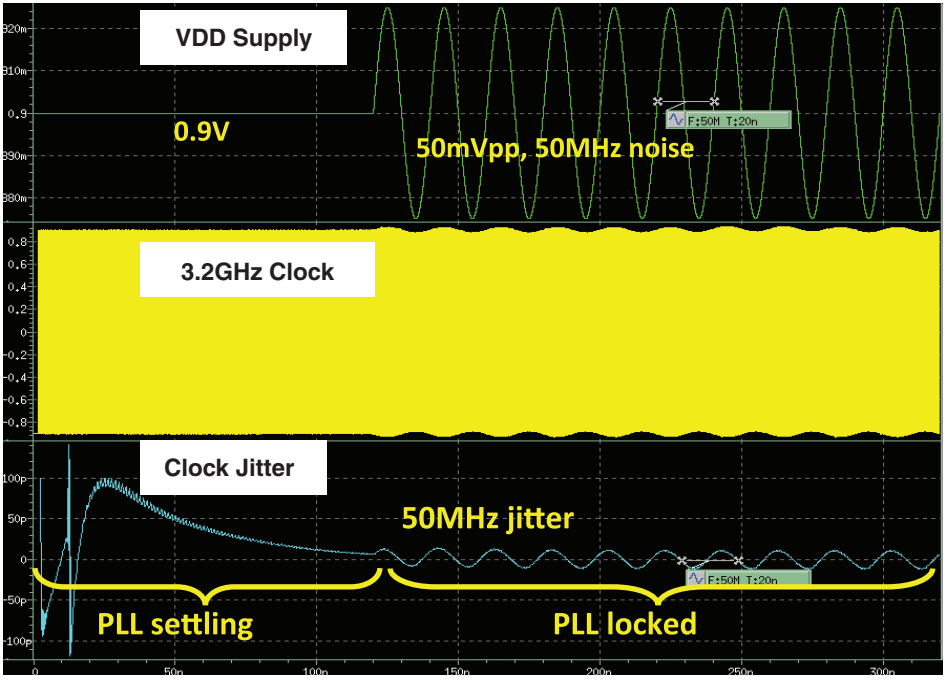


Figure 14.6 Single-Tone Test for Supply Noise Induced Jitter Sensitivity Extraction

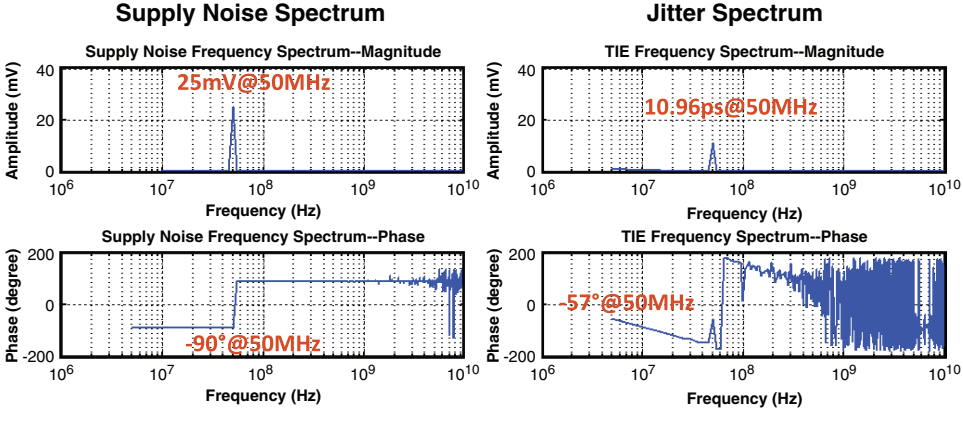


Figure 14.7 Sample of Jitter Sensitivity Extraction using Frequency Spectrum, Supply Noise Spectrum and Jitter Spectrum Showing

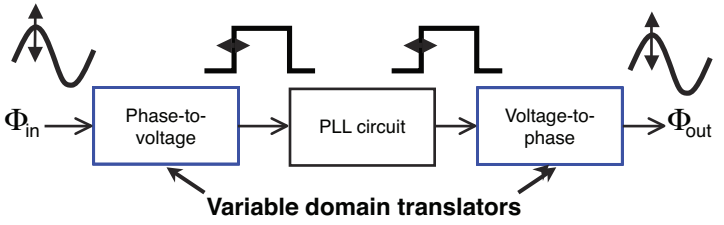


Figure 14.8 Phase Domain Transformation in PLL

voltage-to-phase translator. The phase domain simulation was performed by using Spectre-RF periodic steady state (PSS), and periodic AC (PAC) features. PSS performs large signal analysis to compute the periodic steady-state response of a circuit at a specified fundamental frequency. PAC analysis linearizes the circuit over its PSS response. The impact of small perturbations was obtained by using a linear analysis method, which results in a much faster simulation. Figure 14.9 shows a jitter sensitivity profile simulated by using the PSS/PAC method. It takes about 1 hour to get PSS converged, followed by a few minutes to run PAC, to sweep over the desired frequency range and obtain the final sensitivity profile.

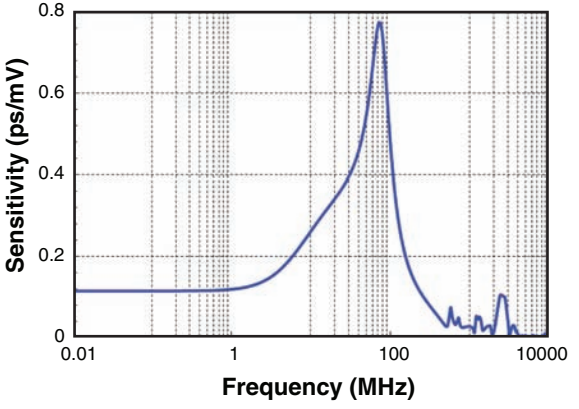


Figure 14.9 Sample of Jitter Sensitivity Simulated by PSS/PAC Method

The transient simulation approach is straightforward to set up and simulate. In most cases, it offers the best accuracy and is regarded as the reference point. However, transient analysis is notably time consuming, and becomes even more so when simulating the jitter sensitivity at relatively low frequencies of about 10MHz. Instead of directly solving the time domain problem, the PSS/PAC-based approach essentially solves a "small-signal" problem around its "DC" operating point, in the periodic domain sense. Therefore, it only requires a minor simulation to sweep over

the frequency range, and get the jitter sensitivity profile. In principle, PSS/PAC promises a much faster solution. However, its main challenge is PSS convergence. Sometimes, even experienced engineers have difficulty getting PSS converged. Figure 14.10 compares these two methods, using a jitter-sensitivity profile extraction for a typical PLL design. A good correlation is achieved in terms of the peaking frequency location and spike.

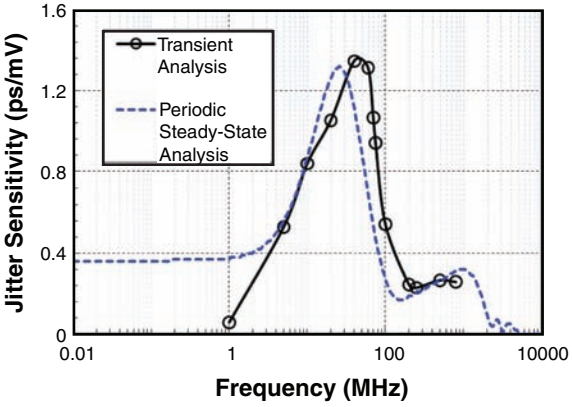


Figure 14.10 Comparison of Two Approaches to Jitter Sensitivity Extraction

14.3.3 Supply Noise Induced Jitter Prediction

The supply noise induced jitter is predicted by applying the following equation:

$$J(t) = \text{IFFT}(J(f)) = \text{IFFT}(V(f) \times S(f)) \tag{14.9}$$

Here, we provide an example of predicting the supply noise induced jitter. The system under consideration is a high-speed I/O interface, operating at 6.4Gb/s. Following the simulation setup illustrated in Figure 14.4, the VDD supply noise is obtained and shown in Figure 14.11. The VDD noise is about 19mVpp, as shown in the figure. To evaluate the jitter induced by this VDD noise, the jitter sensitivity profile must be extracted. The transient simulation-based single-tone test is used to extract the overall jitter sensitivity profile. Figure 14.12 summarizes the PSIJ prediction results. Figure 14.12(a) shows the PSIJ sensitivity profile, including both the magnitude and phase information. It exhibits a band-pass behavior. By combining the PSIJ sensitivity with the VDD noise spectrum, the PSIJ spectrum is derived; see Figure 14.12(b). The major jitter components can be clearly identified in the jitter spectrum. In this particular case, the major jitter components are related to the system reference clock and to sub-harmonics of the data rate. By applying (14.9), the time-domain jitter sequence is computed; see Figure 14.12(c). This is a full characterization of the VDD supply noise impact on the timing jitter. In this case, the 18.9mV peak-to-peak noise, on the VDD supply, ultimately induces about 4.2ps peak-to-peak jitter. The jitter histogram, shown in Figure 14.12(d), can be derived from the jitter sequence.

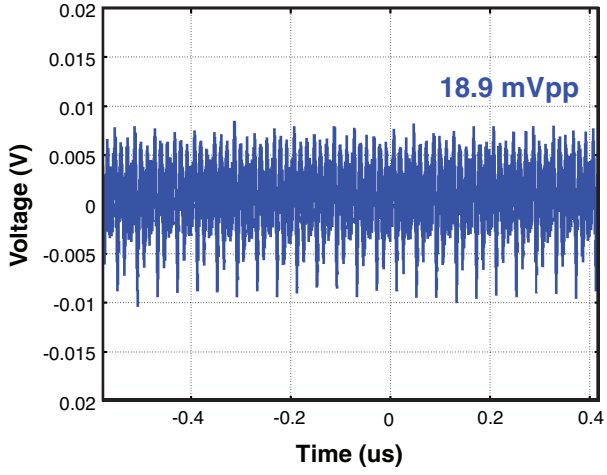


Figure 14.11 Example of Simulated VDD Supply Noise in a 6.4Gb/s I/O Interface

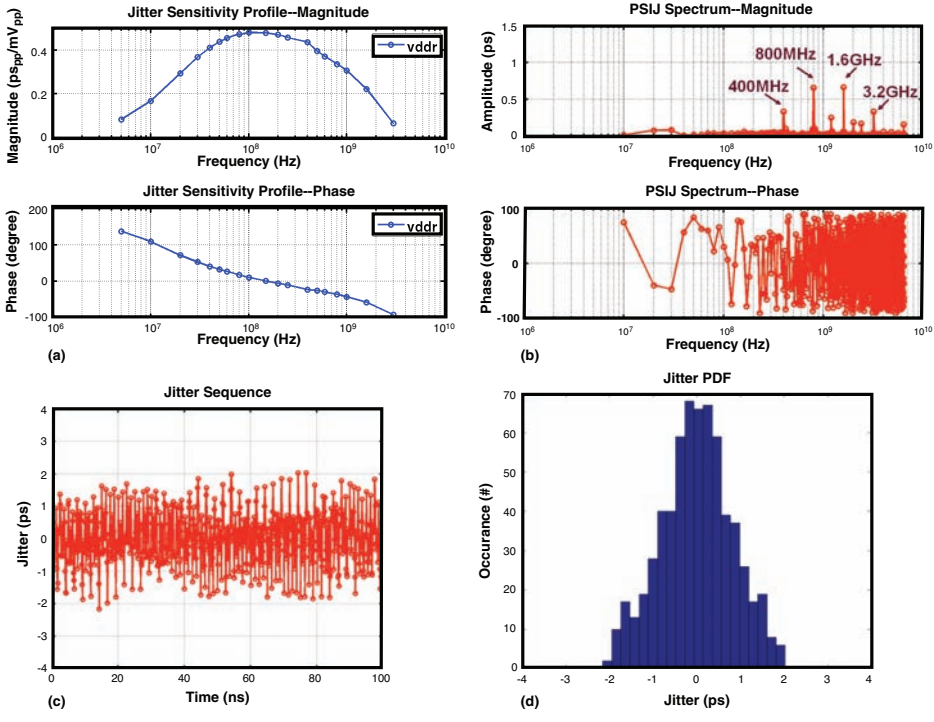


Figure 14.12 Example of Computing PSIJ: (a) PSIJ Sensitivity Profile, (b) PSIJ Spectrum, (c) PSIJ Time-Domain Sequence, and (d) PSIJ Histogram

14.4 Case Study

Two examples are discussed in this section to illustrate the analysis of power supply noise and its jitter impact in high-speed I/O interfaces. The first example discusses the supply noise analysis and jitter sensitivity analysis for a high-speed SerDes system with comparison between simulation results and measurement data [8]. The second example focuses on correlation of the net PSIJ impact between the prediction and measurement data on a high-speed memory controller interface operating at 6.4Gbps.

14.4.1 Supply Noise in High-Speed SerDes System

This section provides an analysis of the on-chip power-supply noise characteristics using the test system, shown in Figure 14.13. The interface is a serial-to-serial transceiver delivering high bandwidth point-to-point interconnections across system backplanes at data rates of up to 6.4Gb/s. It integrates eight bi-directional serial links, each operating at a data rate of up to 6.4Gb/s. Additionally, it contains Data Route Logic to control the data flow traffic between the different links, and Control Logic, which contains general logic for the initialization of the links, as well as logic used for characterization (for example, pattern generators). Four separate on-chip supply domains are used for different circuit blocks inside the test system. The sensitive timing-control circuits (for example, PLLs) are supplied by VDDA. All the circuits inside the eight link slices are supplied by VDD, except for the channel termination resistors, which are supplied by VTT. This case study focuses on VDD and VDDA, because they are the major sources of system jitter. The on-chip supply-noise-measurement circuits are integrated onto the interface test chip for the purposes of correlation. Chapter 16, "On-Chip Link Measurement Techniques," covers the details of these circuits.

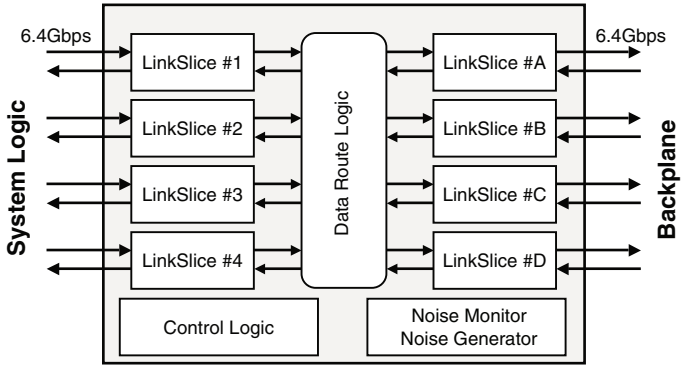


Figure 14.13 Example of Multi-Gigabit/sec SerDes System

The pre- and post-layout simulation methods, previously described, are applied to one of the link slices in the I/O interface test system, to predict the on-chip supply noise in the VDD and

VDDA domains. The link slice under test was configured to continuously transmit a PRBS7 data pattern bit stream at 6.4Gb/s, with the PLL reference clock at 640MHz. The same conditions are then applied to the test system in the lab. The on-chip supply-noise-measurement monitors are used to capture the VDD and VDDA supply noise transient waveforms in real-time sampling scope mode, and their frequency spectrums in autocorrelation mode. The off-chip PDN models are extracted using EM simulation. The lumped-model of the on-chip PDN, for pre-layout simulation, is determined based on design knowledge, process information, and the static IR drop target. The distributed on-chip power grid and decaps, for post-layout simulation, are extracted by a commercially available tool [6]. The current profiles are extracted by probing the top-level supply currents during a full chip transistor-level circuit simulation. Using the earlier PDN model and current profiles, pre-layout simulation are performed using HSPICE, and perform post-layout simulation using the tool in reference [6]. Figure 14.14 summarizes the final post-layout simulation results: The first plot shows the detailed device-level current distribution, the on-chip decap distribution is shown in the second plot, and the third plot illustrates the supply noise map.

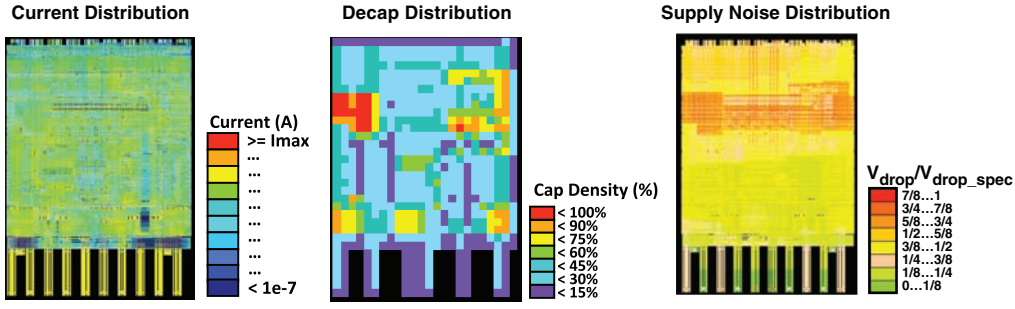


Figure 14.14 Post-Layout Simulation Results for a High-Speed SerDes Test System, Current Distribution, Decap Distribution, and Supply Noise Distribution

Figures 14.15 and 14.16 compare the pre- and post-layout simulation results with the measurement data for VDD and VDDA, respectively. As shown in Figure 14.15, the time-domain peak-to-peak VDD supply noise from pre-layout simulation, post-layout simulation, and measurement are 42mVpp, 37mVpp, and 45mVpp, respectively. Both simulation results are in line with the measured result. More insight can be gained by looking at the frequency spectrum. All noise spectra indicate that the strong frequency components are at the data rate frequency, as well as its sub-harmonics, for VDD. This reflects the fact that the VDD current profile is shaped by the VDD PDN impedance profile, which tends to emphasize more at the 100MHz–300MHz PDN resonance range. Figure 14.16 shows that the time-domain VDDA peak-to-peak noise, from the two simulation methods and the measurement data, all agree with each other (12mV, 16mV, and 18mV, respectively). Also comparing the frequency spectra, the measurement data clearly shows a spike at 640MHz. The pre-layout simulation also shows a strong presence at this frequency. The post-layout simulation, however, shows a hump at around 640MHz.

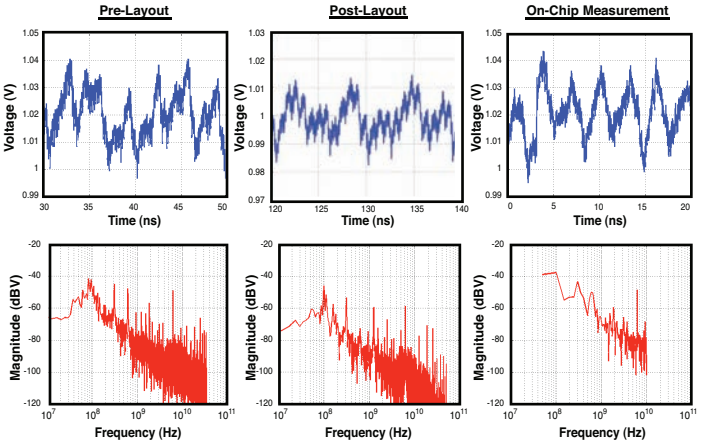


Figure 14.15 VDD Supply Noise in a 6.4Gb/s SerDes Test System: Pre-Layout Simulation Results, Post-Layout Simulation Results, and On-Chip Measurement Results

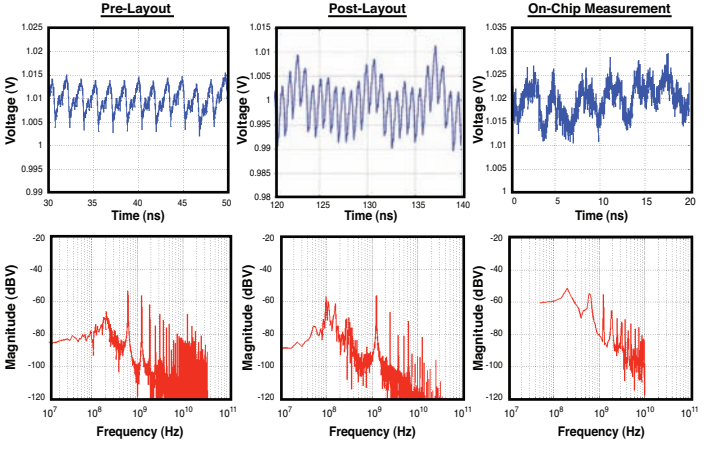


Figure 14.16 VDDA Supply Noise in a 6.4Gb/s SerDes Test System: Pre-Layout Simulation Results, Post-Layout Simulation Results, and On-Chip Measurement Results

In this case study, the on-chip power supply noise of a high-speed serial link I/O system, operating at 6.4Gb/s, is analyzed in terms of simulation and measurement. Supply noise simulation, using a lumped on-chip power grid model, is suitable for pre-layout design verification and optimization. Post-layout simulation is preferred for sign-off accuracy. The extracted distributed power grid network, in combination with externally extracted current profiles, provides further spatial resolution and the best accuracy. The simulated supply noise correlates with the measured results in both the time and frequency domains. Under the representative data pattern transmission

environment, the test I/O interface exhibits 37mVpp–45mVpp dynamic drop in VDD, and ~20mV in VDDA. Strong noise components are related to the reference clock and the data rate.

14.4.2 Jitter Sensitivity in High-Speed Memory Controller PHY

The next case study uses a high-performance memory controller PHY interface test system, operating at 6.4Gb/s–12.8Gb/s. The measurement data is compared with a simulated jitter profile, based on a brute-force transient simulation. Figure 14.17 illustrates the final comparison, between the measured and simulated jitter profile in the test system. The measurement data points generally match the simulation curve. A few outliers exist, which are justifiably removed. These outliers mainly result from the higher-than-expected background noise level present in the noise monitors. In general, both profiles suggest band-pass behavior for the supply jitter sensitivity. The most sensitive frequency range centers around 50MHz–60MHz. The sensitivity is low at both low frequency and high frequency. This band-pass behavior is expected, because it is closely related to the PLL loop bandwidth configuration in the test system. The correlation further verifies that the simulation methodology can very reasonably predict the system jitter behavior.

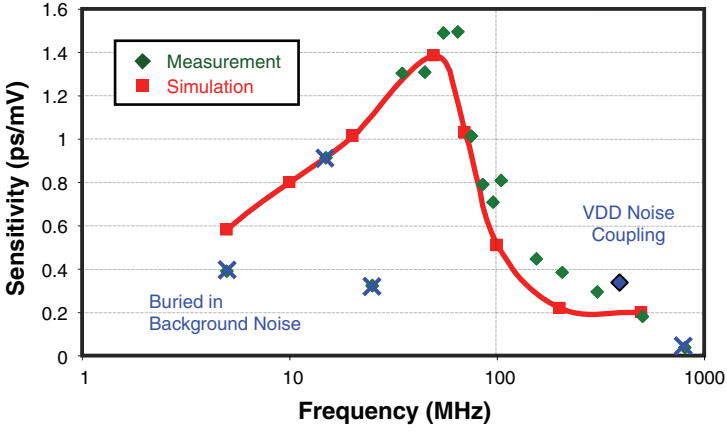


Figure 14.17 Measurement and Simulation Results for VDDA Jitter Sensitivity

14.4.3 PSIJ Prediction in High-Speed Memory Controller PHY

In this last example, the PSIJ prediction and measurement is performed for a high-speed memory controller PHY I/O interface, with all 32 data links running at 6.4Gb/s with a PRBS7 data pattern. The noise on each of the supply rails contributes to the induced jitter. Figure 14.18 shows the final PSIJ results. The overall predicted PSIJ is about 5.8ps, and the overall measured PSIJ is about 7.2ps. In terms of the jitter spectral contents, the predicted PSIJ spectrum largely captures the major jitter components at 6.4GHz, 3.2GHz, and 800MHz, as confirmed by the measurement data.

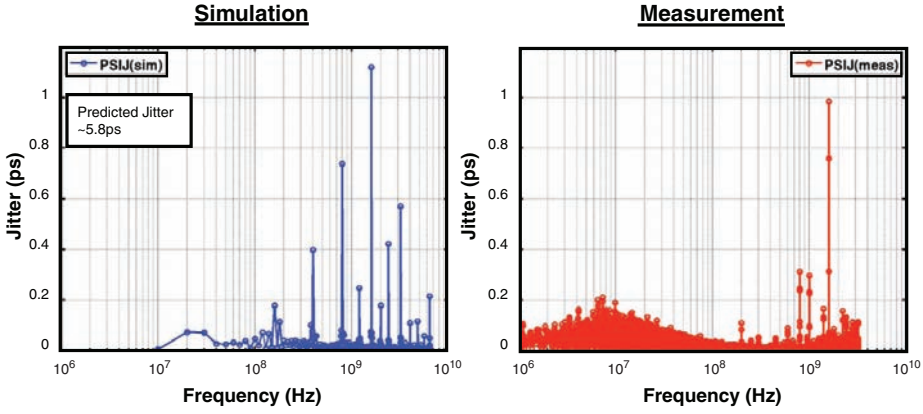


Figure 14.18 Overall PSIJ Spectrum in Test System, Prediction and Measurement

14.5 Summary

Characterization of power supply noise, and its impact on system timing jitter, is one of the most critical tasks in budgeting, designing, and analyzing high-speed interfaces. This chapter discusses a systematic approach to addressing this design challenge, which includes modeling methodology and simulation techniques. (Chapter 16 covers advanced on-chip measurement techniques for characterizing supply noise, power distribution impedance, and jitter sensitivity.)

Off- and on-chip PDN models, and on-chip supply current profiles, are required to simulate the supply noise and its frequency spectrum. Depending on the current stage of the design, a lumped on-chip PDN model, in combination with a lumped supply current, is suitable for pre-layout design analysis. Usually, the pre-layout simulation captures the majority of the overall supply noise characteristics early in the design. However, the post-layout simulation is required for sign-off analysis. During the post-layout simulation, we extract the distributed power grid network, as well as extract and back-annotate the block- or device-level current profiles. PSIJ sensitivity is an important parameter that describes the system jitter response to the supply noise as a function of frequency. The jitter spectrum is obtained by combining the supply-noise frequency spectrum and the PSIJ sensitivity profile. The resulting jitter spectrum reveals strong frequency components where supply noise contributes the most to jitter degradation. It can be also used to derive the time-domain jitter sequence, and statistical-domain jitter characteristics.

References

1. K. Chang, H. Lee, J. Chun, T. Wu, T. J. Chin, K. Kaviani, J. Shen, X. Shi, W. Beyene, Y. Frans, B. Leibowitz, N. Nguyen, F. Quan, J. Zerbe, R. Perego, and F. Assaderaghi, "A 16Gb/s/link, 64GB/s bidirectional asymmetric memory interface cell," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 18–20, 2008, pp. 126–127.
2. O. Mandhana, "Optimizing the output impedance of a power delivery network for microprocessor systems," in *Proceedings of Electronic Components and Technology Conference*, Jun. 2004, Las Vegas, NV, pp 1975–1982.
3. S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation," in *Proceedings of Design Automation Conference*, Jun. 2000, pp. 156–161.
4. R. Schmitt, H. Lan, C. Madden, and C. Yuan, "Analysis of supply noise induced jitter in Gigabit I/O interfaces," presented at the IEC DesignCon, Santa Clara, CA, 2007.
5. R. Schmitt, H. Lan, C. Madden, and C. Yuan, "Investigating the impact of supply noise on the jitter in gigabit I/O interfaces," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 28–31, 2007, pp. 189–192.
6. *RedHawk*, Apache Design Solutions, Inc., San Jose, CA.
7. J. Kim, K. Jones, and M. Horowitz, "Variable domain transformation for linear PAC analysis of mixed-signal systems," in *Proceedings of International Conference on Computer-Aided Design*, Nov. 2007, pp. 887–894.
8. H. Lan, R. Schmitt, and C. Yuan, "Simulation and measurement of on-chip supply noise in multi-gigabit I/O interfaces," in *Proceedings of International Symposium on Quality Electronic Design Conference*, Mar. 2009, 660–675.

This page intentionally left blank

CHAPTER 15

Substrate Noise Induced Jitter

Hai Lan

It has been widely acknowledged that the substrate coupling noise is one of the bottleneck issues preventing the smooth and successful integration of the sensitive analog and RF circuits, with the noisy digital blocks, in a System-on-Chip (SOC) [1]. When sensitive circuits are integrated with noisy digital circuits sharing a common substrate, substrate noise is inevitable, primarily due to the switching activity of the digital circuits. The injected noise propagates through the entire substrate. This noise attenuates differently, depending on the substrate type, doping profile, backside epoxy choice, and layout implementation, but it eventually arrives at the sensitive circuit blocks. In a 3-D through-silicon-via (TSV) environment, the substrate noise can also couple from the TSV structure into the substrate [2]. Substrate noise affects the performance of sensitive circuits by disturbing the device bulk terminal voltages, coupling noise into the power grids (via their ties to the substrate), and sometimes, directly coupling to the signal lines to corrupt the signals.

In recent years, similar concerns regarding the impact of substrate noise on system timing jitter have appeared in the field of high-speed I/O interface design. In high-speed I/O interfaces, the switching activity in the digital core (ASIC core or memory core) can generate strong substrate noise. This noise impacts the sensitive circuits inside the I/O interface; specifically, it degrades the jitter performance of the timing circuits (for example, voltage-controlled oscillator (VCO), phase-locked loop (PLL), delay-locked loop (DLL), and clock distribution). The increasing degradation of the relative jitter performance has been recognized as one of the most critical issues limiting the data rates in modern high-speed I/O interfaces. Chapter 14, "Supply Noise and Jitter Characterization," describes *power supply noise induced jitter* (PSIJ) as one of the dominant timing error components [3] [4]. As I/O interfaces move toward higher data rates and lower power consumption, the impact of substrate noise on jitter performance is becoming more visible. Therefore, understanding the characteristics of substrate noise and its effect on timing jitter is important for high-speed I/O designers. A comprehensive approach to characterizing

substrate noise induced jitter (SNIJ) is becoming ever more crucial in designing and optimizing high-speed I/O interfaces.

This chapter covers substrate noise, and its impact, from both the modeling and the measurement points of view. The goal of the modeling is to answer the following questions:

- What are the basic characteristics of substrate noise?
- How do we efficiently model a substrate-coupling network?
- How do we quantify substrate noise, preferably using an on-chip measurement structure, in high-speed I/O interfaces?
- What is the net jitter impact induced by substrate noise in I/O interfaces?

Section 15.1 presents an overview of substrate noise coupling. Section 15.2 discusses the substrate modeling techniques, targeted at different frequency ranges (including a generic scalable macro model suitable for the low- to medium-frequency region, and a CAD-oriented frequency-dependent equivalent circuit model suitable for high-frequency applications). Section 15.3 defines the on-chip noise measurement requirements, followed by descriptions of proposed noise monitor and generator circuits. (Chapter 16, "On-Chip Link Measurement Techniques," describes the circuit techniques used for on-chip supply noise measurement, as well as the modifications required for substrate noise measurements.) Section 15.4 presents a case study, including the measurement results using the on-chip noise measurement structure, implemented in a low-power memory controller PHY interface test-chip environment. Additionally, the chapter discusses the impact of the measured substrate noise on the link jitter performance, in concert with the introduction of the jitter sensitivity concept.

15.1 Introduction

Figure 15.1 illustrates an overview of the substrate noise-coupling phenomenon (including substrate noise generation, propagation, and coupling processes) in an SoC environment containing a noisy digital core and a sensitive high-speed I/O PHY. The three major noise generation mechanisms are the following:

- Impact ionization due to hot electron effect
- Capacitive noise coupling through reverse-biased source/drain-bulk junction capacitances, well capacitances, and channel capacitances
- Supply noise injection via substrate and well ties

As technology scaling advances, impact ionization becomes negligible when compared to the other two noise sources [1]. The source/drain junction capacitive coupling is more significant than impact ionization, but does not have the impact of direct supply noise injection at the power and ground contacts on the substrate. Therefore, the most dominant noise source is power supply noise. Figure 15.2 illustrates the representative waveform signatures of these two substrate noise

sources. The figure indicates that substrate noise is largely cyclo-stationary (that is, statistical properties are repeating cyclically).

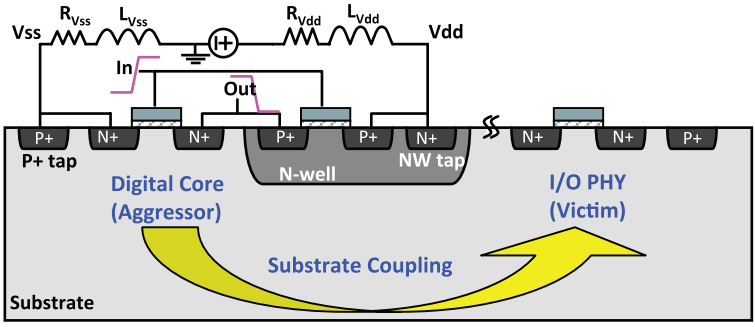


Figure 15.1 Substrate Noise Coupling and its Impact on an I/O Interface

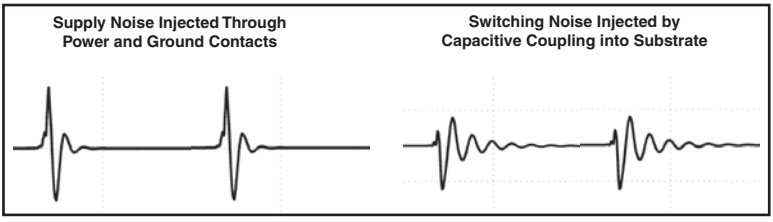


Figure 15.2 Representative Waveforms of Major Sources of Substrate Noise: Supply Noise Injected Through Power and Ground Contacts and Switching Noise Injected by Capacitive Coupling into Substrate

The substrate noise seeks a propagation path with minimum impedance, and once injected into the substrate, it propagates through it. Depending on different substrate types, doping profiles, and frequencies of interest, the noise sees the substrate primarily as a resistive (and sometimes resistive and capacitive) medium. To model the substrate accurately, including the substrate noise coupling effects in the mixed-signal design process is crucial.

Considerable research effort has been invested in characterizing and modeling the substrate. This research has evolved into two major schools of methodology for modeling the substrate: *mesh-based numerical approaches* and *macro-based compact modeling*.

Mesh-based numerical approaches rely on fine 3-D grid meshing schemes to isolate the substrate medium and contacts, in order to solve the Poisson and continuity equations, quasi-static Laplace equations, or full-wave Maxwell equations with suitable boundary conditions. Many mesh-based numerical extraction methods using the finite difference method (FDM), finite element method (FEM), or boundary element method (BEM), have been proposed [1], [5–9].

These methods are accurate, but they are computationally expensive, and normally are limited to simple configurations.

The macro-based compact modeling method seeks to develop macro models in a scalable and compact form. It has the advantage of fast model generation, which allows practical and efficient substrate extraction for large-scale problems. This modeling method was first applied to heavily doped substrates [10–12]. However, having compact macro models for lightly doped substrates is important, as well. Consequently, a surface potential-based model [13] is proposed for modeling layered lightly doped substrates. However, this model requires discretization of the rectangular contacts into circular meshes, and empirical parameter fitting, which is partially geometry-dependent. A more attractive macro modeling approach is one that is truly scalable, using layout geometry with a limited set of process-dependent fitting coefficients, without requiring 3-D meshing on the substrate. We need to formulate and develop a modeling methodology that efficiently generates a substrate network for noise analysis, and provides scalability with layout geometries. Such a methodology will provide insightful information, which we can use to reduce re-layout efforts and facilitate noise-aware layout synthesis. The following section presents a synthesized compact modeling (SCM) approach [14], which addresses the aforementioned modeling challenges.

15.2 Modeling Techniques

The efficient modeling of substrate depends on the substrate material property, doping profile, and frequency of interest. Figure 15.3 qualitatively illustrates the three fundamental operating modes of a silicon substrate: dielectric quasi-static mode, skin effect mode, and slow-wave mode. When the frequency product is so low that it generates a negligible dielectric loss angle, the substrate can be treated as a dielectric. In this mode, the quasi-static characterization is sufficient to model the substrate. On the other hand, when the frequency-conductivity product is large enough that the eddy current can be generated in the silicon substrate as a result of the penetrating magnetic field, the substrate layer becomes lossy. In this mode, the electric fields, and thus the current flow lines, tend to crowd in the skin depth region in the substrate. When the frequency-conductivity stays in the intermediate region (for example, when the frequency is only modestly high, and the silicon substrate conductivity is moderate), the propagation velocity slows down, owing to the energy transfer across the interface associated with the dielectric dispersion and strong interfacial polarization in the substrate. This results in the slow-wave propagation mode. For silicon substrate to operate in any of these modes, the conductance is related to the capacitance by the dielectric relaxation time constant: $C_{si}/G_{si} = \varepsilon_{si}/\sigma_{si}$.

For the heavily doped processes typically used for digital designs, the silicon substrate is treated as a purely resistive medium for frequencies below approximately 2GHz. For the lightly doped processes typically used for analog, RF, and mixed-signal designs, the silicon substrate is modeled as a purely resistive medium for frequencies below approximately 10GHz. The substrate exhibits both resistive and capacitive behavior [15] for frequencies above this range. In

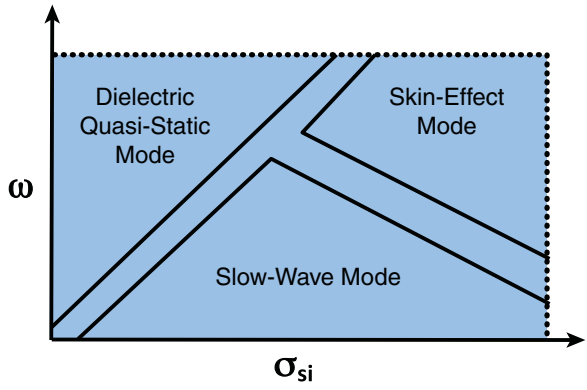


Figure 15.3 Frequency-Conductivity Chart for Silicon Substrate Operating Modes

general, substrate noise has wideband spectral contents, but typically, the major frequency components are within the frequency ranges mentioned previously. Therefore, for low- to medium-frequency applications, modeling the substrate as a resistive network is sufficient. There is a small loss in accuracy, but the model's complexity is significantly reduced. Section 15.2.1 discusses the resistive substrate modeling technique with the focus on the synthesized compact modeling (SCM) approach. For high-frequency applications, the substrate starts to exhibit significant frequency-dependent behavior. Therefore, modeling the substrate as a resistive and capacitive network is necessary. Section 15.2.2 quantifies when the substrate starts to show capacitive behavior, using rigorous 3-D device simulations, and describes a CAD-oriented equivalent circuit modeling approach that is suitable for high-frequency applications.

15.2.1 Low- to Medium-Frequency Substrate Modeling

Figure 15.4 illustrates the cross section of a heavily doped substrate with a lightly doped epitaxial layer on top of it. This type of substrate features thick bulk silicon with very low resistivity (~0.1-0.01$\Omega \cdot$ cm), and a thin epitaxial layer with high resistivity (~10-100$\Omega \cdot$ cm). This type of substrate is typically used in digital circuit design, because its low bulk resistivity provides better latch-up prevention. The entire bulk substrate can be treated as a single, lumped node. Typically, the backplane of the bulk substrate is grounded so the backplane serves as the common ground node for all the substrate coupling ports.

For multiple substrate contact configurations, the substrate-coupling network can be represented by a Z-matrix as:

$$\mathbf{Z} = \begin{bmatrix} Z_{11} & \cdots & Z_{1N} \\ \vdots & \ddots & \vdots \\ Z_{N1} & \cdots & Z_{NN} \end{bmatrix} \tag{15.1}$$

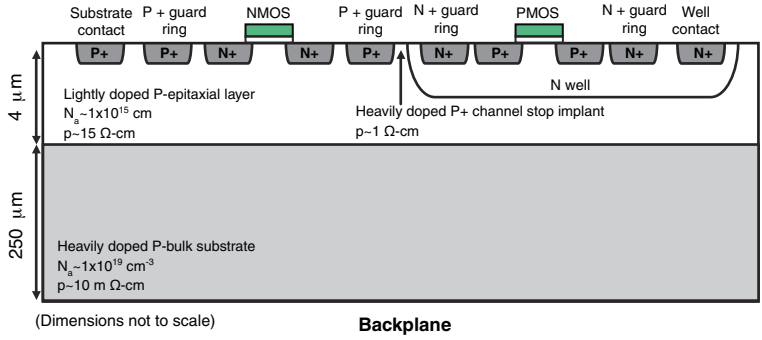


Figure 15.4 Heavily Doped Substrate with Lightly Doped Epitaxial Layer

where all the self- and mutual-impedance terms can be modeled, using the geometry-dependent analytical expressions originally developed by Ozis, Fiez, and Mayaram [12], and later improved for accurate modeling of near-field coupling by Lan et al. [14]. The self-impedance term Z_{ii} is expressed as follows:

$$Z_{ii} = \frac{1}{k_1 \cdot area + k_2 \cdot perimeter + k_3} \tag{15.2}$$

where k_1, k_2, and k_3 are fitting coefficients that are only process-dependent. The mutual-impedance term Z_{ij} is obtained using the following analytical expression:

$$Z_{ij} = \begin{cases} Z_0 e^{-\beta x} & \text{for far field} \\ Z_{01} e^{-\gamma_1 x_1} + Z_{02} e^{-\gamma_1 x_2} & \text{for near field} \end{cases} \tag{15.3}$$

where β, γ_1, and γ_2 are process-dependent fitting coefficients, and Z_0, Z_{01}, and Z_{02} are derived using the self-impedance formula in (15.2).

Figure 15.5 shows the cross-sectional view of a lightly doped substrate, which is widely used in analog, RF, and mixed-signal circuits, due to less eddy current and better noise isolation in the substrate. It features a bulk silicon substrate with a high resistivity of approximately 10Ω·cm. The nature of this highly resistive bulk substrate prevents it from being modeled as a single lumped node. Unlike the example of the heavily doped processes, there is no common ground for all the substrate-coupling ports. Therefore, a resistance-based formulation is used to develop the SCM model for lightly doped substrate processes. The substrate's noise-decay behavior is almost exponential in the near-field region. As the separation further increases into the far-field region, the coupling does not decay indefinitely to zero. Instead, it tends to approach

a saturation level. The following closed-form expression incorporates the aforementioned behavior into one formulation to calculate the coupling resistance between two substrate contacts:

$$R_{ij} = \beta \cdot [\ln(d_{ij} + 1)^{\alpha_1} \cdot (s_i + s_j)^{\alpha_2} \cdot (p_i + p_j)^{\alpha_3}] \tag{15.4}$$

where d_{ij} is the geometric mean distance (GMD) between the two contacts, s_i and s_j are the areas of contacts i and j, and p_i and p_j are the perimeters of contacts i and j. The terms β, α_1, α_2, and α_3 are fitting coefficients; they characterize the substrate coupling resistance's dependence on the substrate doping profiles, separation between two contacts, contact sizes, and contact perimeters, respectively. These four coefficients are independent on the layout geometry, and only process-dependent.

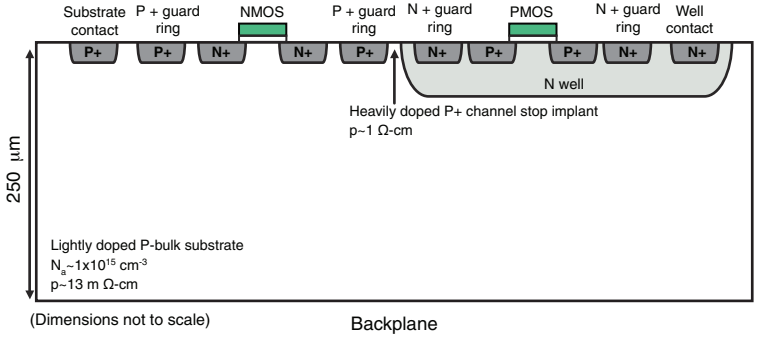


Figure 15.5 Example of a Lightly Doped Substrate

These modeling parameters are obtained from 3-D device simulations, and are then calibrate using lab measurements from the test chip [14]. First, a set of benchmark substrate-test structures are designed. They typically consist of various substrate contacts with different sizes and distances. Second, these test structures are characterized with detailed semiconductor device simulations. Although often difficult, configuring the proper silicon substrate doping profile in the simulation setup is essential. Due to the near-field substrate coupling effects, 3-D device simulators (for example, Davinci) are recommended to accurately capture the strong coupling in the near-field region. Next, the substrate coupling resistances is extracted and the results are used to fit the modeling coefficients. In the last step, the coefficients are calibrated, using the test chip as well as including the previously described benchmark test structures. This last step is optional, yet strongly recommended, because having an accurate substrate doping profile for the process is often difficult.

Figure 15.6 compares the SCM modeling results with the measurement data for a variety of contact geometries. The first two plots show that the substrate coupling trends, over distance, exhibit similar behavior: Coupling increases more than linearly or exponentially for smaller separations, and continues to increase in a linear fashion with very slow slope for larger separations. This confirms the near-field and far-field effects. The last two plots show the coupling resistance between two contacts of arbitrary sizes, and with different offsets.

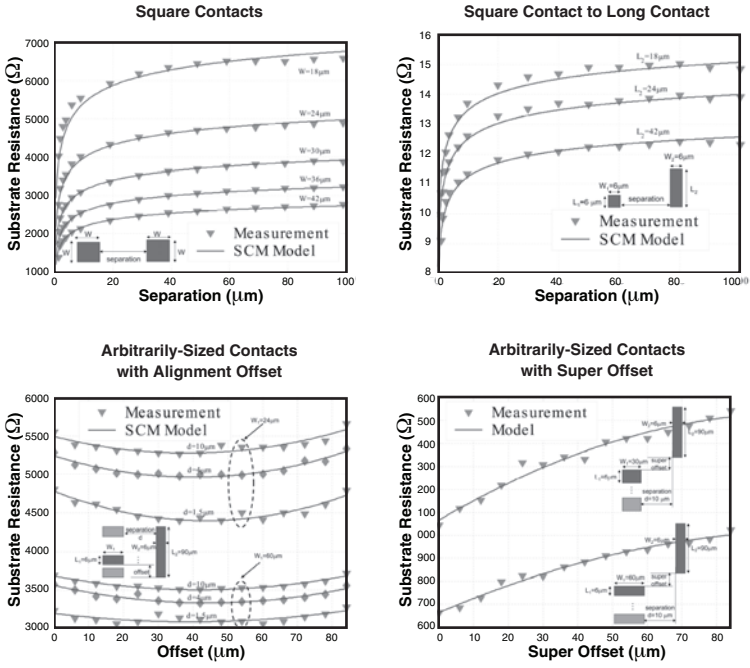


Figure 15.6 SCM Model and Measurement Data for a Custom Lightly Doped Process

A *guard ring* is one of the most common layout techniques used to mitigate substrate noise. This section examines the guard ring's effectiveness, and its sensitivity to the layout geometry. Figure 15.7 shows both the SCM modeling results and the measured data. The first plot illustrates the noise isolation sensitivity of the guard ring's width. As shown in the plot, the voltage noise isolation is roughly proportional to the guard ring width. In this particular example, for every 2μm increase in the guard ring width, a gain of about 2dB more noise isolation occurs. The second plot shows how the positioning of a noise victim inside a guard ring can affect the noise isolation. As the plot shows, the noise isolation degrades as the victim moves from position 1 (close to the corner of the guard ring and far from the aggressor) towards position 4 (centered inside the guard ring). This behavior is expected because the major factor is the decreasing distance between the victim and aggressor. More importantly, note how the noise isolation is actually enhanced by 2dB, as the victim continues to move from position 4 to position 7 (close to the corner of the guard ring and the aggressor). During this process, two competing factors exist: the decreasing distance between the aggressor and the victim, and the decreasing distance between the victim and the guard ring wall. This observation provides the design trade-off when determining the placement of a sensitive circuit inside the guard ring.

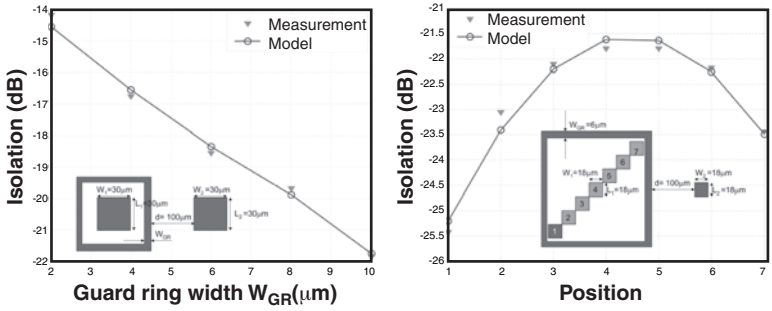


Figure 15.7 Noise-Isolation Sensitivity of Guard Ring Width and Victim Location inside Guard Ring

Figure 15.8 shows the representative substrate-resistance results for a 0.18μm BiCMOS lightly doped process, generated by both SCM and by measurement. The results verify the substrate coupling decay trend observed in the lightly doped process; the decay increases steeply when the distance between the victim and aggressor is small and then starts to slow down when the distance is large. The SCM modeling agrees with the measurement data. The average error is about 10%, with a maximum error of 15%.

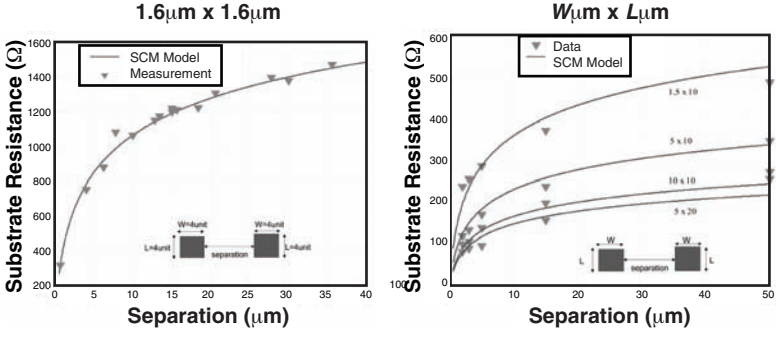


Figure 15.8 SCM Model and Measured Data for a 0.18μm BiCMOS Lightly Doped Substrate Process: Contacts of 1.6μm×1.6μm and *W*μm×*L*μm

15.2.2 High-Frequency Substrate Modeling

A Y-parameter based macro-model is used to develop a high-frequency behavior model. This model synthesizes an equivalent circuit model, in terms of ideal lumped RC elements, by constructing a rational formulation from the frequency-dependent Y-parameter. Simulation study shows that a second-order polynomial approximation is sufficient to model the silicon substrate's

frequency response up to 100GHz [15]. Figure 15.9 shows the equivalent circuit model, synthesized using the Y-parameter. This section examines two cases to demonstrate the validity of this methodology: one for a heavily doped substrate, and the other for a lightly doped substrate.

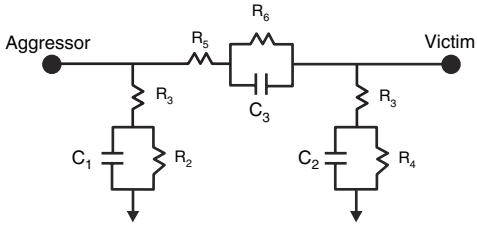


Figure 15.9 Equivalent Circuit Model Synthesized from Y-Parameters

Figure 15.10(a) illustrates the case of a heavily doped substrate with one aggressor, one victim, and one biasing contact. Figure 15.10(b) shows the self and mutual conductance and capacitance simulated by a 3-D device simulator. Note that the conductance and capacitances remain constant for frequencies up to 10GHz, and then start to show significant frequency dependence beyond 10GHz. Figure 15.10(c) and (d) compares the equivalent circuit modeling results for Y_{11}, and Y_{12}, respectively, with the device simulation data. A good match exists between the circuit models and the device simulation data. The synthesized circuit model characterizes the frequency-dependent behavior very accurately, with very low computational cost, as compared to the device simulation. It is observed that the purely resistive substrate model is valid up to around 1GHz–2GHz, beyond which the imaginary part of the Y-parameter becomes more and more comparable to the real part.

Figure 15.11(a) is an example of a lightly doped substrate with one aggressor, one victim, and one biasing contact. Figure 15.11(b) shows the self- and mutual-conductance/capacitance simulated by the 3-D device simulator. Due to the less lossy nature of the lightly doped substrate, the conductance and capacitance stays constant over a very wide frequency range. One observes only slight frequency-dependence at frequencies above approximately 90GHz. The Figure 15.11(c) and Figure 15.11(d) comparison of the equivalent circuit modeling results for Y_{11}, and Y_{12}, respectively, with the device simulation data, shows a good match between the circuit model and the device simulation data. The purely resistive substrate model is valid up to approximately 10GHz, beyond which the imaginary part of the Y-parameter becomes more and more comparable to the real part.

15.2.3 SCM Model Validation Example

A test circuit, fabricated in a 0.13μm CMOS process with lightly doped bulk silicon, is used to validate the SMC model. Figure 15.12 shows the layout of the test circuit, which consists of a p+ contact (the substrate noise injector), and four substrate noise sensors, located at various

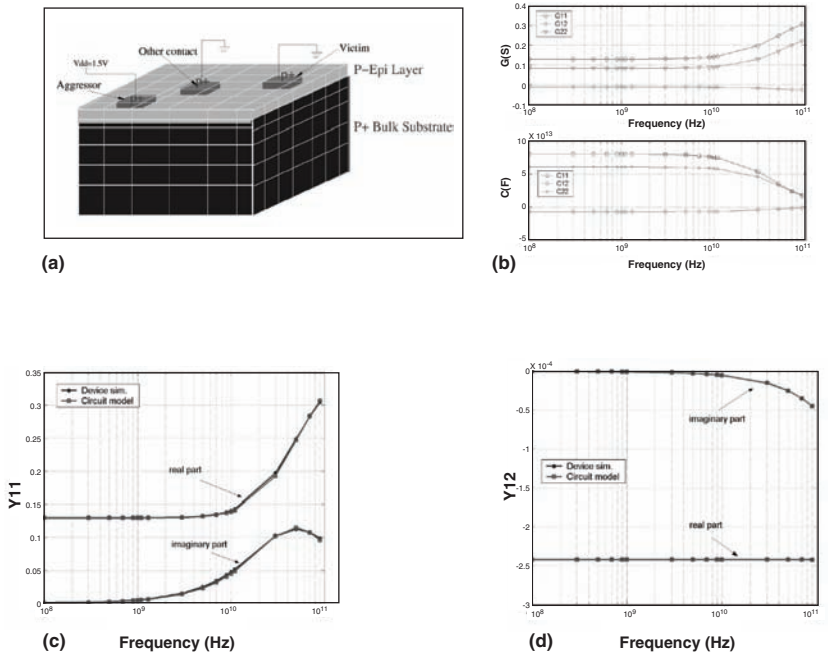


Figure 15.10 High-Frequency Substrate Modeling Example: (a) Heavily Doped Substrate with Lightly Doped Epitaxial Layer, (b) 3-D Device Simulation with Frequency-Dependent Behavior, and (c) Device Simulation and Circuit Modeling for Y11 and (d) Y12.

distances (ranging from 25μm to 16μm) from the noise injector. The substrate sensors are wideband differential PMOS-only sensors, as proposed by Iorga, Lu, and Dutton [16].

For the measurement, a 600mV pulsed wave at 2MHz is injected through the noise injector. The noise propagates through the substrate, and is sensed at different locations by the noise sensors. Simulation is performed by including the noise source, the substrate-coupling network generated by the SCM modeling method, and the sensor circuit. Figure 15.13 compares the measurement data to the simulation results. The simulation results generally agree with the measured data, except for when the distance reaches 165μm, where the noise is below the sensor resolution. The noise propagation and decaying behavior, in this lightly doped process, is interesting. The noise decreases rapidly in the range of from 0μm to 25μm, confirming the near-field effect discussed in Section 15.2.1. The noise continues to decay, as it propagates over distance into the far-field region. The simulation data shows that the noise attenuation trend decreases as the distance grows. The noise level drops only 3mV, when the distance increases from 110μm to 165μm, while the noise drop is 20mV when the distance increases from 25μm to 55μm. This again confirms that substrate noise does not attenuate, in a strictly linear fashion, over its propagation distance. It drops fast in the near-field region, but asymptotically approaches a saturation level in the far-field region.

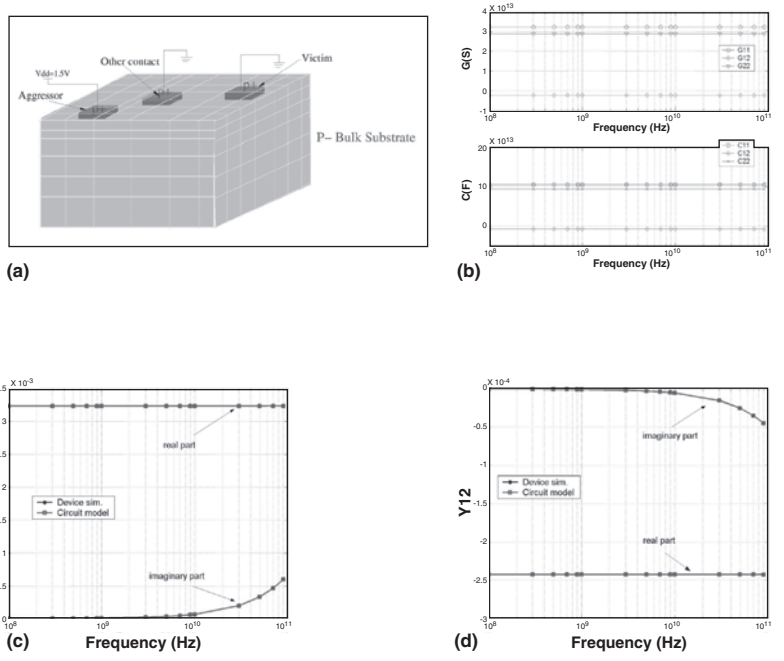


Figure 15.11 High-Frequency Substrate Modeling Example: (a) Lightly Doped Substrate Structure, (b) 3-D Device Simulation Results with Self/Mutual Conductances/Capacitances Almost Constant over Wide Frequency Range, (c) Comparison of 3-D Device Simulation and Equivalent Circuit Modeling Results for Y_{11}, and (d) Y_{12}

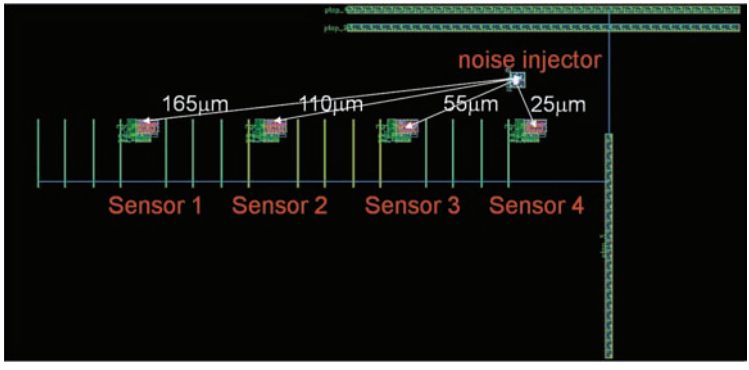


Figure 15.12 Test Circuit in a 0.13µm CMOS Process for Substrate Noise Propagation Measurement and SCM Model Validation

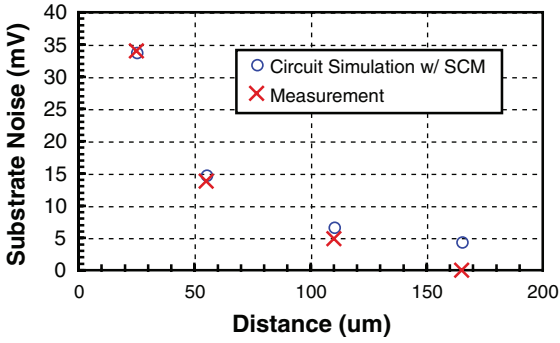


Figure 15.13 Correlation Between Measured Substrate Noise and Simulated Noise Using SCM Model

15.3 Measurement Techniques

The measurement of substrate noise plays an essential role for understanding the noise characteristics, propagation behavior, and its impact to sensitive analog circuits. Because of the spatial-dependent nature of the substrate noise, it is ne-cessary to design and deploy compact measurement module to allow in-situ substrate noise characterization. This section first describes the general requirement for substrate noise measurement and then discusses a specific on-chip substrate noise measurement structure consisting of noise generator and noise monitor.

15.3.1 Substrate Noise Property and Measurement Requirement

Developing noise-measurement techniques requires an understanding of substrate noise properties. First, substrate noise, in the strictest sense, is a stochastic process, due to the random nature of digital switching activity. However, in most cases, the system reference clock synchronizes all switching activities. Substrate noise typically exhibits strong periodic behavior in the time domain, and contains spikes from the reference clock, and its harmonics, in the frequency domain. Consequently, the substrate noise can be characterized using its auto-correlation and power spectral density. Second, substrate noise is normally not as significant as supply noise. Due to the substrate's intrinsic attenuation, and the biasing contacts spread over the chip, the peak-to-peak noise ranges from a few tens of mV (in a highly noisy environment) to a few hundred μV in a less noisy design, such as the example of a low-power PHY interface. Third, substrate noise is distributed over both its aggressor origins and victim destinations, making it layout- and floorplan-dependent.

The substrate noise-measurement module must meet a few critical design requirements: It must provide voltage resolution finer than sub-mV, and the measurement bandwidth needs to cover at least the third harmonic of the system reference clock, in order to capture the majority of the substrate's noise dynamics. Because the substrate noise is layout and location dependent,

in-situ measurement is highly desirable. Therefore, the module has to be compact enough so that it can be placed in any location for probing noise, with minimum or no interference with the nearby circuits. Preferably, the measurement is performed entirely on-chip, with the minimum pin count and package resources required to support the on-chip measurement. Later sections address these design challenges, where the proposed measurement technique and circuit implementation are discussed.

15.3.2 Substrate Noise Monitor Circuit

Figure 15.14 is a block diagram of a substrate noise monitor [17]. This noise monitor is based on the power-supply noise monitor, described in Section 16.4.1 "Supply Noise Measurement Circuits." After making only minor modifications to the sensing front end of the power-supply noise monitor, one can use it to measure near-ground substrate noise. The sensing front end includes a p+ substrate probe contact, a sample-and-hold circuit (S/H), and a level shifter (L/S). It captures instantaneous substrate noise, at the desired location, using the S/H via the p+ substrate contact. Because the sampled substrate noise is typically tens of mV around the reference ground voltage, a PMOS source follower-based L/S is used, in order to provide some gain and add a proper DC bias. The rest of the circuit is identical to the supply-noise measurement circuit described in Chapter 16.

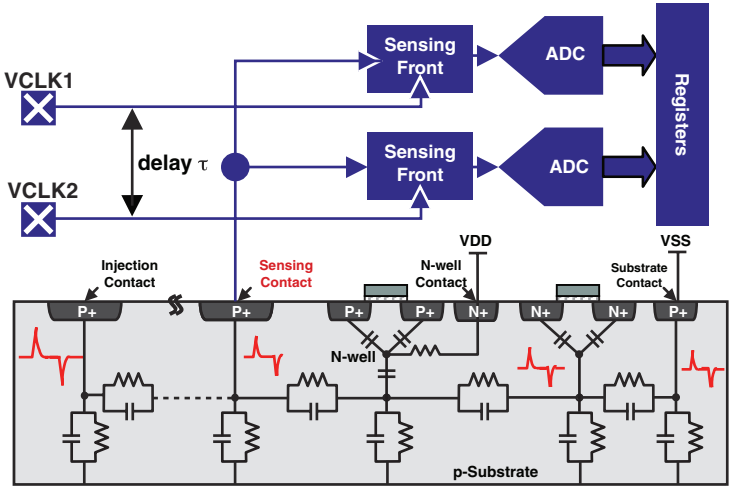


Figure 15.14 On-Chip Substrate Noise Monitor Circuit

15.3.3 Noise Generator Circuit

Figure 15.15 is a simplified schematic diagram of the substrate noise generator [17]. As shown in the figure, an external clock (NCLK) drives the staged buffers with MOSCAP loading. The

source/drain side of the MOSCAP connects directly to the p+ substrate contact. Switching currents are generated, associated with the NCLK edge transitions. These currents are then injected directly into the chip substrate. One determines the strength of the excited substrate voltage noise using these factors: the driver output impedance, MOSCAP capacitance, NCLK frequency, effective clock edge sharpness, and the input impedance (as seen from the noise injection contact into the entire substrate network). The actual implementation includes an array of binary-weighted noise sources that allow the adjustment of the generated noise strength.

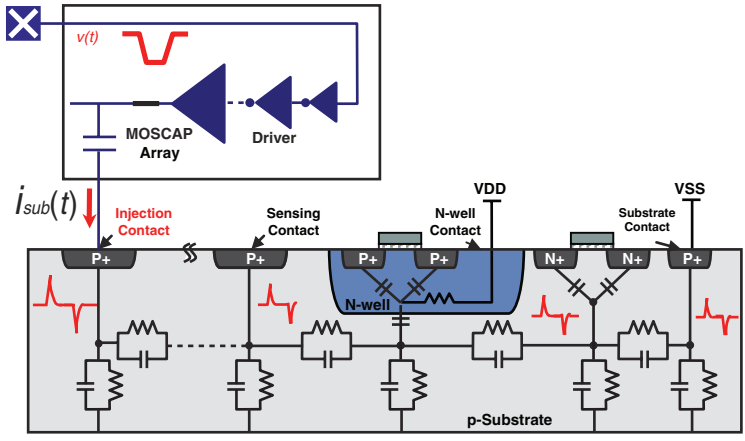


Figure 15.15 On-Chip Substrate Noise Generator Circuit

15.4 Case Study

In this section, the substrate noise and its jitter impact in a high-speed, low-power memory controller interface design example are analyzed using the on-chip measurement structure previously discussed in Section 15.3.

15.4.1 Test System Overview

For illustration, this section considers the on-chip substrate noise-monitor circuit, described in Section 15.3.2. It is built in a 40nm low-power CMOS process. The test chip is primarily a low-power memory controller PHY interface (similar to the one described by Palmer, et al. [18]), which achieves data rates 3.2GB/s–4.3Gb/s using low-swing differential signaling with advanced power-management features. Figure 15.16(a) shows the layout of the substrate noise-measurement structure and the placement in the entire PHY interface. The structure is composed of two noise monitors in the upper part, and the noise generator in the lower part. The entire measurement structure is about 40μm×250μm in this particular process. The noise module is not located inside the

PHY, nor is it particularly close to any of the interface links. This location is selected for the noise module because the major emphasis of this initial implementation work is to demonstrate the feasibility, functionality, and performance of the self-contained noise measurement module, not to aggressively deploy the noise probes at the more sensitive locations (such as the PLL and clock distribution circuits).

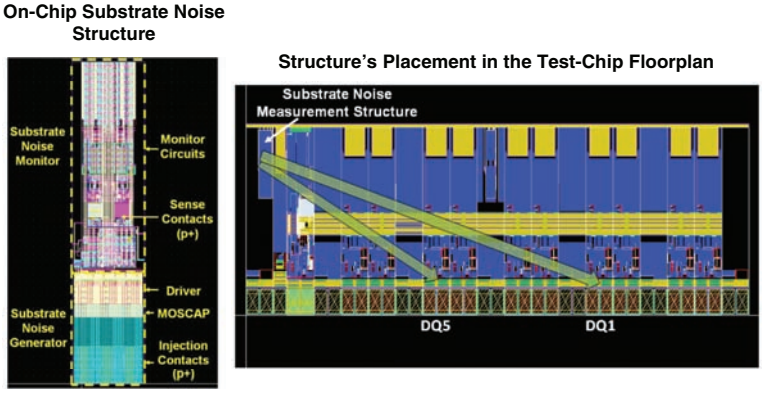


Figure 15.16 Low-Power Memory Controller PHY Test Chip with On-Chip Substrate Noise Measurement Circuit

15.4.2 DC Calibration Results

The first step in the measurement procedure is performing the DC calibration. During this step, the DC voltage on the p+ substrate contact, at the sensing front end, is swept in a certain range, typically ±100mV. At each given DC voltage, multiple samples are taken from each of the two noise monitors, and the corresponding counter values for data post-processing are recorded. As mentioned previously, averaging the multiple values effectively filters out the high-frequency random noise from the VCO device's intrinsic noise sources. At well within the lock range of the VCO, the linear fitting approximation is sufficient to represent the mapping from the measured voltage to the digital count. Figure 15.17 shows the DC calibration results for the two noise monitors. They show similar slopes, but with different y-intercepts. This difference is due to the device mismatch and process variation. The slope parameter is a function of the VCO gain and conversion time. It represents the quantization error, and thus imposes a voltage resolution limit. As shown in the figure, the two monitors achieve their voltage resolution at 142μV/LSB and 138μV/LSB, respectively. This performance exceeds the sub-mV resolution requirement.

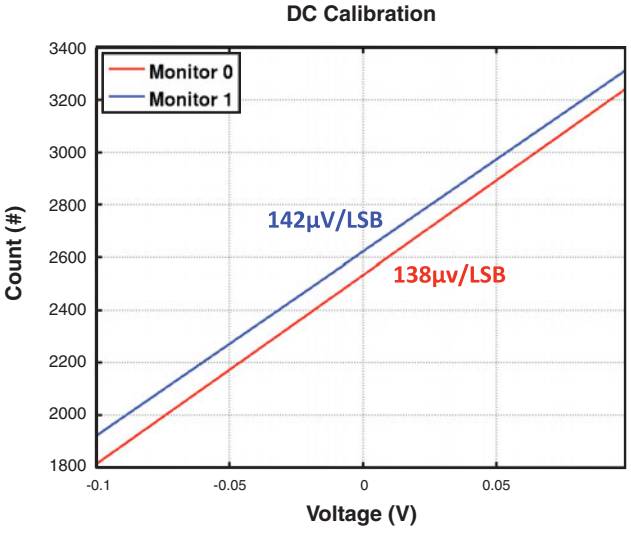


Figure 15.17 Substrate Noise Monitor DC Calibration

15.4.3 AC Measurement Results

Both monitors simultaneously capture the instantaneous substrate noise, but at two time instances with a relative delay (τ), which is precisely controlled by the external clock source equipment. The substrate noise can be characterized by its auto-correlation:

$$R(\tau) = E[V_{sub}(t)V_{sub}(t+\tau)] \tag{15.5}$$

or, by its auto-covariance:

$$R(\tau) = E\left\{[V_{sub}(t) - \overline{V}_{sub}(t)][V_{sub}(t+\tau) - \overline{V}_{sub}(t+\tau)]\right\} \tag{15.6}$$

The frequency-domain property is described by the power spectral density (PSD) of the substrate noise:

$$S_{Vsub}(f) = \int_{-\infty}^{+\infty} K(\tau)e^{-j2\pi f\tau}\, d\tau \tag{15.7}$$

The maximum realizable delay (τ_{max}) determines the frequency resolution. The minimum realizable delay step (τ_{step}) determines the measurement bandwidth.

Figure 15.18 shows the amplitude spectra of the measured substrate noise with quiet state and noise generator turned on. During the quiet state, there is no data transaction activity, and the PLL is inactive. The measurement noise floor is at about –15dBmV. The small spikes are the

result of the switching of the logic clock, necessary for register access and system initialization. When the noise generator is turned on with NCLK at 700MHz, one can clearly see the intentional noise peak at 700MHz, as well as its major harmonics.

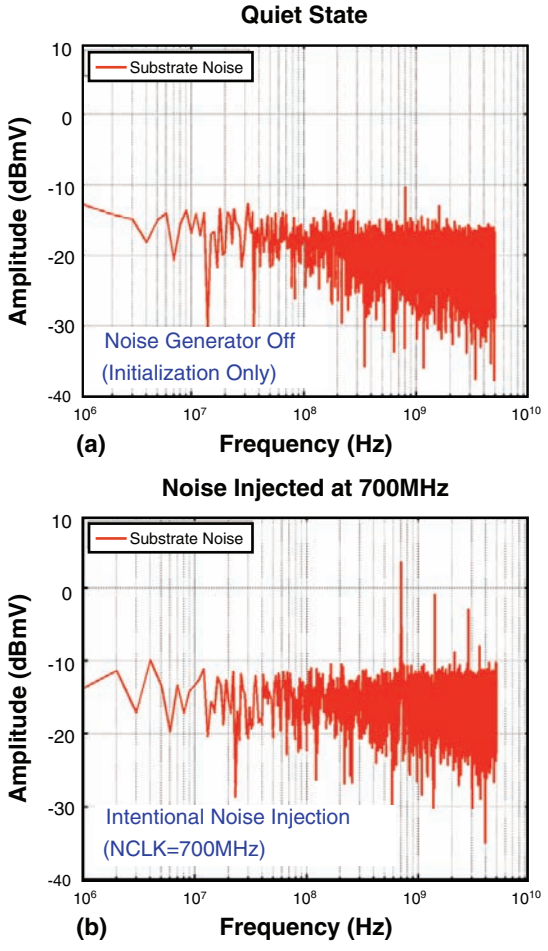


Figure 15.18 Measured Substrate Noise Spectrum for PHY Initialization Only and Intentional Noise with 700MHz Modulation Frequency

Figure 15.19 shows the measurement results for the substrate noise due to the active operation of the PHY itself. In this mode, all the data links transmit a certain data pattern continuously at 3.2Gb/s per link, with the PLL reference clock at 400MHz. Various sources generate the substrate noise, including data path transactions, differential output driver activity, and PLL clock

generation, buffering and distribution. The first plot shows the measured substrate noise spectrum when all the data links transmit a 0xAAAA data pattern. The major noise spike is about 660μV at 800MHz. The second plot shows the substrate noise spectrum when all the data links transmit the same PRBS data pattern. The spectrum shows the group energy centering around 800MHz, due to the random data pattern. Because the goal of the test chip is to achieve low-power using differential low swing signaling, seeing that even the peak substrate-noise component is below 1mV is not surprising. Moreover, this test chip only includes the low-power memory controller PHY. When integrated with the memory or ASIC core, the substrate noise is expected to become much worse, due to the strong switching activity during the memory core operation. Depending on the digital circuit's size, activity, switching scenario, and substrate contact placement, the substrate noise can be 10mV–30mV.

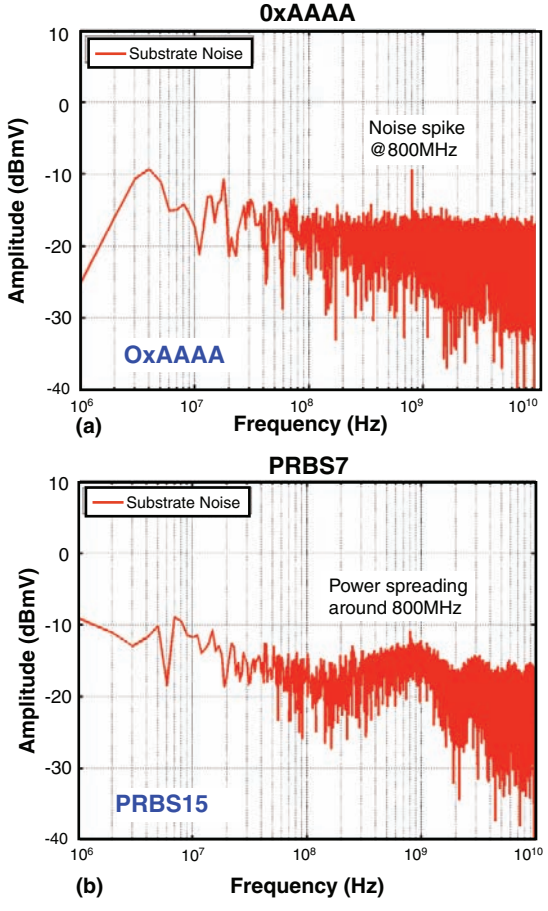


Figure 15.19 Measured Self-Induced Substrate Noise with Different Data Pattern Activity

15.4.4 Substrate Noise Induced Jitter Measurement

When analyzing high-speed link systems, being able to identify the major noise sources, for example, supply noise and substrate noise, is important. Understanding how great an impact the noise has on the link jitter performance is even more important. In this study, the on-chip measurement structure is utilized to investigate the substrate noise induced jitter (SNIJ). As an example, Figure 15.20 illustrates the impact of substrate noise on the PHY jitter performance, measured at two data links, which are different distances from the substrate noise generator. The figure shows the measured jitter spectra, both with and without the 1GHz intentional substrate noise injection.

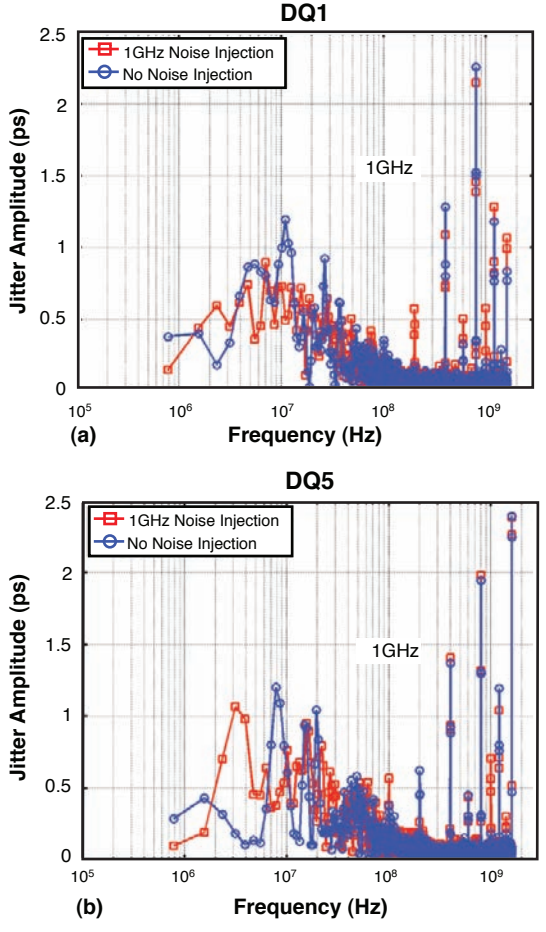


Figure 15.20 Measured Jitter Spectrum Showing the Substrate Noise Impact on PHY Jitter Performance at DQ1 and DQ5

One can make a few observations here. First, single-frequency substrate noise generally excites a jitter component at the same frequency. In this example, the 1GHz substrate noise mainly excites the jitter component at 1GHz. This is clearly seen in Figure 15.20, by the significant increase in jitter amplitude resulting from the 1GHz noise injection. Second, the substrate noise experiences different levels of attenuation when it arrives at different locations. In this example, DQ5 is closer to the substrate noise generator than DQ1; see Figure 15.16(b). Therefore, DQ5 suffers from a higher substrate noise than DQ1. Figure 15.20 shows that DQ5 sees about a 1.1-ps peak-to-peak jitter increase at 1GHz, whereas DQ1 sees about an 0.8-ps peak-to-peak jitter degradation at 1GHz.

Based on the earlier observations, a parameter called SNIJ sensitivity, S(f), is introduced here and defined as follows:

$$S(f) = \frac{SNIJ(f)}{V_{sub}(f)}[ps,pp/mV,pp]. \tag{15.8}$$

Similar to the concept of PSIJ sensitivity (discussed in Chapter 14), SNIJ sensitivity is also a system-dependent characteristic, which links the substrate noise sensed by the victim, to the jitter impact seen at its output clock or data. SNIJ sensitivity is determined by the choice of system architecture choice and circuit implementation, including the clocking scheme, data path, signaling, and so on. It is independent of activity or its location within the floorplan. In the example low-power PHY interface, identical link slice architecture is used to instantiate all the data links. Depending on their distance to the noise source, one expects these links to see different levels of substrate noise. However, all of them should have the same, or similar, SNIJ sensitivity.

Figure 15.21 shows the measurement results of the SNIJ sensitivity on DQ1 and DQ5. Although not exactly the same, the two DQ links generally exhibit similar sensitivity behavior. They both show strong sensitivity (as high as 1ps/mV–2ps/mV) at the frequencies related to the 400MHz PLL reference clock. Clear spikes at the fundamental tone, its harmonics, and sub-harmonics are observed. These frequencies are where the system clock is most susceptible to disturbance due to substrate noise. Single-tone substrate noise, at the PLL reference clock, would have the greatest impact on the PLL dynamics, and thus its impact on the output clock jitter would be the most pronounced. The spike amplitudes are also roughly in line with each other. This further validates the underlying assumption in the concept of SNIJ sensitivity. When the controller PHY is integrated with the ASIC, the substrate noise is expected to be in the order of 10mV, which is much stronger than the PHY's self-induced noise, due to the strong switching activity in the digital core. The induced jitter will be approximately in the order of 10ps, consuming as much as 3% UI for the 3.2Gb/s link.

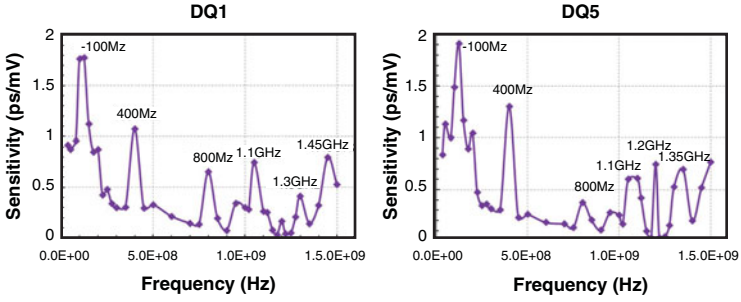


Figure 15.21 Measured SNIJ Sensitivity Profile for DQ1 and DQ5

15.5 Summary

Substrate noise is the next challenge, following power supply noise, to budgeting, designing, and analyzing high-speed interfaces. This chapter discusses substrate noise and its jitter impact on high-speed I/O interfaces.

The chapter first discusses the substrate modeling methodology, including both DC and high-frequency approaches. The chapter also presents an on-chip substrate noise-measurement structure (noise monitor) and implements its prototype with a low-power memory controller PHY interface. The auto-correlation-based measurement methodology greatly reduces the bandwidth requirement, which would otherwise be challenging to meet, using direct time-domain measurement techniques. The noise monitor achieves the voltage resolution of finer than 150μV, and a measurement bandwidth of at least 10GHz. Measuring the PHY's self-induced substrate noise verifies that it is not very significant in this stand-alone low-power PHY test chip environment. Additionally, the SNIJ sensitivity is characterized (aided by the implemented substrate noise generator), and observed consistent sensitivity results on two DQ links in the test chip. The on-chip measurement structure is proven useful in investigating the substrate noise and its impact. Self-contained and compact in size, it serves as a vital tool for further in-depth study of the impact of substrate noise jitter on the high-speed and low-power I/O interfaces for the future.

References

1. R. Gharpurey and R. Gray, "Modeling and analysis of substrate coupling in integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp.344–353, Mar. 1996.
2. J. Cho, J. Shim, E. Song, J. Pak, J. Lee, H. Lee, K. Park, and J. Kim, "Active circuit to through silicon via (TSV) noise coupling," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct. 2009, pp. 97–100.
3. R. Schmitt, H. Lan, C. Madden, and C. Yuan, "Analysis of supply noise induced jitter in Gigabit I/O interfaces," presented at the IEC DesignCon, Santa Clara, CA, 2007.

4. H. Lan, R. Schmitt, and C. Yuan, "Prediction and measurement of supply noise induced jitter in high-speed I/O interfaces," presented at the IEC DesignCon, Santa Clara, CA, 2009.
5. M. Badaroglu, P. Wambacq, G. Van der Plas, S. Donnay, G. Gielen, and H. De Man, "Impact of technology scaling on substrate noise generation mechanisms," in *Proceedings of IEEE Custom Integrated Circuits Conference*, San Jose, CA, Sep. 2004, pp. 501–504.
6. N. Verghese and D. Allstot, "Computer-aided design consideration for mixed-signal coupling in RF integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 314–323, Mar. 1998.
7. E. Charbon, R. Gharpurey, R. Gray, and A. Sangiovanni-Vincentelli, "Substrate optimization based on semi-analytical techniques," *IEEE Transaction on Computer-Aided Design*, vol. 18, no. 2, pp. 172–190, Feb. 1999.
8. M. van Heijningen, J. Compiet, P. Wambacq, S. Donnay, M. Engels, and I. Bolsens, "Analysis and experimental verification of digital substrate noise generation for epi-type substrates," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 1002–1008, Jul. 2000.
9. M. Badaroglu, M. van Heijningen, V. Gravot, J. Compiet, S. Donnay, M. Engels, and H. De Man, "Methodology and experimental verification for substrate noise reduction in CMOS mixed-signal ICs with synchronous," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1383–1395, Nov. 2002.
10. E. Schrik and N. van der Meijs, "Combined BEM/FEM substrate resistance modeling," in *Proceedings of Design Automation Conference*, New Orleans, LA, Jun. 2002, pp. 771–776.
11. A. Samavedam, A. Sadate, K. Mayaram, and T. Fiez, "A scalable substrate noise coupling model for design of mixed-signal ICs," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 895–904, Jun. 2000.
12. D. Ozis, T. Fiez, and K. Mayaram, "A comprehensive geometry-dependent macro model for substrate noise coupling in heavily doped CMOS processes," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Orlando, FL, Sep. 2002, pp. 497–500.
13. S. Kristiansson, F. Ingvarson, S. Kagganti, and K. Jeppson, "A surface potential model for predicting substrate noise coupling in integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1797–1803, Sep. 2005.
14. H. Lan, T. Chen, C. Chui, P. Nikaeen, J. Kim, and R. Dutton, "Synthesized compact models and experimental verifications for substrate noise coupling in mixed-signal ICs," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1817–1829, Aug. 2006.

15. H. Lan, Z. Yu, and R. Dutton, "A CAD-oriented modeling approach of frequency-dependent behavior of substrate noise coupling for mixed-signal IC design," in *Proceedings of International Symposium on Quality Electronic Design Conference*, Mar. 2003, San Jose, CA, pp. 195–200.

16. C. Iorga, Y.-C. Lu, and R. Dutton, "A built-in technique for measuring substrate and power supply digital switching noise using PMOS-based differential sensors and a waveform sampler in system-on-chip applications," *IEEE Transactions on Instrumentation and Measurement*, vol.56, no. 6, pp. 2330–2337, Dec. 2007.

17. H. Lan, M. Aleksic, R. Schmitt, N. Nguyen, and C. Yuan, "Investigating substrate coupling noise impact on low-power memory controller PHY interface using on-chip measurement structure," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference,* Oct. 2010, Austin, TX, pp. 137–140.

18. R. Palmer, J. Poulton, B. Leibowitz, Y. Frans, S. Li, A. Fuller, J. Eyles, J. Wilson, M. Aleksic, T. Greer, M. Bucher, and N. Nguyen, "A 4.3GB/s mobile memory interface with power-efficient bandwidth scaling," in *Symposium on VLSI Circuits Digest of Technical Papers*, Kyoto, Japan, Jun. 16–18, 2009, pp.136–137.

Part IV

Advanced Topics

This page intentionally left blank

CHAPTER 16

On-Chip Link Measurement Techniques

Dan Oh, Hai Lan, Ralf Schmitt, and Elad Alon

High-speed digital design depends on more than the accurate modeling described in previous chapters; it also depends on the tests and measurements necessary to verify the accuracy of those models.

Passive interconnects can be characterized either in the time domain, using Time Domain Reflectometry (TDR), or in the frequency domain, using Vector Network Analyzer (VNA). On the other hand, active devices (such as transmitters and receivers) can be tested using real-time oscilloscopes, digital sampling oscilloscopes, or a bit error tester (BERT). These measurement tools are a powerful and integral part of high-speed design yet they have the following limitations.

First, the aforementioned measurements are at the component level, so they may miss the complex interactions between the various components of the channel. For example, consider a case where large reflections force a transmitter out of saturation, so that it cannot reach the desired voltage swing, which in turn, causes the receiver to fail.

Second, the observed waveform may be distorted due to reflections, and may not represent the actual waveform at the transmitter or receiver. For example, Figure 16.1 illustrates two signal eye diagrams, one at the receiver ball and one at the pad. Due to reflections, the signal at the ball has significant distortion, whereas the signal at the pad has a very clean eye.

Third, certain channel behavior, such as jitter tracking (described in Chapter 10, "Clock Models in Link BER Analysis"), cannot be measured at the component level, because the noise cancellation occurs at the system level. High-frequency supply noise is another example where the noise measured at the package pin or ball is significantly different from the noise at the devices on-chip, due to the filtering nature of the package.

Finally, 3D package technologies, such as Package-on-Package (PoP), System-in-Package (SiP), and Multichip Package (MCP), make measuring the signal quality at the component level even harder. The traditional method of measuring each device at the component level does not

capture the complex interactions in 3D integration. For example, power supply noise and jitter are two of the most dominant voltage and timing error components in 3D-package systems, due to the limited area available to provide sufficient power delivery. This is particularly true for the top DRAM device (shown in Figure 16.1) where the supply path has a higher inductance. This inductance causes significant supply voltage noise.

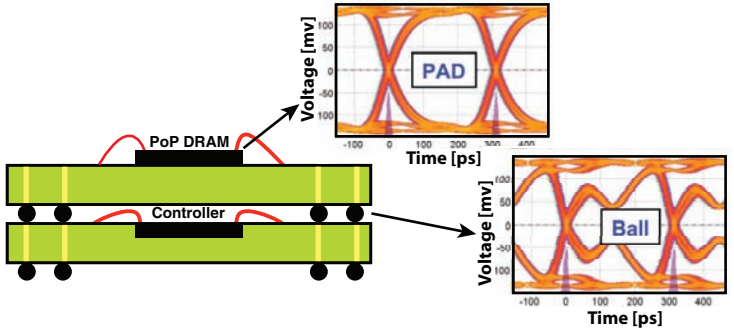


Figure 16.1 Data Signal (Using a PoP Differential Memory System Running at 3.2Gb/s) Measured at the Ball and Pad Locations

One way to overcome these limitations is to deploy accurate and reliable on-chip link performance measurements. This chapter describes several on-chip (or *in-situ*) measurement techniques. Key electrical parameters (such as voltage and timing margins, bathtub plots, BER eye diagrams, and signal waveforms) are measured using minimal, and commonly available, on-chip circuitry [1] [2]. This chapter demonstrates a waveform-capturing feature that is particularly useful when modeling the overall channel, with accurate analog amplification, or equalization, at the receiver front-end. In-situ measurements of the voltage and timing margins can be performed for system qualification during the production ramp. A loopback path within the device can be implemented and used to perform component-level device testing. Finally, a simple noise-monitoring circuit is designed to measure the power supply noise [3] [5]. Additionally, this noise-monitoring circuit can be used in conjunction with an additional noise generator, to obtain the system power distribution network (PDN) impedance, as well as the supply noise induced jitter sensitivity.

On-chip measurements not only provide accuracy and convenience when characterizing high-speed systems, they also enable low-cost testing without expensive testing equipment (such as a high-frequency BERT, or a digital sampling oscilloscope). On-chip measurements are also very powerful and useful in the production environment. Section 16.1 discusses the measurement circuitries and techniques required for shmoo and BER eye diagrams. Section 16.2 extends these techniques to capture signal waveforms. Section 16.3 presents hardware measurement data (using a low-power differential memory system, based on PoP package [2], [6], and [7]), as well as correlation data (based on the statistical link simulation described in Chapter 9, “Fast Time-Domain Channel Simulation Techniques”). Section 16.4 describes the noise monitor and generator

circuitry, along with the measurement techniques to measure on-chip noise in the time and statistical domains. Section 16.5 describes more measurements for power distribution network impedance, jitter sensitivity, and link margin sensitivity.

16.1 Shmoo and BER Eye Diagram Measurements

A shmoo plot is a process used to map passing and failing regions over varying timing or voltage parameters. Specifically, at a given voltage and timing setting, the transmitter sends a pattern (typically PRBS) to the receiver through the channel. The receiver compares the received pattern, bit-by-bit, to the known pattern. The receiver reports a "pass" if no errors are found, finding an error results in a "fail." This process is repeated for the entire range of voltage and timing values. Typically, the timing is varied within a bit time, and the voltage within the allowable swing range for the transmitter voltage. The hardware requirement for shmoo testing is a bit-error detector and a means of varying the desirable parameters.

First, let us consider a *timing shmoo*. The transmitter data signal or clock, or the receiver clock can be skewed to shmoo the timing to measure the link-timing margin. In a memory interface design, the timing adjustments for read and write operations can optionally be made at the controller, which simplifies the DRAM interface.

In today's high-speed interfaces, timing-adjustment circuitry is commonly implemented to handle pin-to-pin timing variations, making timing adjustments a part of the high-speed system design, rather than a specific feature of shmoo testing. Regardless of where the timing-adjustment circuitry comes from, it can be used to characterize the link voltage and timing margin.

In the case of a *voltage shmoo*, several approaches are available. With single-ended signaling, either the reference voltage or transmitter common-mode voltage can be adjusted. With differential signaling, the common-mode voltage of either the transmitter or receiver is adjusted. Note that, due to transmitter voltage headroom issues, adjusting the receiver is better, because the signal swing is attenuated significantly at the receiver.

Yet another approach is to use an adaptive sampler to adjust both voltage and timing [1]. This approach allows measurements of real-time traffic data. However, it adds a significant cost to the hardware design. In the test vehicle used in the low-power I/O interface design [2], an offset calibration circuit in the receiver sampler is used to adjust the common-mode voltage. Low power I/O interface designs use low-swing signaling, which requires a sensitive sampler design [6]. A receiver offset calibration circuit is highly desirable to improve the sampler sensitivity. The drawback of this technique is that, typically, the offset cancellation is designed for a limited voltage range, and a full swing characterization is not possible, in most cases. When a transmitter has a half-swing mode, and it can be used in combination with offset calibration to take measurements.

The first plot in Figure 16.2 illustrates a shmoo eye. The green opening of the eye is the passing region. The gray area surrounding the eye is the failing region. (A location is "failed" if there is a single error.) The horizontal and vertical eye openings provide the timing and voltage margin, respectively. If an error counter is available, a BER plot can be generated using the existing shmoo features. With an error counting feature, bit error rates can be mapped, instead of a

simple pass and fail, which results in the 2D BER eye shown in the second plot in Figure 16.2. The horizontal cross-section of the BER eye represents a timing bathtub curve, whereas the vertical cross-section represents a voltage bathtub curve. A single side of the voltage bathtub curve is referred to as a *waterfall* plot.

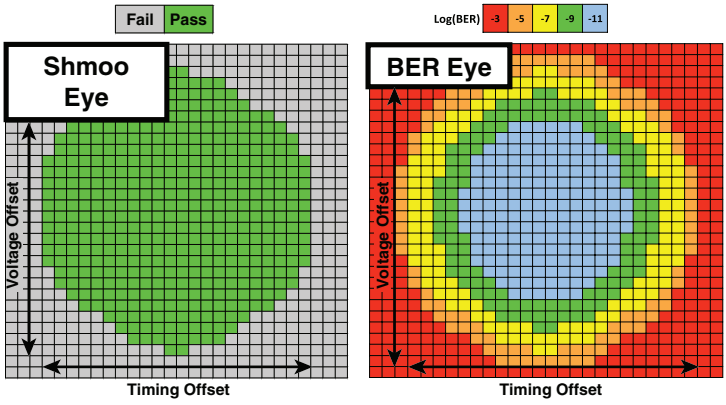


Figure 16.2 2-D Shmoo and BER Eye Diagrams

In summary, the following hardware is required to generate a BER eye plot:

- Pattern generator to send known data pattern
- Bit error detection circuitry and counter
- Voltage shmoo feature (based on either the transmit swing adjustment, reference voltage adjustment, or receiver offset adjustment)
- Timing shmoo feature (based on either the transmitter data or clock phase adjustment, or the receiver side clock phase adjustment)

16.2 Capturing Signal Waveforms

Capturing signal waveforms using on-chip measurement circuitry is challenging, because it requires a sub-sampling technique using high-bandwidth samplers [8–11]. Modern I/O designs often utilize the maximum transistor bandwidth to send data, so implementing a sub-sampling circuit could take significant power and silicon area overhead. Consequently, although sub-sampling techniques can be implemented in test vehicles, they are not suitable for production chips. A simpler version of the sub-sampling circuit, based on an additional adaptive sampler, is used for serial link designs [1] [12], and the same version (without the adaptive sampler) is implemented in memory interfaces [2].

In [2], a waveform is indirectly captured by measuring the BER using a technique similar to the technique described in the previous section. The only additional hardware required is a

masking feature, used to select an error at a particular bit location. The basic principle is illustrated using a simple step as the input. First, the input pattern ...000111... is sent repeatedly. Then, the 2-D BER maps are measured for each bit location. Figure 16.3 demonstrates the 2-D BER maps for the pre-transition and post-transition bits, respectively. In this example, the error locations are mapped by comparing the received data to 1. By tracking the location with the error probability of 0.5, the signal waveform can be traced. To locate the 0.5 error location, the voltage offset is shmooed, as described in the previous section. Because the error rate of 0.5 is considered, knowing the correct bit value is not necessary. In a real implementation, the shmoo process can be optimized to track only near the 0.5-bit error probability line, instead of sweeping the entire voltage range. To filter out jitter due to random noise an averaging scheme can be used, or the measurement curve can be fitted with smooth functions. The overall process is summarized in Figure 16.4.

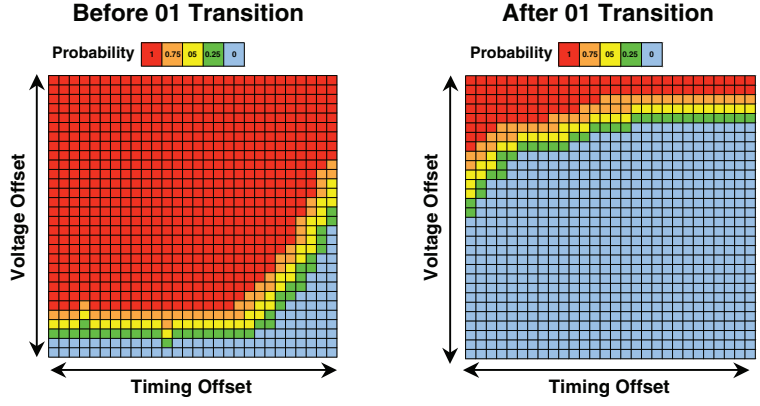


Figure 16.3 BER Plots of the Step Responses: Before and After Transition

For SerDes applications, Clock Data Recovery (CDR) requires an edge transition. The example in Figure 16.5 illustrates the addition of toggling bits, placed before and after the transition, but at a sufficient distance to minimize their impact. This bit error-based method is not limited to simple data patterns; it can also be used with arbitrary patterns, as long as they can be sent repetitively, which is not an issue for a synchronous I/O interface.

Because this measurement is based on on-chip timing and voltage shmoo features, the adjusted voltage and timing values can contain non-linearity errors. Most on-chip voltage and phase adjustments deviate from the ideal settings; this is referred to as a non-linearity error. Non-linearity errors are particularly severe for extremely high or low voltage values, and for timing values near octal boundaries. Fortunately, non-linearity errors can be characterized using off-chip measurements, and the result is used to correct the on-chip measured waveform. Linearity measurement can be done by comparing the measurement data with the supplied static value. Detail techniques are implementation-dependent, and are not covered in this book.

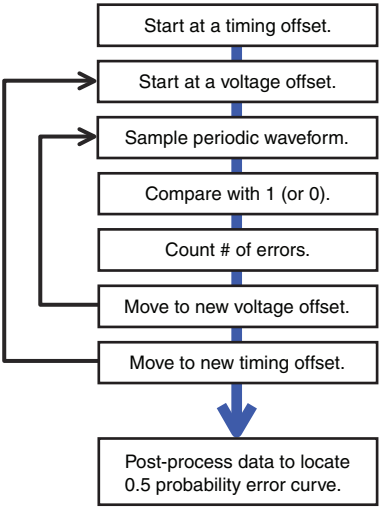


Figure 16.4 Waveform Measurement Flow

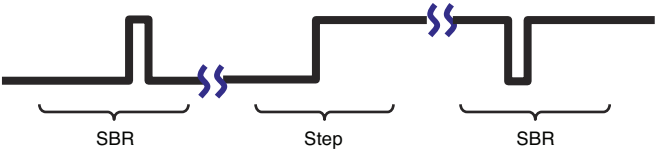


Figure 16.5 A Step Response with Single Bits to Create Edge Transitions for CDR

Figure 16.6 shows measurement results, based on SerDes applications [1]. The first plot shows a measured raw step response. The second plot shows the calibration curve, based on linearity measurement. The final plot illustrates the final step response after correction. (Note the relatively small non-linearity.) On the other hand, Figure 16.7 shows a step measurement of the PoP channel, illustrating both the original and corrected responses. A significant non-linearity is observed in this case, because the channel is targeted for a low-power application, and the design tolerances of the phase mixer and voltage twister are relaxed to reduce power consumption.

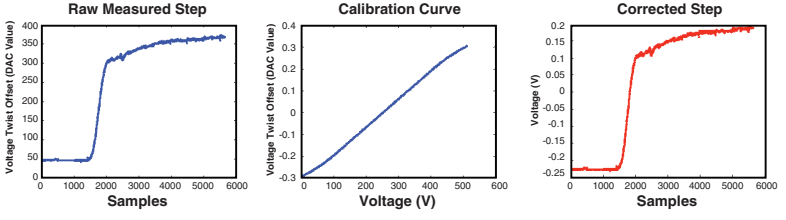


Figure 16.6 Measured Response, Calibration Curve, and Corrected Response

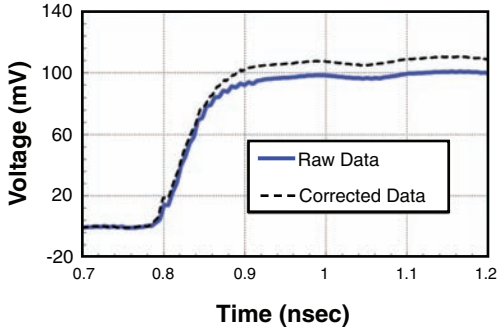


Figure 16.7 Measured and Non-Linearity Corrected Step Responses for PoP Test Vehicle

16.3 Link Performance Measurement and Correlation

Although any data pattern can be captured using these on-chip measurement techniques, they are particularly useful for capturing single-bit or step responses. These responses provide a complete channel model, which includes the driver and receiver analog effects (such as driver slew rate, receiver bandwidth, chip parasitic, any receiver continuous-time linear equalizer, or preamp). After a single-bit or step response is obtained, it can be used to generate ISI distributions for margin prediction and BER calculations, as described in Chapter 9. Figure 16.8 illustrates the overall process for computing the BER. Using the input step response, a fast-time domain simulation is performed to generate an ISI histogram. Then, this ISI histogram is convolved with other noise, or jitter distributions, if necessary. The resulting Probability Density Function (PDF) is integrated to compute the Cumulative Density Function (CDF), which is used to generate a BER eye. The final system bit error rate is calculated by taking the conditional PDF with a receiver sampling distribution. By sweeping the receiver sampling distribution, the link bathtub is finally obtained.

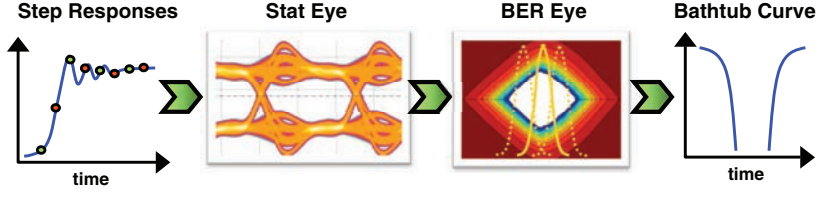


Figure 16.8 Fast-Time Domain Statistical Simulation Flow Based on Step Responses

Using this fast statistical simulation method, the BER eye diagram can be obtained based on the measured step response. Then, this simulated eye diagram is compared to the measured eye diagram, using the BER eye measurement feature described in Section 16.1. In simulation, 1.2ps of random jitter (obtained from other measurements) is added. The two eye diagrams are

shown in Figure 16.9. Although this figure shows a slight mismatch, in terms of the eye opening, overall, a reasonably good match exists between the two plots. Sometimes, using the half swing mode to capture BER eyes is useful. Figure 16.10 shows BER eye measurements, using both the full and half swing modes of a transmitter twister.

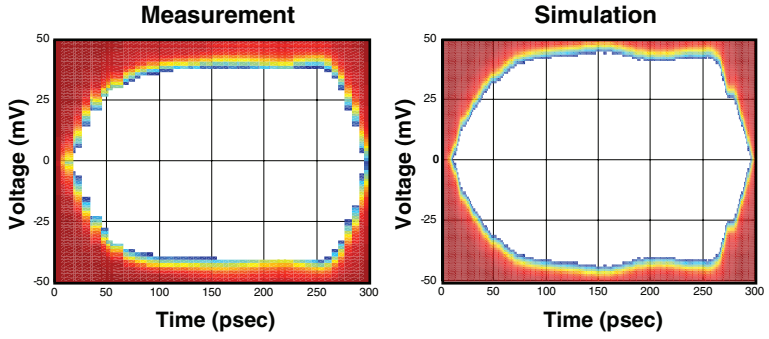


Figure 16.9 BER Eye Diagrams Based on Simulation and Measurement

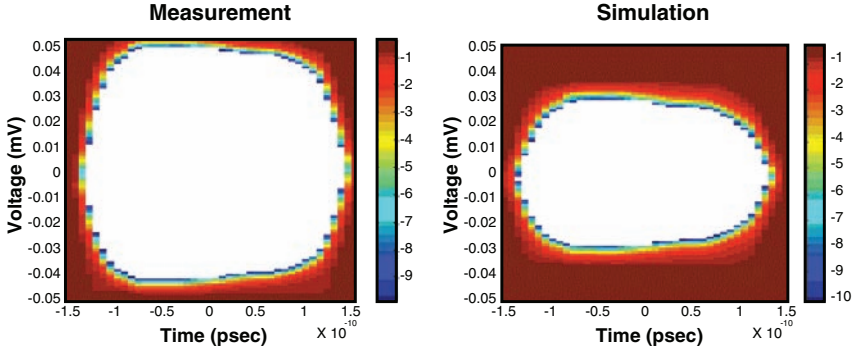


Figure 16.10 BER Eye Diagrams Based on Full-Swing and Half-Swing Mode of Transmitter Swing Twister

16.4 On-Chip Supply Noise Measurement Techniques

Power supply noise is one of the major factors constraining the performance of modern high-speed link designs. The characterization of power supply noise is crucial for a robust and reliable channel implementation. However, as discussed earlier, accurately measuring high-frequency supply noise off-chip is very difficult. As a result, various on-chip supply-noise measurement techniques have been developed to observe over- or under-shoot events over a certain time window [13], or reconstruct repetitive noise in sub-sampling scope mode [10]. Unfortunately, although they are useful when measuring specific properties of supply noise, these techniques

cannot capture the high-frequency noise present in a multi-gigahertz I/O interface. The voltage resolution of these techniques is usually inadequate for measuring the noise present in high-speed designs.

Ideally, one would like to measure the supply noise, on the internal power rails, for frequencies up to a multiple of the fastest toggle frequency. For a system with a data rate of 6.4Gb/s (and a maximum data toggle rate of 3.2GHz), the supply noise frequency can reach 10GHz or more. Measuring supply noise on an internal supply rail at these high frequencies is a challenging task, and requires special noise-monitor circuits. Both the frequency domain spectrum of the supply noise and the supply noise waveforms in the time domain that facilitates correlation with simulation need to be measured. So, a noise monitor with sufficiently high bandwidth for frequencies beyond 10GHz is desirable. This monitor can not only provide measurement results for the frequency domain noise spectrum, but can also be used to construct the time domain noise waveforms. Therefore, a measurement technique with bandwidth up to multi-gigahertz, and voltage resolution finer than mV is highly desirable.

To meet this requirement, this section reviews the on-chip supply-noise characterization method and measurement circuits introduced in references [3] and [4]. Figure 16.11 is the block diagram of the on-chip noise monitor and generator. It consists of a noise monitor block, driven by two sampling clocks with controllable delays (VCLK1 and VCLK2), and a noise generator block with one modulation clock (NCLK). The monitor block samples the analog supply voltage and converts it to digital code. The overall supply noise characteristics are reconstructed by their auto-correlation (see Section 16.4.2), and power spectral density (PSD). The noise generator block creates intentional noise on the power supply grids, at the desired frequency, and at a controllable level. This noise is then used to characterize various system responses (for example, timing margin and supply noise jitter sensitivity).

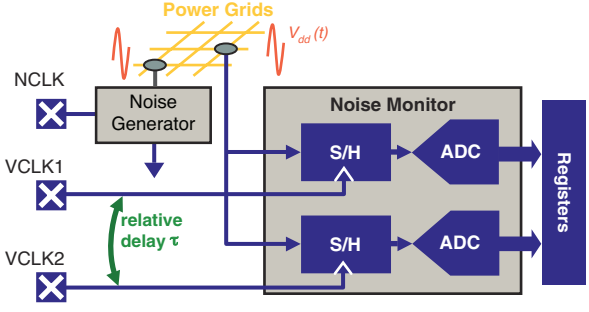


Figure 16.11 On-Chip Supply Noise Monitor and Generator

16.4.1 Noise Monitoring Circuit

Figure 16.12 is the schematic of the noise monitor circuitry. It consists of a pair of identical supply noise monitoring circuits. Each monitor includes a sample-and-hold (S/H) circuit as the sensing front-end, a ring type VCO as the voltage-to-frequency converter, and a digital counter as the

A/D converter. Two sampling clocks running at the same frequency and with a precisely controlled relative delay clock the two noise monitors. The instantaneous voltage on the supply rail is sampled on the toggling edge of the sampling clock, and subsequently held when it de-asserts. The sampling circuit is implemented using PMOS switches, which makes measuring the supply voltage easier. Moreover, a simple PMOS switch can achieve very high bandwidth, which is necessary for the monitoring circuits to capture the dynamic behavior of the supply noise up to several harmonics of the highest data toggle rate. During the hold mode, the buffered sample voltage is used as the control voltage of the VCO, in order to set its output oscillation frequency.

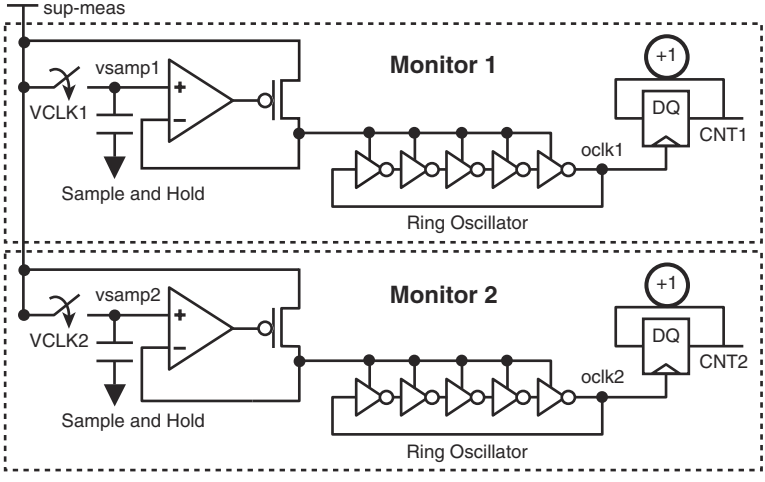


Figure 16.12 On-Chip Noise Monitor Circuit

The ring-type VCO is chosen to handle high-frequency measurements with a wide tuning range, because it can easily achieve several times of a FO4 cycle time with better tuning range. The output of the VCO is the input to a 16-bit counter. The counter is enabled during the hold mode, so that the VCO provides an output clock with a stable frequency. The counter output is then stored in registers, and read out for post-processing. The resulting digital count is proportional to the VCO control voltage, which, in turn, is proportional to the instantaneous supply voltage. If the supply voltage is swept in DC, the corresponding digital count can be saved producing the DC calibration curve that can be used to map the digital register count to the supply voltage value sampled at the sensing front-end. The counting process also ensures that the averaged VCO frequency is measured. It essentially filters out the high-frequency random noise, which stems mostly from device intrinsic noise sources, rather than from the supply noise of interest. The VCO gain, and the conversion time, determines the overall achievable resolution.

16.4.2 Noise Generator Circuit

Figure 16.13 is a schematic diagram of the noise generator. It consists of an array of binary-weighted current sources. It connects supply to ground by a shorting current modulated by the input clock. The shaped current contains the fundamental tone and odd harmonics of the clock signal frequency. When injected into the supply rail, the current interacts with the supply PDN, generating supply voltage noise with its dominant component at the desired frequency. Meanwhile, the strength of the generated voltage noise is adjusted by selectively turning on the binary-weighted current sources. This makes adjusting the noise amplitude possible, based on the PDN characteristics at any frequency. For a fixed amount of current injection, the resulting voltage noise will be more emphasized at frequencies near the peak of the PDN resonance frequency, and more suppressed at other frequencies, where the PDN impedance is small.

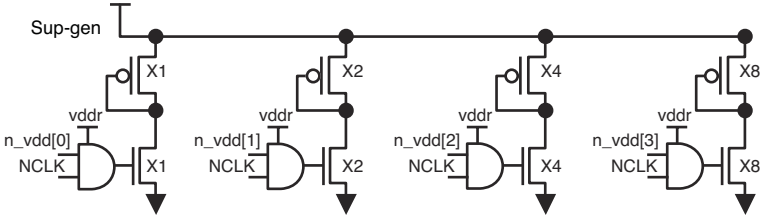


Figure 16.13 On-Chip Noise Generator Circuit

16.4.3 Supply Noise Measurement Techniques

Depending on the measurement requirements, the previously described on-chip noise measurement structure can be operated in two modes: *auto-correlation mode* (for frequency-domain measurements), and *sampling-scope mode* (for time-domain measurements). Supply noise typically exhibits steady-state behavior, when the I/O system continuously reads or writes repeated, fixed, or PRBS data patterns. In these situations, characterizing the supply noise in the frequency domain usually makes sense. The auto-correlation mode provides reliable measurements for frequency-domain characterization. In the auto-correlation mode, two free-running clocks (VCLK1 and VCLK2), separated by a fixed delay (τ), are used to drive two noise monitors. These input sampling clocks are independent of the I/O interface. Therefore, no fixed phase relationship exists between the sampling clock edges and the toggling edges of the I/O interface clock.

For each fixed delay (τ), and at any random time instance (t), one noise monitor samples and holds the instantaneous voltage present on the power supply, and then converts its analog value to a digital count. Meanwhile, the other noise monitor converts the instantaneous voltage present on the same supply, but at a later time (that is, $(t + \tau)$). These measurements are repeated multiple times to collect enough samples to reconstruct the statistical property. The relative delay (τ) can be swept from 0 to τ_{max}, and for each fixed (τ), the preceding process can

be repeated. As a result, the two noise monitors measure and output the statistical supply voltage data at time instances that differ by the varying delay (τ). Because the supply noise is assumed to be primarily cyclostationary (the statistical properties are repeating), it can be characterized by its auto-correlation:

$$R(\tau) = E\left\{\left[V_{dd}(t)\right]\left[V_{dd}(t+\tau)\right]\right\} \tag{16.1}$$

or auto-covariance:

$$K(\tau) = E\left\{\left[V_{dd}(t) - \overline{V_{dd}(t)}\right]\left[V_{dd}(t+\tau) - \overline{V_{dd}(t+\tau)}\right]\right\} \tag{16.2}$$

Subsequently, the supply noise frequency-domain property is characterized by using its power spectral density (PSD):

$$S_{vdd}(f) = \int_{-\infty}^{\infty} K_{Vdd}(\tau) e^{-2\pi f\tau} d\tau \tag{16.3}$$

Figure 16.14 provides an example of measurement data obtained by using the noise monitor in auto-correlation mode. In this example, the noise generator is used to excite 100MHz of intentional noise. Because the free-running sampling clocks are asynchronous to the system reference clock, all the time instances at which the measurements are carried out are not correlated with the system clock. Therefore, any portion of data in Figure 16.14 can represent the supply noise distribution. Figure 16.15 shows the PSD result. The 100MHz component can be clearly identified, and its noise amplitude derived, from the PSD spectral reading.

Time-domain measurements are as important as frequency-domain measurements. Often, measuring a supply noise transient waveform is more desirable, due to load shifting or power mode switching. In these situations, using the sampling scope mode to conduct a time-domain measurement is preferable. In sampling scope mode, both sampling clocks (VCLK1 and VCLK2) are not free running, as they were in auto-correlation mode. Instead, they are derived from the I/O system clock. In this mode, VCLK1's delay is fixed at '0,' and VCLK2's delay varies, with respective to the I/O system clock. This configuration ensures that the rising edge of VCLK1 is always aligned with the toggling edge of the system clock. Consequently, the noise monitor driven by VCLK1 will always collect the instantaneous supply voltage at the toggling edge of the switching activity. It is not expected to see any supply noise at the instant that current switching starts. The noise distribution is due to other noise sources, such as the device intrinsic random noise and the monitor quantization error [3]. Therefore, the results here can be used to characterize the noise floor performance of the measurement system. Meanwhile, the other noise monitor, driven by VCLK2 with a varying delay of τ, collects the instantaneous supply voltage at a time delayed by τ with respect to VCLK1 (or the system clock). For each fixed delay τ, many samples are collected to reconstruct the time-domain waveform. The mean value of samples, as a function of delay τ, represents the transient waveform of the supply noise.

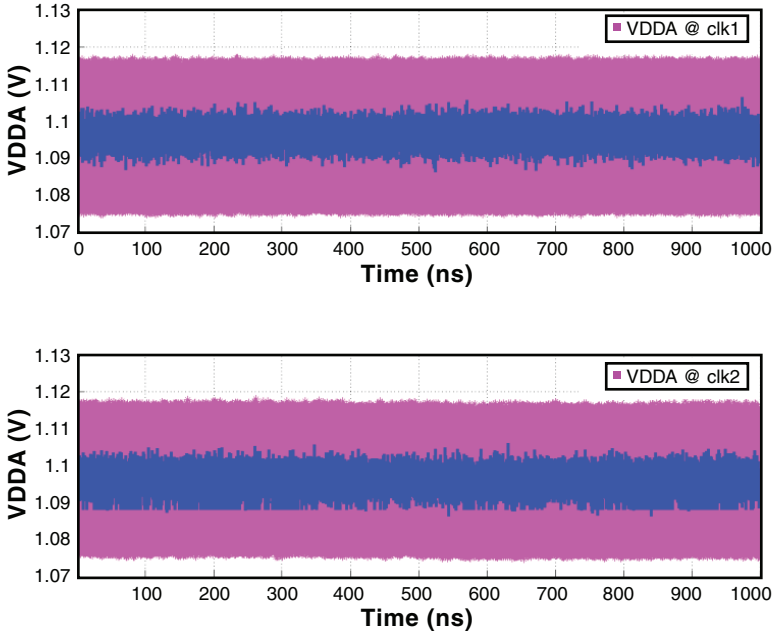


Figure 16.14 Sample Supply Noise Measurement Data in Auto-Correlation Mode

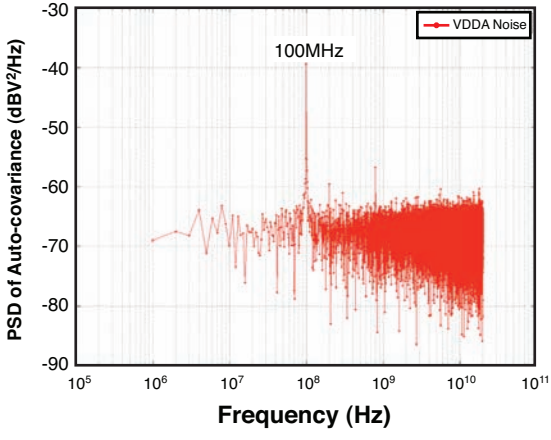


Figure 16.15 Power Spectral Density (PSD) of 100MHz Intentional Supply Noise

Figure 16.16 shows an example of the supply noise data obtained in sampling scope mode. The upper graph depicts the data measured from the monitor clocked by VCLK1, with zero delay with respect to the reference clock. It shows both the noise distribution and the average value of the supply voltage dynamics that occur at the rising edge of the reference. In particular, the mean

value curve represents the DC IR drop from the nominal voltage. The lower graph shows the data measured from the monitor clocked by VCLK2, with a delay varying from 0ns–20ns with respect to the reference clock. The data here shows the voltage noise distribution and its transient waveform in terms of the mean value trajectory. The later part essentially is the time-domain supply noise waveform, reconstructed by operating the noise monitor in sampling scope mode. One can obtain the supply noise characteristics, such as dynamic range, average power, and frequency spectrum, from the time-domain waveform.

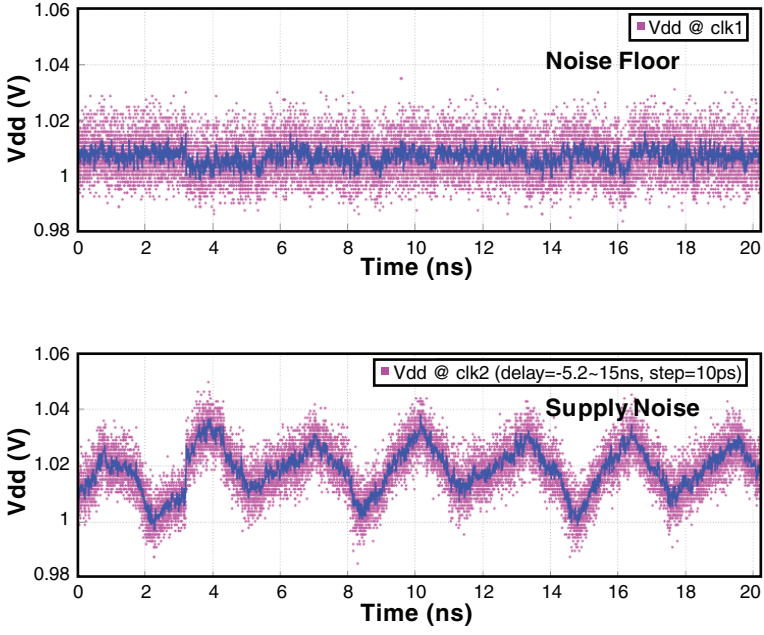


Figure 16.16 Sample Supply Noise Measurement Data in Sampling-Scope Mode

16.5 Advanced Power Integrity Measurements

This section describes a few advanced measurement techniques, based on the previously described on-chip measurement techniques. First, the impedance of the power distribution network (PDN) is considered as seen from the on-chip circuit location. A conventional off-chip measurement technique using a sense line is difficult, particularly in the high-frequency region where noise can be easily filtered out by package.

Second, the jitter sensitivity to power supply noise is considered. Building an on-chip jitter monitor device is both difficult and expensive, so an off-chip measurement approach for the jitter sensitivity measurements is used.

Third, this section introduces a margin sensitivity concept and measurement technique designed to bypass the limitations of the jitter sensitivity measurement. The margin sensitivity can be measured using only on-chip circuitry, in contrast to the jitter sensitivity.

16.5.1 PDN Measurement Technique

Accurate measurement of the on-chip power delivery network (PDN) is another important requirement of system characterization and verification. The on-chip noise monitor and generator (described in the previous section) can be utilized to characterize the on-chip PDN impedance profile, over frequency, using custom measurement scripts and flows developed to support the specific application. (An additional fixture that directly measures the supply current is helpful but optional.) The measurement procedure injects intentional supply current at varied frequencies with a fixed or known strength, while measuring the amplitude of the resulting supply noise voltage. At each frequency, the voltage amplitude from the corresponding supply noise spectrum is read. The PDN impedance can be obtained by dividing the measured voltage with current. A typical on-chip PDN measurement usually requires characterizations from around 10MHz to several hundred MHz, so that the chip-package resonance peak is clearly revealed.

If a current measurement fixture is available, then the measurement accuracy can be further improved in merits of differential measurement method. At a given frequency f, the voltage difference, as a result of the current difference between the two measurements, describes the AC impedance as follows:

$$|Z(f)| = \Delta v(f)/\Delta i(f) = [v_2(f) - v_1(f)]/[i_2(f) - i_1(f)] \tag{16.4}$$

where the subscripts 1 and 2 denote two measurements with two different strengths of intentional current injection. A direct measurement of the average current is required. A simple way to measure this current is monitoring the output of voltage regulator module. Although direct current measurement is useful during the chip development and characterization stage, it is not feasible for in-situ measurements of 3D package systems.

Alternatively, the voltage measurement can be scaled to impedance using the high-frequency measurement data where the impedance is dominated by on-chip decap. The value of the on-chip decap is typically known in advance, and provides the reference impedance for the high-frequency region. By scaling the voltage curve to fit this high-frequency region, the overall impedance profile can be obtained. For example, if the total on-chip decap value is 1nF, then the impedance change from 1GHz to 2GHz is roughly $1/(2 \cdot \pi \cdot 2) = 0.08\Omega$. The PDN impedance curve is generated by scaling the voltage curve, using a constant factor, which produces the desired impedance change over 1GHz to 2GHz.

Figure 16.17 shows the measurement results of the PDN impedance profile for a test system [2]. As shown in the figure, the PDN measurement result is compared to another measurement method using sense lines. As demonstrated in this figure, the two methods correlate well

with each other. The PDN profile peaks at around 130MHz, which is the chip-package resonance frequency of the test system.

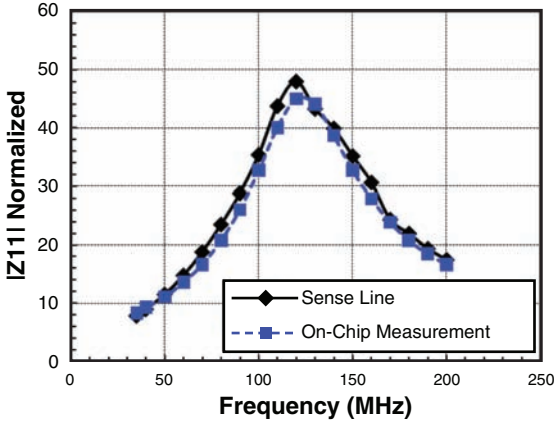


Figure 16.17 On-Chip PDN Measurement Results

16.5.2 PSIJ Sensitivity Measurement

The on-chip noise measurement structure can also be used to characterize the system response to intentionally generated supply noise. For this purpose, the noise generator is used to inject and excite intentional voltage noise in the power supply, with adjustable amplitude at a desired frequency. The excited noise amplitude can be accurately measured, by using either time-domain or frequency-domain methods, as discussed previously. Figure 16.18 shows the relationship between supply noise and jitter. The dominant components in the jitter spectrum are obviously caused by supply noise components at the same frequencies. Therefore, characterizing how much jitter impact the supply noise can introduce to the clock is important. Depending on different PLL types, and clocking architectures, such an impact can also be frequency-dependent. Jitter sensitivity can be defined as:

$$PSIJ(f) = J(f)/V(f) \quad [\mathrm{ps/mV}] \tag{16.5}$$

where $V(f)$ is the supply noise voltage peak-to-peak value at frequency f, and $J(f)$ is the clock timing jitter peak-to-peak value at the same frequency (f). For the sake of convenience, the unit of power supply noise induced jitter (PSIJ) sensitivity is ps/mV. (PSIJ is also known as supply noise induced jitter [SNIJ]). As clearly indicated by its definition, PSIJ sensitivity denotes how much peak-to-peak jitter in ps is introduced by 1mV peak-to-peak supply voltage noise, as a function of frequency.

Figure 16.19 provides an example of measured PSIJ sensitivity profiles over various PVT corners. As the figure indicates, all the PSIJ sensitivity curves exhibit a qualitatively similar

low-pass behavior with the cut-off frequency just above 100MHz. Despite having different roll-offs at different device corners, all the sensitivity curves show tails extending into the high frequency region. Used in concert with the supply noise spectrum information (either by simulation or measurement), these PSIJ sensitivity profiles serve as key link parameters that can be used to predict PSIJ and to identify the major jitter sources.

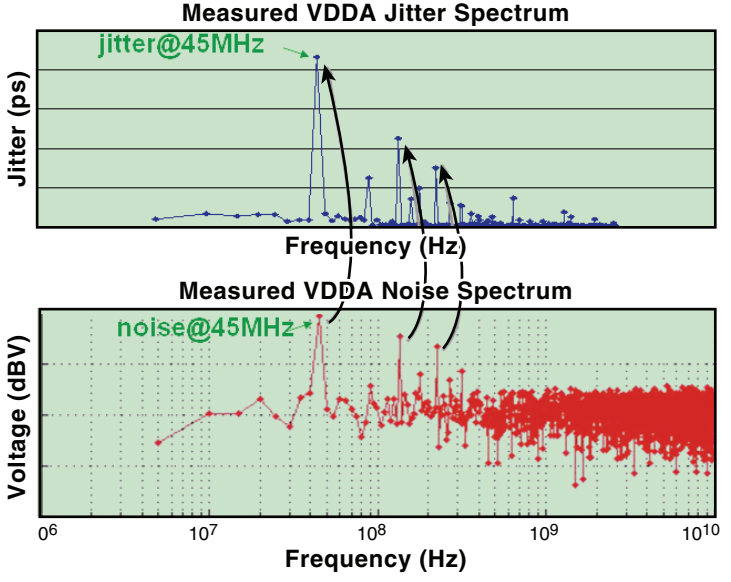


Figure 16.18 Measurement Data Showing Intentional Supply Noise and Its Induced Jitter

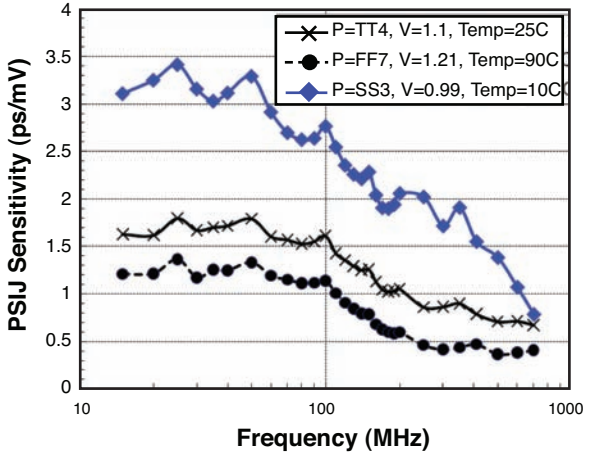


Figure 16.19 Sample PSIJ Sensitivity Measurement Results

16.5.3 Modeling Impact of Power Supply Noise

Although PSIJ sensitivity provides an important circuit parameter, which relates the supply noise to timing error, it, unfortunately, requires an on-chip jitter monitor or external oscilloscope. For in-situ measurements, the preferred method is to characterize the margin sensitivity, called power supply noise induced margin loss (PSIM), which measures the link margin loss due to supply noise, instead of jitter. The link margin loss represents the true system-level impact due to power-supply noise. The link margin accounts for any jitter tracking, cancellation, or amplification. The margin sensitivity can be easily measured using the on-chip timing margin technique described earlier in Section16.1. No additional hardware is needed for this measurement, just supporting scripts. Figure 16.20 shows sample margin sensitivity curves of the memory channel. Note that the margin sensitivity peaks near the power resonant frequency and PLL loop bandwidth.

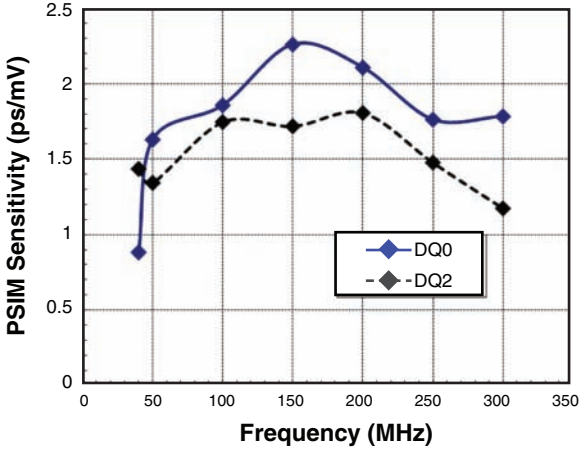


Figure 16.20 PSIM Sensitivity Measurement Using On-Chip Noise and Signal Monitor Features

16.6 Summary

This chapter presents on-chip circuits and techniques that support in-situ measurements. A link simulation flow based on the on-chip wave-capturing feature is demonstrated. These circuits can be built into production devices, and the implementation cost is reasonably small. Using on-chip measurement techniques provides the following benefits:

- In-situ testing and characterization
- Device and system qualification
- At-speed testing
- No testing or measurement equipment

- No measurement parasitic (probes, cables, and so on)
- Includes non-idealities of circuits

References

1. Q. Lin, D. Oh, J. Ren, B. Leibowitz, J. Zerbe, and C. Yuan, "In situ characterization of high speed signaling systems with on chip measurements," presented at the IEC DesignCon, Santa Clara, CA, 2008.
2. D. Oh, H. Lan, C. Madden, S. Chang, L. Yang, and R. Schmitt, "In-situ characterization of 3D package systems with on-chip measurements," in *Proceedings of Electronic Components and Technology Conference*, Las Vegas, NV, Jun. 2010, pp. 1485–1492.
3. E. Alon, V. Stojanovic, and M. Horowitz, "Circuits and techniques for high-resolution measurement of on-chip power supply noise," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 820–828, April 2005.
4. R. Schmitt, H. Lan, C. Madden, and C. Yuan, "Analysis of supply noise induced jitter in Gigabit I/O interfaces," presented at the IEC DesignCon, Santa Clara, CA, 2007.
5. H. Lan, R. Schmitt, and C. Yuan, "Simulation and measurement of on-chip supply noise in multi-gigabit I/O interfaces," in *Proceedings of International Symposium on Quality Electronic Design Conference*, pp. 670–675, Mar., 2008.
6. R. Palmer, J. Poulton, B. Leibowitz, Y. Frans, S. Li, A. Fuller, J Eyles, J. Wilson, M. Aleksic, T. Greer, M. Bucher, and N. Nguyen, "A 4.3GB/s mobile memory interface with power-efficient bandwidth scaling," in *IEEE Symposium on VLSI Circuits*, pp. 136–137, 2009.
7. D. Oh, S. Chang, C. Madden, J.-H. Kim, R. Schmitt, M. Li, C. Yuan, F. Ware, B. Leibowitz, Y. Frans, and N. Nguyen, "Design and characterization of a 12.8GB/s low power differential memory system for mobile applications," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, pp. 33–36, Oct. 2009.
8. P. Larsson and C. Svensson, "Measuring high-bandwidth signals in CMOS circuits," *Electronics Letters,* vol. 29, no. 20, pp. 1761–1762, Sep. 1993.
9. R. Ho, B. Amrutur, K. Mai, B. Wilburn, T. Mori, and M. Horowitz, "Applications of on-chip samplers for test and measurement of integrated circuits," in *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 1998, pp. 138–139.
10. M. Takamiya, M. Mizuno, K. Nakmura, "An on-chip 100GHz-sampling rate 8-channel sampling oscilloscope with embedded sampling clock generator," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2002, pp. 182–184.

11. Y. Zheng and K. L. Shepard, "On-chip oscilloscopes for noninvasive time-domain measurement of waveforms in digital integrated circuits," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 11, Jun. 2003, pp. 336–344.

12. V. Stojanovic, A. Ho, B. Garlepp, F. Chen, J. Wei, E. Alon, C. Werner, J. Zerbe, and M. Horowitz, "Adaptive equalization and data recovery in a dual-mode (PAM2/4) serial link transceiver," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 2004, pp. 348–351.

13. A. Muhtaroglu, G. Taylor, and T. Rahal-Arabi, "On-die droop detector for analog sensing of power supply noise," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 4, pp. 651–660, Apr. 2004.

CHAPTER 17

Signal Conditioning

Jihong Ren and Jared Zerbe

During the past decade, computing platforms have evolved from single-core processors to multi-core processors. Currently, eight-core processors are available in the consumer market. It seems certain the world is entering into a multi-core era. However, without adequate off-chip bandwidth, I/O speed will be the factor limiting system performance. A 256-core processor, assuming four-way SIMD FMACs at 2.5GHz–5GHz, would need terabytes/s of off-chip I/O bandwidth. Package technology has simply not kept pace with the rapid growth in required bandwidth. In 2007, the maximum pin-count for high-performance chips was about 2100; it is expected to grow to only about 5400 pins by 2017 [1]. Consequently, per-pin bandwidth must grow in order to achieve the high-bandwidth that multi-core processors require.

At higher data rates, losses, reflections, and crosstalk severely degrade signal integrity and limit the performance of off-chip links. Thankfully, these effects are linear processes, and on-chip signal processing can compensate for them. In particular, equalization techniques have been widely used to compensate for band-limited channels across many applications. These techniques are not new: Early in 1941, Hendrik Bode received a patent for a broadband receiver that employed equalization, as shown in Figure 17.1 [2]. For years, telephone systems have effectively used equalization for crosstalk and echo cancellation. In the past decade, designers have begun using equalization to compensate for the dispersive losses of high-speed off-chip links, and it has now become an active area of research. In comparison with other applications (such as telephone subscriber systems and wireless communication), equalization for high-speed off-chip links suffers from stringent power and transmit peak-power constraints. This is due to its high level of integration, along with minimum latency and area requirements. These constraints have motivated the design of simple filters to address the most significant signal integrity issues. This chapter focuses on reviewing the current-state-of-the-art analog equalization techniques and coefficient adaptation algorithms for high-speed off-chip links.

As complexity increases in both electrical and optical communication links, interest is growing in implementing the transceiver circuits based on analog-to-digital converters (ADCs) and digital signal processors (DSP) [3] [4]. Therefore, in addition to analog equalization techniques, this chapter also explores the pros and cons of ADC-based receivers and compares these approaches to partial-response DFE (PrDFE) receivers.

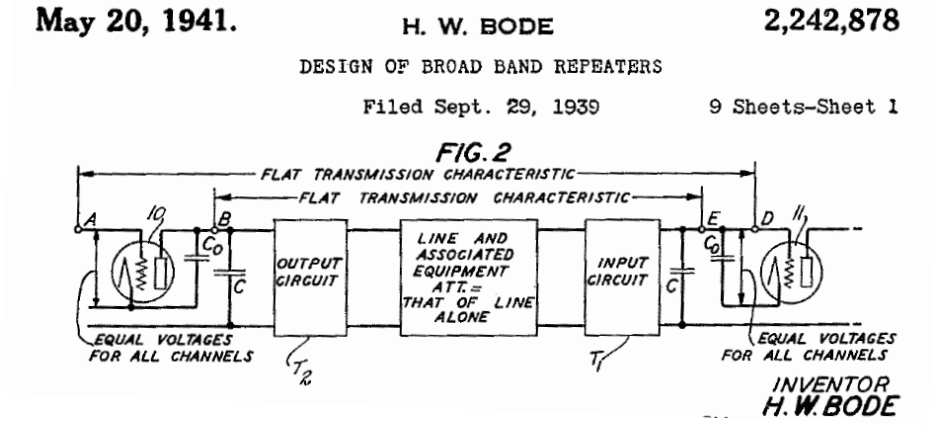


Figure 17.1 Drawing from the Equalization Patent, Filed by H.W. Bode for a Broadband Receiver [2]

This chapter first introduces the concept of the single-bit response (SBR) and reviews various conventional analog equalization techniques, such as transmitter equalization, receiver linear equalization, and decision-feedback equalization (DFE). Then, different equalization adaptation algorithms for the adjustment of the equalizer settings are covered. This is followed by a discussion of the interaction between the clock and data recovery (CDR) loop, and equalization adaptation loop. Finally, the chapter covers ADC-based receiver equalization.

17.1 Single-Bit Response

To understand how signal integrity issues, such as attenuation, dispersion, and reflections, affect the received signal quality, the single-bit response (SBR) is often used. The SBR is the channel response observed at the receiver, when the transmitter sends an unequalized single-bit-wide pulse (Figure 2.11 provides an example). Each circle on the SBR represents the symbol sample time. Here, several phenomena can be observed simultaneously: The spreading of the narrow pulse (beyond a single bit time) shows the dispersion of the channel, whereas the ripples (later in the SBR) are the reflections due to the impedance discontinuities. Any non-zero energy at other bit times in the single-bit response is referred to as inter-symbol interference (ISI). ISI corrupts

the signal received at other bit times, and may lead to bit failure (as illustrated in Figure 2.12). Finally, the reduction in the peak amplitude shows that the SBR was attenuated when it passed through the channel.

17.2 Equalization Techniques

To flatten the frequency response of the channel and remove ISI, equalization can be performed on the transmitter side and/or receiver side. This section provides an overview of three common equalization techniques used in high-speed links: transmitter equalization, receiver linear equalization, and decision-feedback equalization (DFE). The following section discusses the algorithms used to optimize these equalizers.

17.2.1 Transmitter Equalization

Transmitter equalization is one of the simplest ways to compensate for ISI and is now widely in use in high-speed links. Typically, transmitter equalization is implemented with finite impulse response (FIR) filters. Although infinite impulse response (IIR) filters can be more flexible than FIR filters, IIR filters are generally not used for high data rate transmission, because of the difficulty in calculating the IIR recurrence at very high rates. Equalizing transmit FIR filters are relatively easy to implement at very high speed, due to the availability of symbol spaced data at the transmit end. Furthermore, because each transmitted symbol is either a 1 or a 0, for non-return-to-zero (NRZ) signaling, the multiplication with the filter coefficients is relatively straightforward. One simple approach is to implement the filter in the digital domain and use a digital-to-analog converter (DAC) to generate pre-distorted pulses onto the channel [5]. This approach presents the lowest parasitic capacitance to the channel, because the output devices can be the minimum size required to drive the full transmit swing. The DAC approach can also be very flexible, in terms of wave shaping. However, a full DAC implementation suffers from both performance and power/performance issues. The high-throughput requirements and small power budgets (less than 10mW/Gb/s in many cases) of many off-chip link environments make this straightforward implementation unsuitable for most applications.

A more compact and power-efficient approach is to use analog FIR filters with programmable weighted drivers, as shown in Figure 17.2(a). This implementation of a transmit FIR filter is made by a simple wire-OR, connecting high-impedance current-mode drivers, each driven by sequentially delayed weighted versions of the output data. Due to the limited headroom of the practical current sources used in the output drivers, and the maximum voltage process restrictions, the output peak power of such an implementation is usually constrained. Because of this, transmitter FIR filters commonly equalize the channel by attenuating low-frequency components, in order to match the attenuation of the high-frequency components; see Figure 17.2(b). This approach is often called “de-emphasis.”

Looking at transmit equalization in the time-domain, one can see that any ISI that is caused by the current bit can be canceled by sending smaller negative pulses at later times. Figure 17.2(c) illustrates this concept using single-bit responses for a simple two-tap FIR filter. In the

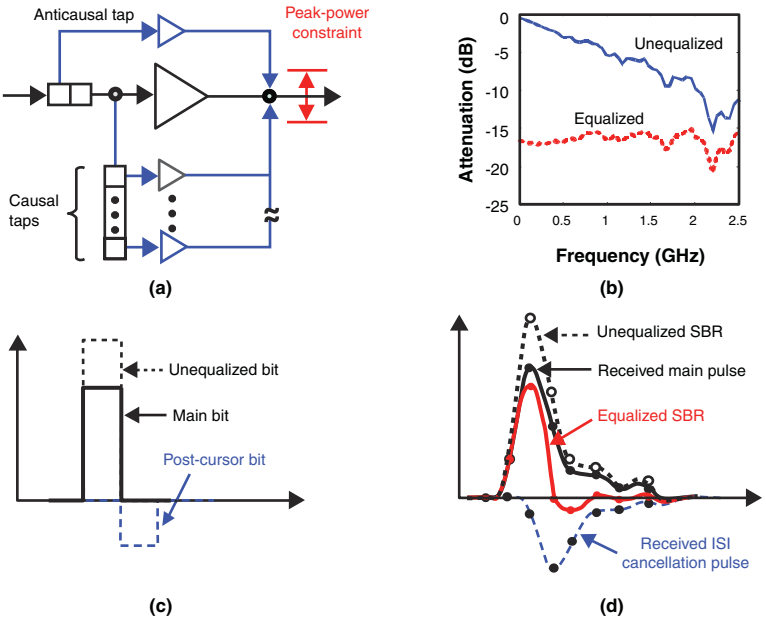


Figure 17.2 (a) Transmit De-emphasis with Peak Power Constraint [6] [7], (b) Unequalized (Blue Line) and Equalized (Red Dashed Line) Channel Responses in Frequency Domain [6], (c) Transmitted and (d) Received Single-Bit Waveforms with a Two-Tap Transmitter FIR Equalizer (The Unequalized Single Bit Response Is Shown by a Dotted Line)

case of no equalization, the driver sends a pulse at time 0. The received unequalized SBR is shown by a dotted line. The circles on the received responses are the symbol-spaced data sampling locations. Due to dispersion, ISI appears at later bit times and corrupts the later bits. To remove the first post-cursor ISI, the ISI tap driver sends a negative pulse with smaller amplitude one bit-time later. The received cancellation pulse is shown as a dashed line. Note that due to transmitter peak power constraint, the ISI cancellation pulse takes away energy from the main tap. Hence, the single bit driven by the main driver is smaller than the unequalized single bit. As a result, the received main pulse has smaller amplitude than the unequalized SBR. Due to the superposition property of LTI systems, the overall equalized single-bit response is the sum of the received main pulse, and the effect of the received cancellation pulse on the current bit. As a result, the first post-cursor ISI is removed from the equalized single-bit response. The relative size of the ISI cancellation pulse to the main pulse is determined by the equalizer coefficients, which are, in turn, determined by the channel characteristics. The details of how to optimally set the equalizer coefficients based on channel characteristics are discussed in the later part of this chapter.

To allow full programmability of the complete flexibility from the analog FIR filters, the most straightforward implementation is to make all the sub-drivers, shown in Figure 17.2(a), the

same size (for example, able to carry the same amount of current). Though simple, this implementation results in significantly more output parasitic capacitance than a transmitter with the same driving capability but no equalization, or a transmitter implemented with digital filters and DACs. This output parasitic capacitance forms a low-pass filter at the output of the transmitter. If not controlled, it can significantly degrade the channel bandwidth. To minimize the parasitic capacitance while preserving programmability, one can split the drivers and share portions of the driver segments among the filter taps [6] [7].

Because transmitters generally do not have direct access to information about the quality of the received signals, the transmitter equalization settings are obtained either by characterization of the channel properties in advance, or with feedback information provided from the receiver through the back channel to the transmitter [8] [9].

17.2.2 Receive Linear Equalization

Placing a linear equalizer at the receiver end avoids the peak power constraint of the transmitter equalizer. Instead of using de-emphasis, the receive equalizer flattens the channel response by amplifying the high frequencies. However, in addition to amplifying the high-frequency components of the received signal, a continuous time linear equalizer (CTLE) cannot discriminate between data and noise—it simply amplifies everything according to its transfer function, and so it can also amplify noise and crosstalk.

Receive equalizers can be implemented with either discrete time FIR filters or with continuous time analog filters. The latter is the more popular of the two approaches as discrete time FIR filters are difficult to implement at high-data rates, and usually consume more power and area than the continuous time version. The disadvantage of a typical continuous time linear equalizer is its limited flexibility. It is not as easy to tune the frequency response of the equalizer to invert the frequency response of the channel.

Figure 17.3(a) shows a differential pair with capacitive degeneration [10]. At high frequencies, the capacitor behaves like a short, and increases the gain of the differential pair. In addition to the pole at the output node, the capacitive and resistive degeneration creates a zero and an additional pole; see Figure 17.3(b):

$$\omega_z = 1\Big/(R_sC_s),\ \omega_{p1} = 1\Big/R_DC_L,\ \omega_{p2} = (1 + g_mR_s/2)\Big/(R_sC_s). \tag{17.1a}$$

The low-frequency gain is:

$$A_v = g_mR_D\Big/(1 + g_mR_s/2). \tag{17.1b}$$

Thus the *boost factor* (that is, the amount of high frequency peaking) is determined by the location of ω_z relative to ω_{p1}, as shown in Figure 17.3(b). By tuning the source-degeneration capacitor (C_s) and resistor (R_s), the location of the zero can be effectively tuned, as well as the low-frequency gain, thereby matching the frequency response of the equalizer to the channel

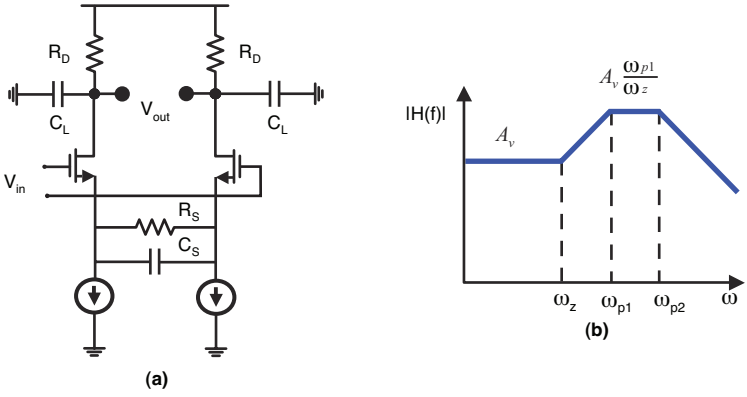


Figure 17.3 (a) Differential Pair with Capacitive and Resistive Source Degeneration [11], (b) Frequency Response

characteristics. However, because tuning R_s and C_s changes the overall shape of the frequency response, perfectly inverting the channel characteristics with a CTLE becomes challenging. Moreover, they are also sensitive to PVT variations. Finally, note that the gain-bandwidth product of the source-coupled differential pair still limits the maximum gain. There is a fundamental trade-off between the low-frequency gain, the boost factor, and the bandwidth of the stage [11]:

$$A_v \frac{\omega_{p1}}{\omega_z} \omega_{p2} \approx {g_m}\Big/{C_L}. \tag{17.1c}$$

To achieve a higher boost factor, multiple stages can be cascaded [11], but this can result in greater low-frequency loss, resulting in degradation in sensitivity.

Another approach to achieving additional gain is inductive peaking, implemented by placing an inductor in series with the load resistor, in order to block the current flow at high frequencies. The result is that most of the high-frequency current flows into the load capacitor, resulting in a sharper output voltage transition. However, this approach is area intensive to implement with on-chip wiring.

17.2.3 Decision-Feedback Equalization

Compared to linear equalizers, decision-feedback equalizers (DFE) can cancel large amounts of ISI without the noise amplification associated with linear equalizers. DFE works especially well for channels with impedance discontinuities that result in multiple reflections. The basic idea behind DFE is to utilize past symbols, determined from previous decisions, to cancel ISI (see Figure 17.4) by simply subtracting the interference on the current received symbol from the previous symbols. Because the DFE feedback removes ISI by direct subtraction, each DFE weight is an estimation of the ISI contribution from the previous bit. Analog pulses, synthesized from previous symbol decisions and multiplied by DFE tap coefficient weights, are then current-summed

with the received signal, as shown in Figure 17.4 [12] [13]. The resulting equalized signal is input to the data sampler, which makes future symbol decisions. Consequently DFEs are causal; for example, they cannot correct for precursor ISI. Precursor ISI is the interference from "future" symbols (that is, symbols that will arrive after the decision instant). However, DFEs can be combined with feed-forward equalizers (either a transmitter equalizer or a receiver equalizer or both), in order to eliminate precursor ISI beforehand.

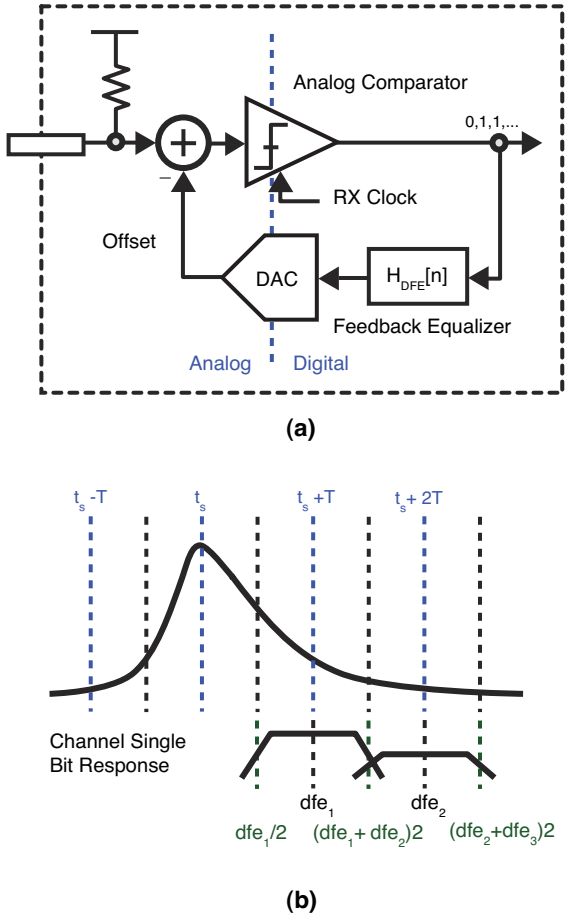


Figure 17.4 (a) A Conventional DFE Receiver (b) Timing Alignment between DFE Feedback Pulses and Incoming Data Pulse

It is worth noting that the signal path around the DFE loop crosses the analog boundary twice: once through the analog comparator, and once through the feedback path generating the analog offset from the prior bits. These two conversions make closing the timing around the loop

to within a single bit-period extremely difficult. Figure 17.4(b) shows an example of the timing alignment between the DFE feedback pulses, and the channel's single-bit response. The DFE clock can be adjusted so the transitions of the DFE correction pulses are phase aligned with the transitions of the input data [12]. With this alignment, the DFE correction pulses reach nearly full swing at the eye center, and add half swing at the edges of the data; see Figure 17.4(b). The subtraction of the two gives the single-bit response for the DFE-equalized channel. Therefore, this requires resolution of the bit, multiplication by coefficient, and analog summation in half a bit time! This makes the conventional DFE extremely hard to implement in high-speed serial link applications. However, correcting ISI from symbols that are far away from the current sample do not have this problem, making the conventional DFE architecture suitable for correcting long latency reflections, which can often be caused by impedance discontinuities in the signal path.

One way to solve the feedback loop timing issue is to simply unroll the loop, making two decisions at each cycle [6] [12] [14], as shown in Figure 17.5. This DFE receiver architecture is referred to as *partial response* DFE (PrDFE). Two slicers are used at the receiver; one assumes the previous bit was a 1, and the other assumes it was a 0. After the decision for the previous bit is available, the correct answer is selected. The two samplers are offset in voltage by $\pm\alpha$, which assumes that the impact of the previous 1 bit on the current bit is $+\alpha$ (or $-\alpha$ for a 0 bit).

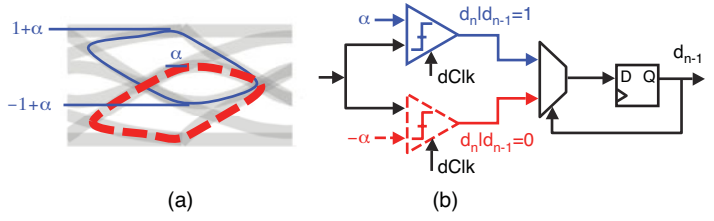


Figure 17.5 (a) Partial Response DFE (PrDFE) via Loop Unrolling (b) PrDFE Eye Diagrams. Upper Eye Preconditioned with Previous 1 Bit; Lower Eye Preconditioned with Previous 0 Bit

Figure 17.5 contains example PrDFE eye diagrams. Even though the two slicers see the same incoming signal, the signals that the two slicers care about are filtered by the previous bit, as indicated by the circles and squares in Figure 17.5. For example, the upper slicer, with the α threshold, effectively sees the eye preconditioned with the previous 1 bit, so it only sees the signals around $1+\alpha$ and $1-\alpha$.

Note that even with PrDFE the feedback loop must be closed within a single bit-time. However, instead of crossing the analog-to-digital boundary twice (as with the conventional DFE architecture), the feedback loop now only involves digital circuitry. The critical timing path includes the multiplexer selection to the output delay ($t_{mux,so}$), the setup time (t_{su}), and the clock to Q delay ($t_{c,q}$) of the flip-flop:

$$t_{c,q} + t_{mux,so} + t_{su} < 1\,\text{bit time}. \tag{17.2}$$

The PrDFE approach can be extended to accommodate additional taps, but the number of slicers required grows as 2^N. Given this exponential growth in the number of slicers, practical designs are limited in the number of PrDFE taps without excessively loading the input. In Section 17.5, equivalences between an ADC-based DFE receiver and PrDFE receiver are discussed, and a reduced-sampler PrDFE (RS-PrDFE) architecture is shown that uses fewer than 2^N slicers to realize additional taps.

Instead of extending PrDFE to accommodate additional taps and incur excessive loading at the input, another option is to combine the strengths of each PrDFE and conventional analog DFE; for example, only use PrDFE for the first few taps that have the stringent timing requirements and use conventional analog DFE for later taps where the timing requirements are less stringent. Leibowitz et al. use such an approach [12], where PrDFE is used only for the first tap and the rest of the 10 DFE taps are implemented with classic analog DFE.

Because DFE is based on previous decisions, it, unlike linear equalizers, can suffer from error propagation [27]. An erroneous decision could result in a burst of errors. The problem of error propagation can be controlled by system choices, such as receive linear equalization, keeping the feedback filter short, and constraining the magnitude of contributions from the feedback filter. Moreover, current high-speed links are generally designed to operate with extremely low bit error rates, which ultimately reduce the impact of DFE error propagation on practical high-performance wireline systems.

17.3 Equalization Adaptation Algorithms

To properly improve system margins, correctly tuning the equalizer frequency responses to invert channel characteristics and remove ISI is important. Three general approaches exist for setting equalizer coefficients: The first approach can be called "look up table and forget," in which the channels are characterized in lab, establishing a best set of filter coefficients that are then used for all links operating over that channel. The second approach is "adapt once and forget." In this approach a channel is characterized in situ at power-up and a single set of filter coefficients are derived for that channel. Using this method takes manufacturing variations and component aging into account due to its "in-situ" nature. The third method is "continuous adaptation," which also considers environmental variations, but also accounts for temporal effects. Zerbe et al. [15] compare these three methods using high-performance backplanes, including manufacturing and environmental variations, such as temperature, humidity, and Vdd. The last two methods ("adapt once and forget" and "continuous adaptation") significantly outperform the "lookup table and forget" method at higher data rates, because they take manufacturing variations into account [15]. Although the results of the last two methods are comparable, continuous adaptation performs slightly better at medium data-rates (6.4Gb/s) and is expected to be required at high data rates (>10Gb/s).

The balance of this section introduces the classic *least means squares* (LMS) algorithm [16]. Three practical adaptive algorithms that require minimum hardware support and are suitable for high-speed I/O interfaces (such as memory channels and backplane channels) are then

reviewed. The algorithms are suitable for use with either the "adapt once and forget" or "continuous adaptation" methods. The first two adaptation algorithms are variants of the *sign-sign least mean squares* (SS-LMS) algorithm, which forces ISI to zero when ISI dominates other noise sources [6] [17–20]. The third algorithm is based on minimizing the bit-error-rate (BER), in order to maximize the receiver voltage margin [21] [22]. These algorithms are applied to transmitter equalization and their performance is compared. In the next section, their application to decision feedback equalization will also be explored, as well as the interaction between the CDR loop and EQ adaptation loop.

17.3.1 Least Mean Squares Algorithm

The least mean squares (LMS) algorithm was first introduced by B. Widrow in the 1960s [16]. Figure 17.6 shows how it can be applied to adaptive linear receive equalizers. The adaptation goal is to minimize the mean-square error $E(e(n)^2)$ between the equalized received signal, and the ideal signal:

$$\hat{x}(n) = \underline{w} P \underline{x}(n) \tag{17.3}$$

$$\underline{u}(n)^T = P\underline{x}(n) \tag{17.4}$$

$$e(n) = \hat{x}(n) - x\,(n-\Delta) \tag{17.5}$$

where $x(n)$, $u(n)$, and $\hat{x}(n)$ are the transmitted signal, channel output, and equalized received signal, respectively. Symbols with underscores denote vectors. Capitalized symbols denote a matrix. $\underline{w}$ is the filter coefficient vector; and P is the convolution matrix, based on the pulse response (SBR) of the channel. Δ is the channel delay.

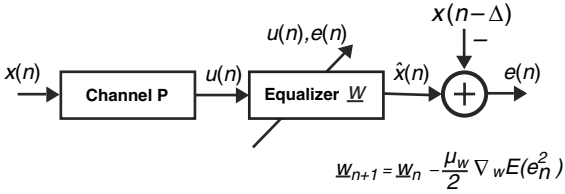


Figure 17.6 Adaptive Linear Rx Equalizer Using LMS

LMS is a "steepest descent" algorithm that follows the negative gradient of the mean-square error between the equalized receive signal and the expected "ideal" signal. However, in general, getting expectation is hard. Typically, the expectation is approximated with a running average. In the simplest case, the mean is approximated with the instantaneous value, as shown in Equations (17.6) and (17.7). Ultimately, the equalizer adapts until the received unequalized signal $u(n)$ and error $e(n)$ are orthogonal to each other, and any future update is useless.

$$\nabla_w E(e_n^2) \approx \nabla_w e_n^2 = 2e_n \frac{\partial e_n}{\partial w} = -2e_n \underline{u}(n) \tag{17.6}$$

$$\underline{w}_{n+1} = \underline{w}_n - \frac{\mu_w}{2} \nabla_w E(e_n^2) \approx \underline{w}_n - \frac{\mu_w}{2} \nabla_w e_n^2 = \underline{w}_n + \mu_w e_n \underline{u}(n). \tag{17.7}$$

Direct implementation of the LMS algorithm requires the amplitudes of the error $e(n)$, and the unequalized signal $u(n)$. Therefore, it requires ADCs, in order to digitize the signal levels which can be very costly for high-speed links. A commonly used alternative is a variant of the LMS algorithm, Sign-Sign-LMS (SS-LMS) [17], which only uses the sign of the information:

$$\underline{w}_{n+1} = \underline{w}_n + \mu_w \operatorname{sgn}(e_n) \operatorname{sgn}(\underline{u}(n)). \tag{17.8}$$

17.3.2 Zero-Forcing SS-LMS Algorithm

Three problems are associated with using (17.8) directly for transmitter equalizer adaptation: First, the update information must be sent back to the transmitter from the receiver in order to adjust the transmit equalizer coefficient settings. This problem can be solved by using a form of back channel (for example, Stojanovic et al. [9] use a low-bandwidth back channel, via the common-mode of the differential link, to send update information back to the transmitter). Alternately, spare bandwidth on an adjacent symmetric channel in the return direction can be used.

Second, the unequalized signal ($u(n)$) is not readily available, because the receiver only observes the equalized signal. In this case, the simplest solution is to give up the minimum mean square error (MMSE) solution, but choose instead to zero-force only ISI, by substituting $\hat{x}(n)$ for $u(n)$:

$$\underline{w}_{n+1} = \underline{w}_n + \mu_w \operatorname{sgn}(e_n) \operatorname{sgn}(\underline{\hat{x}}(n)). \tag{17.9}$$

Note that, if the error $e(n)$ is no longer correlated with a neighboring bit, the ISI caused by the bit must have been zero-forced. In this case, the adaptation procedure reaches steady state, and the average update is zero. Compared to this zero-forcing solution, MMSE takes into account both noise and ISI. Therefore, if there is significant high-frequency noise, MMSE may result in less high-frequency peaking and its resultant noise amplification compared with zero-forcing. Most often ISI remains as the dominate noise source in high-speed links and is more significant than other sources such as random noise. In most cases, the zero-forcing solution is close, if not exactly the same as the MMSE solution.

The transmitter equalizer is generally subject to a peak power constraint:

$$\|\underline{w}\|_1 = 1 \tag{17.10}$$

where $\|\cdot\|_1$ is the l_1-norm of $\underline{w}$, also known as the sum of the absolute value of the elements in $\underline{w}$. This peak power constraint indicates that in transmit equalization, non-zero equalization taps

simply take energy away from the main tap. The target signal level actually changes as filter coefficients are adapted. To solve this problem, a second adaptation loop must be introduced to track the data level *dLev*. *dLev* is simply the expected signal swing in the absence of ISI. The instantaneous difference between *dLev* and the actual data level is thus the error information used in the adaptation; see Figure 17.7(a).

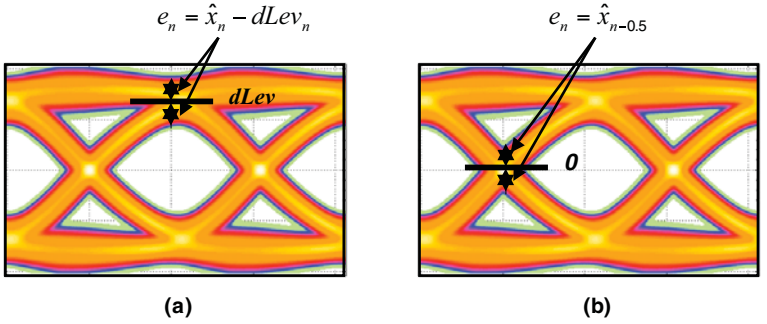


Figure 17.7 (a) Data-Based Adaptation (b) Edge-Based Adaptation

The dual-loop zero-forcing sign-sign least mean square (LMS) algorithm [6] is:

$$\underline{w}_{n+1} = \underline{w}_n + \mu_w \,\mathrm{sgn}(\underline{\hat{x}}_n)\,\mathrm{sgn}(e_n) \tag{17.11}$$

$$dLev_{n+1} = dLev_n - \mu_{dLev}\,\mathrm{sgn}(e_n). \tag{17.12}$$

This algorithm is based on the error information collected at data sampling time and tries to zero-force the ISI at that time. The shape of the eye and any data-dependent jitter are not taken into account. Edge-based adaptation has been proposed as an alternative to data-based equalization. Edge-based adaptation minimizes ISI at the data transition time, and in so doing, achieves improved voltage margin at the eye center. Edge-based adaptation uses the error information collected at the transitions of the data; see Figure 17.7(b). Generally, this error information is already available for use in timing recovery so no additional adaptive sampler is needed as compared with the data-based adaptation.

In further contrast to the data-based adaptation algorithm, the edge-based adaptation only requires one adaptation loop, because the desired signal level is 0 at the edge transition, as shown in Figure 17.7(b). The edge-based SS-LMS algorithm [18–20] is then:

$$\underline{w}_{n+1} = \underline{w}_n + \mu_w \mathrm{sgn}(\underline{\hat{x}}_n + \underline{\hat{x}}_{n-1})\mathrm{sgn}(e_n) \tag{17.13}$$

where e_n is the error signal, shown in Figure 17.7(b), which is simply the edge sample in this case. The adaptation is only done when the data transitions ($\hat{x}_n + \hat{x}_{n-1} = 0$) Therefore, the edge-based SS-LMS algorithm zero-forces ISIs at edge time.

Figure 17.8 shows the single-bit responses of a sample channel with no equalization, data-based equalization (left), and edge-based equalization (right), using a three-tap transmitter FIR equalizer. Note that data-based equalization zero-forces ISI at data-sample times, whereas edge-based equalization zero-forces ISI at edge-sample times. With this channel, because the data ISI and edge ISI are positively correlated, zero forcing the edge ISI results in a significant reduction in data ISI. However, this is not necessarily always true. Figure 17.9 shows the single-bit responses of the same channel, when a strong linear equalizer is included at the receiver. In this case, edge ISI is negatively correlated with data ISI. Edge-based equalization cancels edge ISI, but also introduces more ISI at the data-sample time, thus increasing the BER.

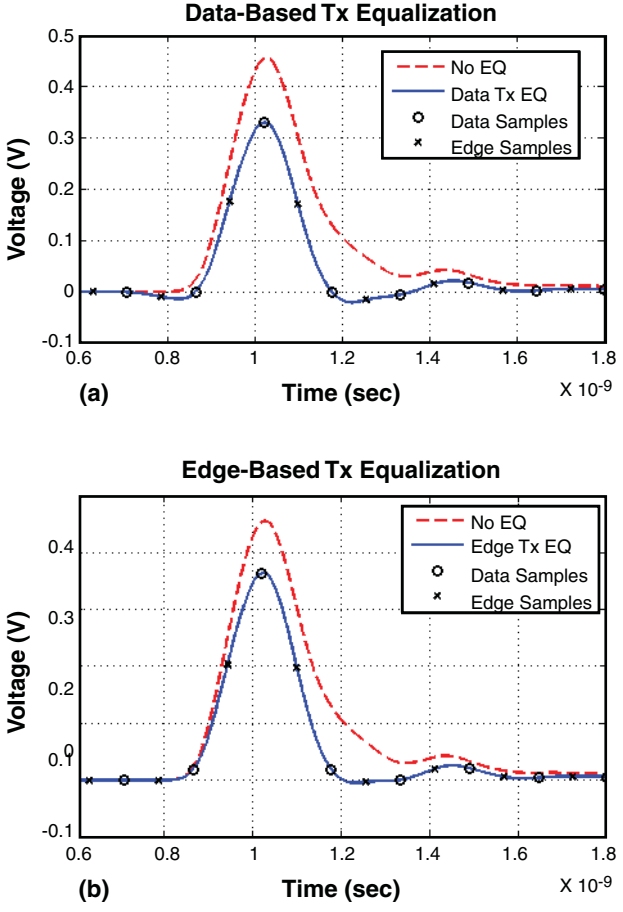


Figure 17.8 Single-Bit Responses for Data-Based (top), and Edge-Based (bottom) Transmitter Equalization (The Transmitter Equalizer Has One Precursor Tap and One Postcursor Tap)

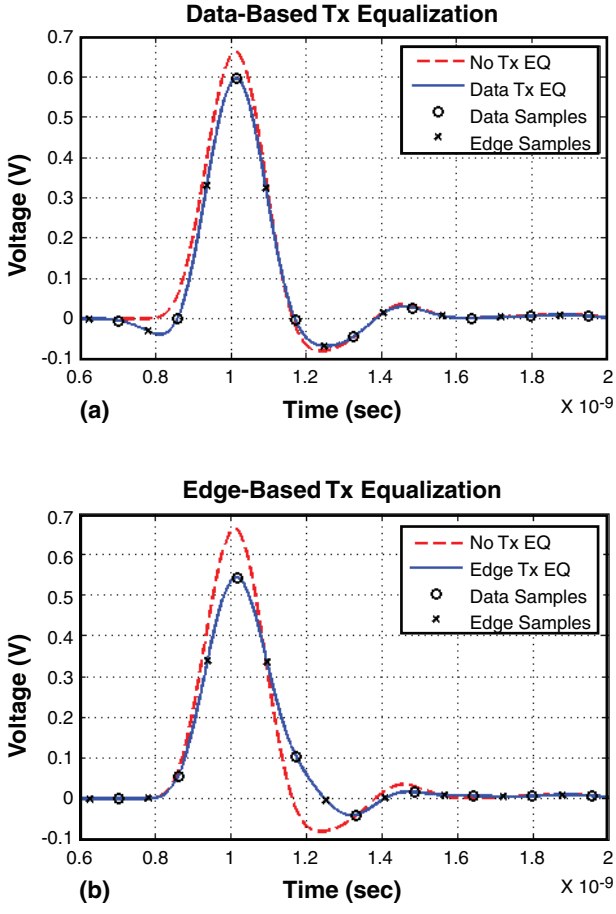


Figure 17.9 Sample Channel with a Strong Linear Equalizer Included in the Receiver to Boost Edge Rate

17.3.3 minBER Algorithm

During SS-LMS adaptation of a transmit equalizer, the main tap value must decrease when the other taps increase to keep the transmitter output swing in compliance. ISI is canceled by taking energy away from the main bit. Better link performance may be obtained by maximizing the eye height under this peak power constraint instead of zero-forcing ISI [22] [23]. This section introduces minBER algorithm, a modified gradient descent algorithm, which achieves eye height maximization. This section first introduces the minBER algorithm and then presents its application to transmit equalization.

17.3.3.1 Algorithm

The worst-case ISI is found by choosing the signs of adjacent data bits, in order to result in a maximum sum of negative and positive interference terms, when the current data bit is 1 and –1, respectively. Given the symbol-spaced single bit response $\underline{p}_{\Delta t}$ at phase Δt, let $p_{\Delta t}^{M}$ be the main cursor, and $p_{\Delta t}^{ISI}$ be a vector that contains all the ISI terms of the single-bit response at phase Δt (this notation extends to the convolution matrix $P_{\Delta t}$ as well). The lowest possible undershoot is given by $p_{\Delta t}^{M} - \|p_{\Delta t}^{ISI}\|_1$. Therefore, the eye height maximization can be formulated under the peak power constraint, as the following linear programming problem:

$$\max_{w} P_0^M \underline{w} - \|P_0^{ISI}\underline{w}\|_1 \quad s.t. \quad \|\underline{w}\|_1 = 1. \tag{17.14}$$

Eye height maximization has been shown to outperform zero-forcing algorithms for setting coefficients for transmitter equalization [21-24].

The minBER algorithm [21] [22] effectively solves this optimization problem based on a modified gradient descent algorithm (Figure 17.10 illustrates this adaptation procedure). It uses information about BER degradation to perform the adaptation. During adaptation, each tap is changed only one step in the direction of decreasing BER before moving to adjustment of the next tap. The algorithm determines the direction in which to modify each coefficient (increase, decrease, or no change) after a two-thirds majority vote; this is done to reduce the effect of random noise. The process continues to iterate through all the taps until the coefficients converge.

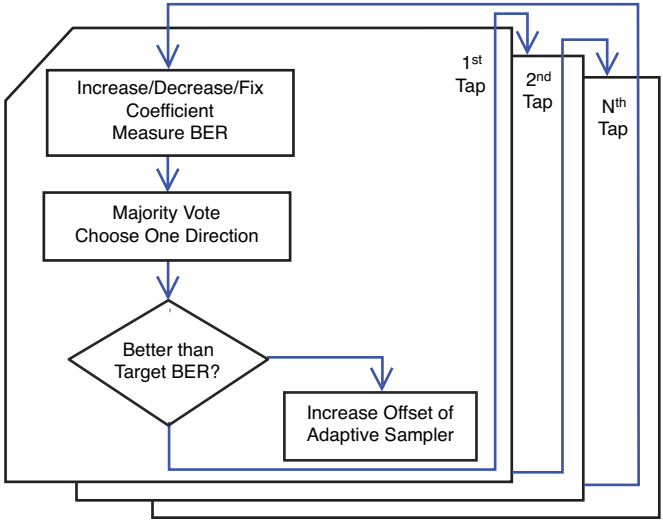


Figure 17.10 Flow Diagram of the minBER Algorithm

The minBER algorithm maximizes the voltage margin at the target BER. One can adapt to the target BER by using a high target BER initially, and then reducing it when no errors are measured. This enables equalizer adaptation for channels with low BER, and speeds up the adaptation process as a low BER requires a long time to measure. To further speed up convergence, a variable coefficient step size can be used. In this version of the algorithm, the adaptation is initialized with large coefficient steps. The step size of the coefficients is then modified depending on whether the coefficients are changing. Infrequent changes in the coefficients will lead to making smaller coefficient steps. The initial large coefficient step size also helps to keep the adaptation from becoming trapped in local minima. The reduction to small coefficient steps results in smaller dithering when the adaptation does converge.

17.3.3.2 Application to Transmit Equalization

Figure 17.11 shows the measured contour plots of received voltage margin at a target BER of 10^{-6}, versus pre- and post-tap coefficients for two different channels running at 6.25Gb/s. Because of the transmit peak-power constraint, the sum of all taps is fixed. The curves show the coefficient paths during the adaptation methods, with the bold line representing the final 50 iterations. These measurements indicate that the SS-LMS adaptation converges to the suboptimal region, whereas minBER adaptation with variable coefficient step sizes results in final coefficients around the optimal region and an improvement of 10mV–20mV of additional voltage margin at the target BER. Figure 17.11 also shows that the minBER adaptation can be trapped in a local minimum or a flat region in the contour when only a minimum step size is used. Variable step sizes not only address this problem, but also serve to speed up convergence.

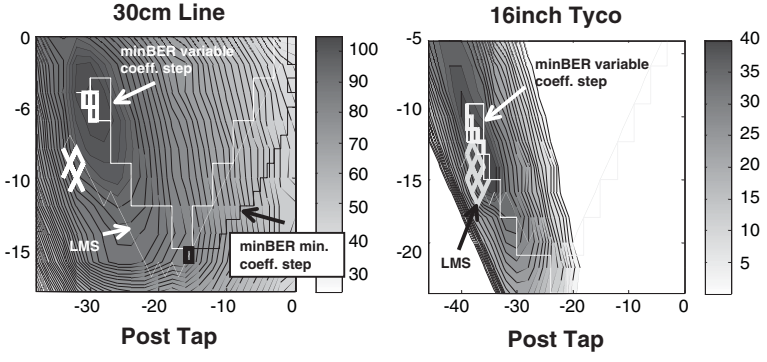


Figure 17.11 Measured Voltage Margin for Various Tx-FIR Pre-Tap and Post-Tap Coefficients. SS-LMS and minBER Adaptation Paths Shown for Two Different Channels

The minBER algorithm converges to a final value in roughly the same number of iterations as a traditional SS-LMS algorithm (see Figure 17.12). Unlike SS-LMS, however, because BER is measured at each step, the actual adaptation time depends on the target BER, and on the number

of taps. For example, for a three-tap Tx-FIR, only two of the three coefficients need to be updated due to transmit peak power constraint. For each coefficient, the voltage margin is measured three times for each increased/decreased/fixed coefficient. Therefore, 18 measurements are done. If the voltage margin is measured at 10^{-4} BER, the total number of bits required is roughly 1.8e5 for each minBER iteration. For SS-LMS, typically 256 measurements are done for each iteration to average out noise effect. Therefore, the minBER algorithm could be about 700X slower than the SS-LMS algorithm, when the target BER=10^{-4} for a three-tap Tx-FIR. The speed of the adaptation can be improved by using a SS-LMS adaptation first to get close to the optimum region, and then using the slower minBER algorithm to fine-tune the coefficients.

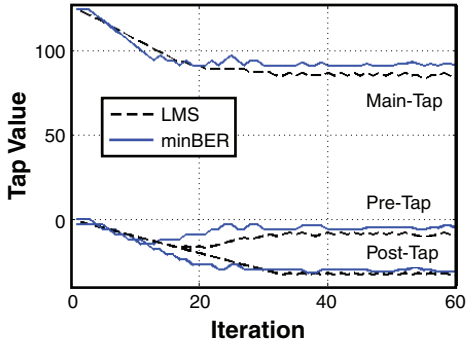


Figure 17.12 Convergence of Coefficients (MinBER Adaptation Only Shows the Coefficient After Each Decision)

Figure 17.13 shows eye diagrams for each method after transmit adaptation as measured by a sampling oscilloscope. Compared with the eye diagram from the SS-LMS adaptation, minBER adaptation shows a larger eye opening despite larger ISI at the data sampling point.

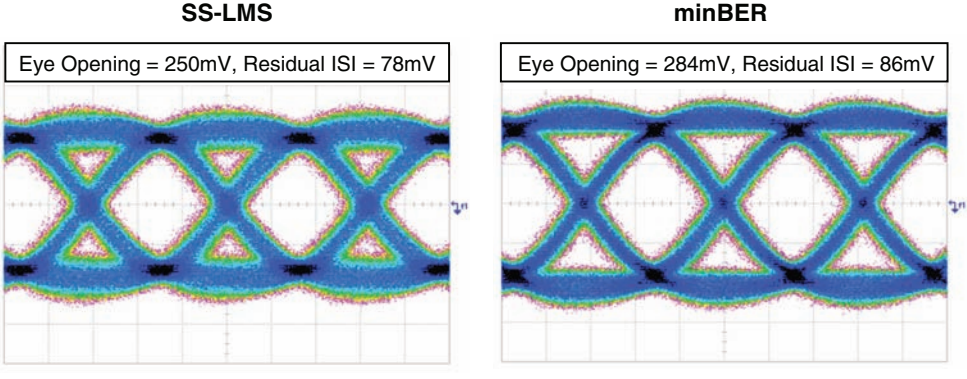


Figure 17.13 Eye Diagram of Signal Received by Oscilloscope Transmitted Across a Tyco 16-inch Channel after Tx-FIR Equalization

17.4 CDR and Equalization Adaptation Interaction

Adapting equalization coefficients will change the overall channel response (as seen by the receiver sampler), whereas operation of a clock and data recovery loop (CDR) will track any changes affecting phase position. (Chapter 10, "Clock Models in Link BER Analysis," shows how to derive the CDR nominal locking position for a given channel.) The CDR shifting the sampling position changes the amount of ISI that is seen by the samplers. This, in turn, changes the equalization tap weights through the adaptation engine [24] [25]. In this section, the impact of this interaction between equalization adaptation and CDR loops on link performance is examined using an architecture that contains both transmitter equalization and receiver DFE. Specifically, it will be shown that with DFE canceling postcursor ISI, canceling precursor ISI with symbol-spaced transmitter equalization degrades rather than improves performance for most channels. This is due to the interaction between equalization adaptation and CDR loops, coupled with the transmitter peak-power constraint.

To exploit the complementary strengths of Tx-FIR and DFE, the design shown in Figure 17.14 shows an example architecture using a Tx-FIR to remove precursor ISI, and a DFE to remove postcursor ISI. As previously discussed in Section 17.2.1, a Tx-FIR must reduce ISI at the expense of signal swing, due to its inherent peak-power constraint. In contrast, a DFE has no peak power constraint, and removes ISI without reducing the signal swing. Due to the causal nature, however, a DFE cannot remove precursor ISI. The receiver uses a 2x over-sampled CDR circuit to recover timing information. It unrolls the first DFE tap by using partial response DFE (PrDFE) to avoid the tight feedback loop of conventional DFE. To reduce the impact of edge ISI on timing recovery, the receiver employs PrDFE on the edge samplers as well.

For the sake of simplicity, assume α_{edge} is set to 0. In this case, PrDFE edge samplers reduce to simple NRZ edge samplers. One can extend the following analysis for nonzero α_{edge}. In this case, CDR nominally locks to the phase where the mean of the edge samples is zero (see Chapter 10):

$$e_0 - e_1 = 0 \tag{17.15}$$

where e_k denotes $p(t_s - T/2 + kT)$. $p(t)$ is the single-bit response (SBR) of the raw channel; t_s is the main cursor location, and T is the bit time. When a precursor-only Tx-FIR is applied, the mean of the edge samples for the rising transitions at the original phase (before the CDR update) is:

$$e'_0 - e'_1 = \sum_{i=-m}^{0} w_i \left(e_{-i} - e_{1-i}\right) = \sum_{i=-m}^{-1} w_i \left(e_{-i} - e_{1-i}\right) \tag{17.16}$$

where w_i is the ith Tx-FIR tap. For dispersive channels, the precursor ISI is generally positive, and dominated by the first precursor (see Figure 17.15). Consequently, the summation in (17.16) is dominated by $w_{-1}(e_1 - e_2)$, which is negative for typical channels. After Tx equalization, the

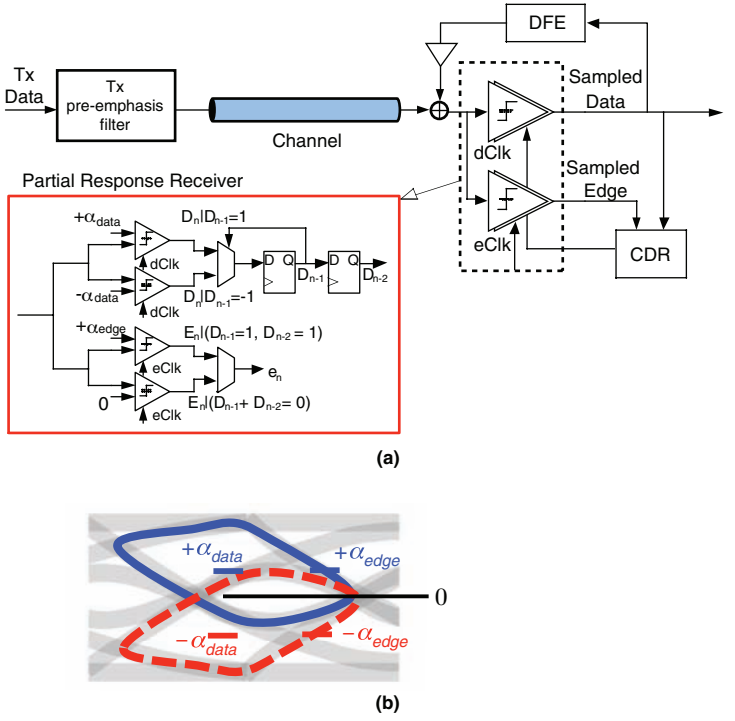


Figure 17.14 (a) Transceiver Architecture with Transmitter Equalization and Receiver DFE (First Tap as PrDFE). (b) PrDFE Eye Diagram. For timing recovery, the edge samplers can be placed at two of the three levels indicated. Data filtering is used for valid edge samples. By default, α_{edge} is set to α_{data} [24].

mean of the edge samples at the same phase becomes negative. The CDR responds by delaying its phase position. As illustrated in Figure 17.15, this delayed CDR position increases the precursor ISI seen by the data sampler. The SS-LMS algorithm tries to zero-force the increased precursor ISI, thus spending more energy in precursor ISI cancellation and lowering the main cursor even more. After adaptation converges, the precursor ISI is canceled, at the cost of more than half of the main cursor for the example channel.

Figure 17.16 shows measured voltage margin contours versus CDR phase, and the transmitter pre-tap value, for a 16" backplane channel. The link starts with a pre-tap value of 0 and converges to a suboptimal point, with too much pre-tap magnitude, and a CDR phase that is too late. The link's operating point, with SS-LMS adaptation, is outside the contour plot with a pre-tap value of approximately –0.2, and CDR phase index of roughly 88.

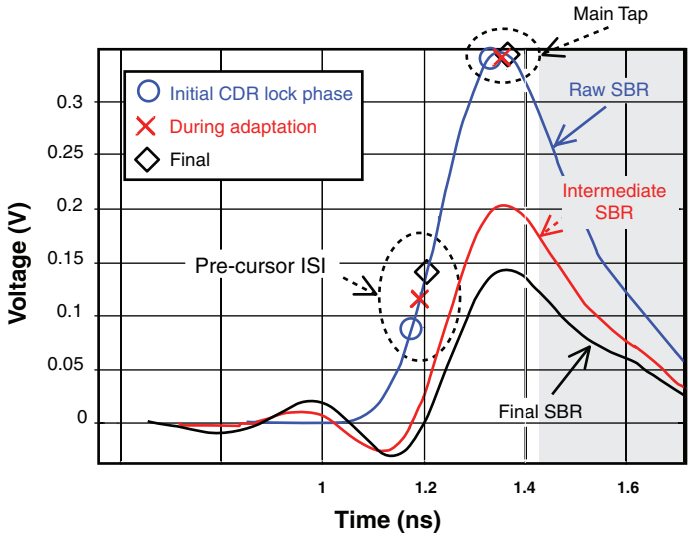


Figure 17.15 Simulated Single-Bit Responses (SBR) Before, During, and After Adaptation. The shaded area indicates postcursor ISIs are taken care of by DFE.

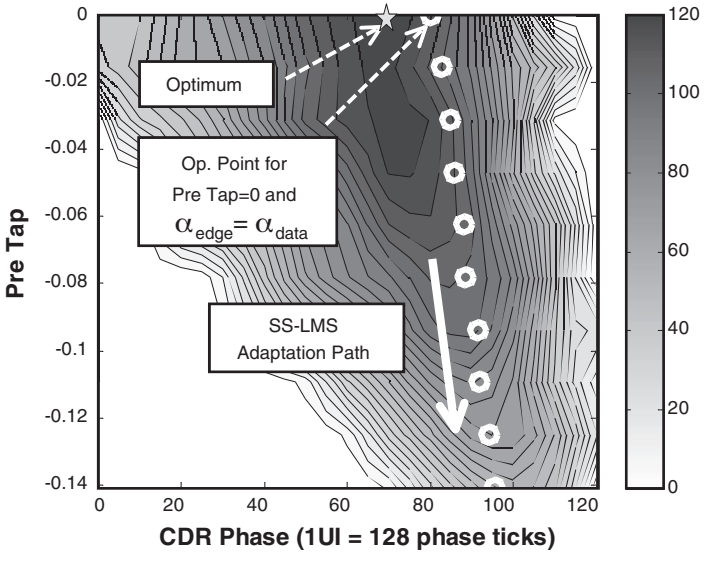


Figure 17.16 Measured Voltage Margin vs. Tx FIR Pre-Tap Value and CDR Phase for a 160 Backplane Channel at 6.25Gb/s

Interestingly, Figure 17.16 shows that the optimal setting for the pre-tap is 0 (no precursor cancellation). When postcursor ISI is canceled, the voltage margin is roughly $d_0 - \sum_{i=-\infty}^{-1}|d_i|$, where d_k denotes $p(t_s + kT)$. Ten different backplane channels with different channel characteristics (–15dB to –35dB attenuation at Nyquist frequency) were simulated for this transceiver architecture. In all cases, the optimal weights of a three pre-tap Tx-FIR are zero. Lab measurements verify this result.

On the other hand, the amount of precursor ISI can be effectively reduced (as seen by the receiver) by adjusting the data sampling location [24] and using DFE. With the receiver architecture shown in Figure 17.14, the CDR locking phase can be optimized by adjusting the edge sampler threshold, using the minBER algorithm to maximize the received voltage margin, $d_0 - \sum_{i=-\infty}^{-1}|d_i|$. At each edge sampler threshold (and thus each CDR locking phase), the DFE is adapted with SS-LMS, and the ISI at the corresponding data sampling location is canceled. Unlike sampling a given eye at a better location, shifting the DFE along with the CDR produces a different equalized eye at each CDR locking position. The eye is maximized when the impact of precursor ISI has been minimized relative to the main cursor.

17.5 ADC-Based Receive Equalization

As the complexity of electrical and optical communication links increases, interest is growing in implementing transceivers based on analog-to-digital converters (ADCs) and digital signal processing (DSP) [3] [4] [26] [30]. The aggressive scaling of CMOS makes building the fast digital logic that can perform sophisticated signal-processing algorithms in the digital domain increasingly feasible. History has also shown how ADC-based transceivers have become dominant solutions for telephone line modems and magnetic disk read channels [31].

Designing an ADC for transceivers operating above 10Gb/s that has high enough resolution so its quantization errors are negligible is very challenging. For example, even at a moderate resolution of only 6 bits, a high-speed ADC may dissipate more than 1W [32] [33]. Murmann [34] also noted that beyond 12-bit resolutions, thermal noise limits the signal-to-noise ratio (SNR), and that further technology scaling will not help to reduce power. High ADC power consumption has discouraged ADC+DSP adoption in backplane transceivers, especially since link power efficiency (measured in mW/Gb/s) has emerged as a key metric in power-constrained networking systems. For this reason, ADCs employed in high-speed links are typically limited to 4–5 bits of resolution [3] [26]. These relatively coarse-resolution ADCs limit the effectiveness of the digital linear equalization filters, due to the accumulation of quantization error [35].

Effectively, the DSP in these systems is used to implement DFE in the digital domain (as shown in Figure 17.17). The balance of this section explores the equivalence between ADC-based DFE and partial-response DFE receivers and introduces reduced-sampler PrDFE (RS-PrDFE) [35].

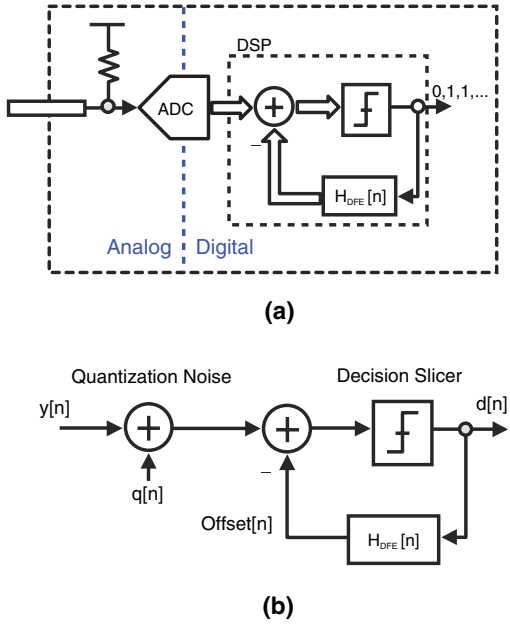


Figure 17.17 An ADC-Based DFE Receiver: (a) Architecture, and (b) Signal Flow Diagram (Where the ADC Is Modeled as a Source of Quantization Noise)

Figure 17.17 illustrates the signal flow in an ADC-based receiver performing DFE. After the ADC converts the received signal into a digital form, the DSP performs the DFE operation by computing and subtracting the appropriate amount of offset from the digitized input based on prior bit decisions. The DSP also contains the decision slicer, which compares the resulting value with a threshold, and determines the current bit.

The ADC-based DFE receiver in Figure 17.17, and the loop-unrolling PrDFE receiver in Figure 17.5 are actually equivalent and can be optimized using the same principles. Recall that the core DFE operation is to subtract an offset from the received signal before the current bit decision. The offset corresponds to the inter-symbol interference (ISI) caused by the previous bits. The current-bit decision is made based on whether the received signal is higher or lower than the offset value.

The ADC-based DFE receiver and the PrDFE receiver can be considered as two types of a general DFE architecture. Both have multiple samplers on the front-end, and combinational deterministic decision logic in the back-end—see Figure 17.18(a). The decision logic is purely combinational has no internal state or feedback. It only calculates the binary data, based on the sampler outputs $\{D_1[k], \ldots, D_N[k]\}$ and the given bit history $\{X[k-M], \ldots, X[k-1]\}$. In a PrDFE receiver, the deterministic decision logic is an N:1 multiplexer, as shown in Figure 17.18(b). One of the sampler outputs ($D_j[k]$) is selected, based on the given bit history.

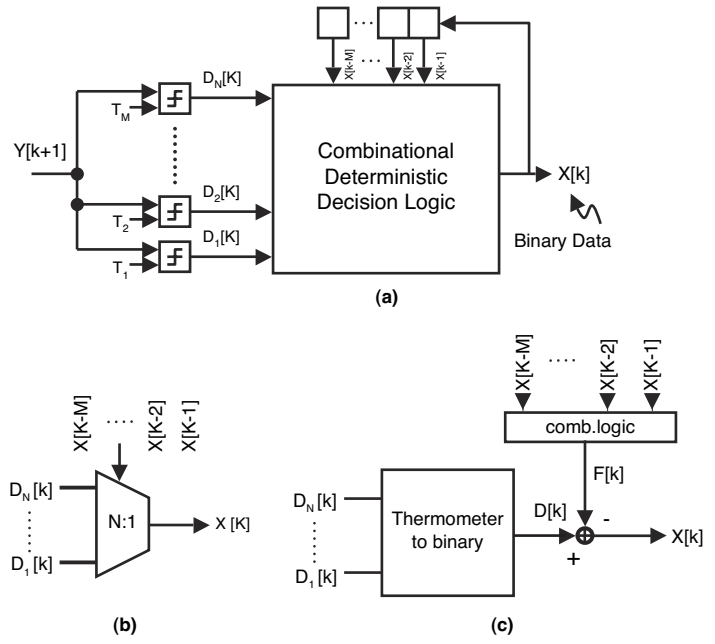


Figure 17.18 (a) A General Architecture with Multiple Samplers in the Front-End and Combinational Deterministic Decision Logic in the Back-End, (b) For PrDFE Receiver, the Deterministic Decision Logic Is a N:1 Multiplexer, (c) Deterministic Decision Logic in ADC-based DFE Receiver

For the ADC-based DFE shown in Figure 17.18(c) the sampler outputs are mapped from a thermometer code to a binary code for ease of computation. The ISI offset ($F[k]$) is computed in the digital domain, based on the given bit history $\{X[k-M], \ldots, X[k-1]\}$. Then, the decision is based on whether the quantized ADC output ($D[k]$) is greater than the ISI offset ($F[k]$). Because the thermometer-to-binary mapping is 1:1, this typical process can be viewed as the computed ISI offset ($F[k]$) being mapped to the thermometer code, and directly compared with the sampler outputs $\{D_1[k], \ldots, D_N[k]\}$. Because the sampler outputs $\{D_1[k], \ldots, D_N[k]\}$ are strictly monotonic, the result of the comparison is determined by a single sampler output value ($D_j[k]$). Thus, it is literally equivalent to selecting a particular comparator output from the thermometer-coded result, just as it is with PrDFE receivers.

Recognizing the similarity between the two receivers suggests an equivalent ADC-based DFE architecture, referred to as *reduced-slicer partial-response* DFE (RS-PrDFE), and shown in Figure 17.19 [35]. This architecture is similar to the PrDFE in Figure 17.5, in that it directly selects a loop-unrolled slicer decision as the current bit value. The key difference is that RS-PrDFE maps the bit history to the slicer selection through a look-up table. Because each entry in the look-up table may not be unique, mapping multiple bit histories to a single slicer is possible,

just as an ADC-based DFE might map multiple bit patterns to the same digital feedback value. This architecture is thus equivalent to an ADC-based DFE with equivalent ADC thresholds, yet it avoids the need for thermometer-to-binary conversion (and the associated metastability hardening and bubble correction considerations), and replaces the binary comparison block with a look-up table and a multiplexer.

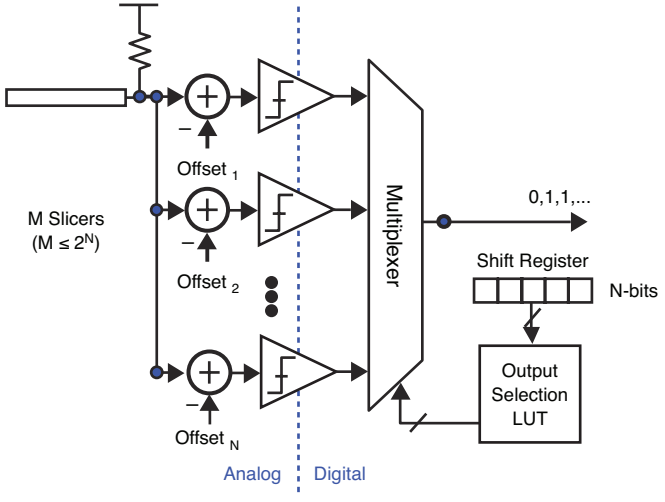


Figure 17.19 Proposed Reduced-Slicer PRDFE Receiver (RS-PRDFE)

Compared with PrDFE, which requires 2^N number of slicers to cover N postcursors, the RS-PrDFE architecture can achieve similar performance with fewer slicers [35]. The key difference between an ADC-based DFE and the RS-PrDFE receivers is the slicer threshold placement. The optimal placements for minimum BER and minimum signal quantization error are significantly different. As Figure 17.17 shows, the quantization errors introduced by the ADC are typically counted towards the unwanted noise. Consequently, the ADC-based DFE architecture strives to have as high a resolution ADC as possible. On the other hand, RS-PrDFE strives to minimize minimum threshold error between the offsets and the corresponding ISI levels. Kim et al. show a recursive algorithm with which to locate the optimum slicer thresholds for Rs-PrDFE [35].

17.6 Future of High-Speed Wireline Equalization

As computer and network bandwidth requirements continue to grow exponentially, off-chip data rates have grown drastically over the past decade [28] [29]. Designers are beginning to incorporate wireline data rates as high as 28Gbit/s by 2011. At such high data rates, equalization will continue

to be one of the critical underlying technologies. For lower speed links, with relatively low dispersion and reflections, designers will continue to favor a binary front-end with an analog equalizer, for both its simplicity and its power efficiency. Although ADC front-ends with digital equalization have gained a lot of interest, they will not take over the whole solution space, as they have done for other applications, due to the higher speeds and stringent power efficiency. Further, ADC front-ends for high-speed links have less stringent requirements than the normal ADCs used to recover analog signals. For example, links are relatively tolerant of DNL errors, sampling jitter, and so on. To save power, and to achieve a minimum BER, it is likely that other new techniques, such as clipped ADC [3], and RS-PrDFE [35], will be used in conjunction with analog equalization.

3D integration technology, such as "through silicon vias" TSV [36], can potentially provide many shorter and cleaner channels between chips, relieving the need for equalization in some applications. For example, a large L4 cache might be placed close to the CPU, via TSV, in order to provide low latency and high aggregate bandwidth access. However, for other applications, where high-speed communication must be achieved over some physical distance, wireline equalization will remain an important part of the solution. In the future, signaling over TSVs at higher data rates might be desirable, in order to provide even greater aggregate bandwidth. TSVs are inherently capacitive loads; for example, ~200fF for a TSV connecting two dies (depending on the geometry and dielectric). Signaling over multiple TSV connections at high data rates will also require some degree of equalization.

In addition to TSV technology, silicon photonics [37] is another technology that could take over a part of the solution space. Silicon photonics technologies integrate optical components on CMOS dies, using a standard or slightly modified CMOS fabrication process to reduce the cost of optical communication. 40Gb/s active optical cables have been shown based on silicon photonics technology. Even though the power efficiency (20mW/Gb/s–25mW/Gb/s) and cost ($2/Gb/s) still do not match electrical links, they are much closer than before and have potential for further scaling. Equalization remains useful, even with silicon photonics; this is because the optical link also suffers from bandwidth limitation (for example, limited responsive time of the photo detector).

17.7 Summary

At high data rates, signal integrity issues such as dispersion and reflections severely limit the available channel bandwidth. For the past decade, signal conditioning techniques have become widely adopted in high-speed interfaces to mitigate the impact of those signal integrity issues and extend the channel bandwidth. This chapter presents the basic concepts of equalization and summarizes the current state of the art equalization techniques. Moreover, to compensate for process and environmental variations, the equalization settings have to be adapted. This chapter also covers a few adaptation algorithms that require minimal hardware support. Finally, the chapter briefly discusses ADC-based serial links and the future of wireline equalization.

References

1. http://www.itrs.net/links/2007itrs/2007_chapters/2007_Assembly.pdf
2. H. W. Bode, "Design of broad band receivers," U.S. Patent 2,242,878, May 20, 1941.
3. C.-K. K. Yang and E.-H. Chen, "ADC-based serial I/O receivers," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2009, pp. 323–330.
4. H. Chung and G.-Y. Wei, "Design-space exploration of backplane receivers with high-speed ADCs and digital equalization," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2009, pp. 555–558.
5. W.J. Dally and J. Poulton, "Transmitter equalization for 4-Gbps signaling," *IEEE Micro*, vol. 17, no.1, pp. 48–56, Jan–Feb. 1997.
6. V. Stojanovic, "Channel-limited high-speed links: modeling, analysis and design," *ProQuest / UMI*, 2006.
7. J. Zerbe, C. Werner, V. Stojanovic, F. Chen, J. Wei, G. Tsang, D. Kim, W. Stonecypher, A. Ho, T. Thrush, R. Kollipara, G-J. Yeh, M. Horowitz, and K. Donnelly, "Equalization and clock recovery for a 2.5-10Gb/s 2-PAM/4-PAM backplane transceiver cell," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2121–2130, Dec. 2003.
8. A. Ho, V. Stojanovic, F. Chen, C. Werner, G. Tsang, E. Alon, R. Kollipara, J. Zerbe, and M.A. Horowitz, "Common-mode backchannel signaling system for differential high-speed links," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 2004, pp. 352–55.
9. V. Stojanovic, A. Ho, B. W. Garlepp, F. Chen, J. Wei, G. Tsang, E. Alon, R. T. Kollipara, C. W. Werner, J. L. Zerbe, and M. A. Horowitz, "Autonomous dual-mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 1012–1026, Apr. 2005.
10. B. Razavi, *Design of Integrated Circuits for Optical Communications*, p. 133, McGraw-Hill, 2003.
11. S. Gondi and B. Razavi, "Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 9, pp. 1999–2011, Sep. 2007.
12. B. S. Leibowitz, J. K. H. Lee, F. Chen, A. Ho, M. Jeeradit, A. Bansal, T. Greer, S. Li, R. Farjad-Rad, W. Stonecypher, Y. Frans, B. Daly, F. Heaton, B. W. Gariepp, C. W. Werner, N. Nhat, V. Stojanovic, and J. L. Zerbe, "A 7.5 Gb/s 10-tap DFE receiver with first tap partial response, spectrally gated adaptation, and 2nd-order data-filtered CDR," in *International Solid-State Circuits Conference Digest of Technical Papers*, 2007, pp. 228–599.

13. R. Payne, B. Bhakta, S. Ramaswamy, S. Wu, J. Powers, P. Landman, U. Erdogan, Ah-Lyan Yee, R. Gu, Lin Wu, Y. Xie, B. Parthasarathy, K. Brouse, W. Mohammed, K. Heragu, V. Gupta, L. Dyson, and W. Lee, "A 6.25Gb/s binary adaptive DFE with first post-cursor tap cancellation for serial backplane communications," in *International Solid-State Circuits Conference Digest of Technical Papers*, 2005, pp. 68–585.
14. Y.-S. Sohn, S.-J. Bae, H.-J. Park, C.-H. Kim, and S.-I. Cho, "A 2.2Gbps CMOS look-ahead DFE receiver for multidrop channel with pin-to-pin time skew compensation," in *Proceedings of IEEE Custom Integrated Circuits Conference*, 2003, pp. 473–476.
15. J. Zerbe, Q. Lin, V. Stojanovic, A. Ho, R. Kollipara, F. Lambrecht, and C. Werner, "Comparison of adaptive and non-adaptive equalization techniques in high performance backplanes over environmental variations," presented at the IEC DesignCon, Santa Clara, CA, 2005.
16. B. Widrow, S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, 1985.
17. S. Dasgupta and C. R. Johnson, Jr., "Some comments on the behavior of sign-sign adaptive identifiers," *Systems & Control Letters* 7, Elsevier Science Publishers B. V. (North-Holland), pp. 75–82, Apr. 1986.
18. J. Ren, H. Lee, B. Leibowitz, R. Ratnayake, Q. Lin, K. Kelly, D. Oh, V. Stojanovic, J. Zerbe, and N. Nguyen, "Performance comparison of edge-based equalization with data-based equalization," presented at the IEC DesignCon, Santa Clara, CA, 2007.
19. B. Brunn and S. Anderson, "Edge-equalization extends performance in multi-gigabit serial links," presented at the IEC DesignCon, Santa Clara, CA, 2005.
20. K-L. J. Wong, C-K. K. Yang, "A serial-link transceiver with transition equalization," in *International Solid-State Circuits Conference Digest of Technical Papers*, 2006, pp. 223–232.
21. E.-H. Chen, J. Ren, B. Leibowitz, H. Lee, Q. Lin, K. Oh, F. Lambrecht, V. Stojanovic, J. Zerbe, and C.-K.K. Yang, "Near-optimal equalizer and timing adaptation for I/O links using a BER-based metric," *IEEE Journal of Solid State Circuits*, vol. 43, no. 9, pp. 2144–2156, Sep. 2008.
22. E.-H. Chen, J. Ren, J. Zerbe, B. Leibowitz, H. Lee, V. Stojanovic, and C.-K. K. Yang, "BER-based adaptation of I/O link equalizers," in *Symposium on VLSI Circuits Digest of Technical Papers,* Jun. 14–16, 2007, pp. 36–37.
23. J. Ren and M. Greenstreet, "A unified optimization framework for equalization filter synthesis," in *Proceedings of Design Automation Conference*, Jun. 13–17, 2005, pp. 638–642.
24. J. Ren, H. Lee, Q. Lin, B. Leibowitz, E.-H. Chen, D. Oh, F. Lambrecht, V. Stojanovic, C.-K. K. Yang, and J. Zerbe "Precursor ISI reduction in high-speed I/O," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 14–16, 2007, pp. 134–135.

25. R. Randall and W. Zucker, "Adaptive equalizer using precursor error signal for convergence control," US Patent 4789994, 1987.

26. M. Harwood, N. Warke, R. Simpson, T. Leslie, A. Amerasekera, S. Batty, D. Colman, E. Carr, V. Gopinathan, S. Hubbins, P. Hunt, A. Joy, P. Khandelwal, B. Killips, T. Krause, S. Lytollis, A. Pickering, M. Saxton, D. Sebastio, G. Swanson, A. Szczepanek, T. Ward, J. Williams, R. Williams, and T. Willwerth, "A 12.5Gb/s SerDes in 65nm CMOS using a baud-rate ADC with digital receiver equalization and clock recovery," in *International Solid-State Circuits Conference Digest of Technical Papers*, 2007, pp. 436–591.

27. J. G. Proakis, *Digital Communications*, Fourth ed., McGraw-Hill, 2000.

28. G. Balamurugan, J. Kennedy, G. Banerjee, J. E. Jaussi, M. Mansuri, F. O'Mahony, B. Casper, and R. Mooney, "A scalable 5-15Gbps, 14-75mW low power I/O transceiver in 65nm CMOS," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun 14–16, 2007, pp. 270–271.

29. K. Chang, H. Lee, J. Chun, T. Wu, T. J. Chin, K. Kaviani, J. Shen, X. Shi, W. Beyene, Y. Frans, B. Leibowitz, N. Nguyen, F. Quan, J. Zerbe, R. Perego, and F. Assaderaghi, "A 16Gb/s/link, 64GB/s bidirectional asymmetric memory interface cell," in *Symposium on VLSI Circuits Digest of Technical Papers*, Jun 18–20, 2008, pp. 126–127.

30. O. E. Agazzi, D. Crivelli, M. Hueda, H. Carrer, G. Luna, A. Nazemi, C. Grace, B. Kobeissy, C. Abidin, M. Kazemi, M. Kargar, C. Marquez, S. Ramprasad, F. Bollo, V. Posse, S. Wang, G. Asmanis, G. Eaton, N. Swenson, T. Lindsay, and P. Voois, "A 90 nm CMOS DSP MLSD transceiver with integrated AFE for electronic dispersion compensation of multimode optical fibers at 10 Gb/s," *IEEE Journal of Solid-State Circuits*, pp. 2939–2957, Dec. 2008.

31. K. Pahlavan and J. L. Holsinger, "Voice-band data communication modems—a historical review: 1919–1988," *IEEE Communications Magazine*, pp. 16–27, Jan. 1988.

32. P. Schvan, J. Bach, C. Fait, P. Flemke, R. Gibbins, Y. Greshishchev, N. Ben-Hamida, D. Pollex, J. Sitch, Shing-Chi Wang, and J. Wolczanski, "A 24GS/s 6b ADC in 90nm CMOS," in *International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2008, pp. 544–545.

33. Y. M. Greshishchev, J. Aguirre, M. Besson, R. Gibbins, C. Falt, P. Flemke, N. Ben-Hamida, D. Pollex, P. Schvan, and S.-C. Wang, "A 40GS/s 6b ADC in 65nm CMOS," in *International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2010, pp. 390–391.

34. B. Murmann, "A/D converter trends: power dissipation, scaling and digitally assisted architectures," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2008, pp. 105–112.

35. J. Kim, J. Ren, B. Leibowitz, P. Satarzadeh, A. Abbasfar, and J. Zerbe, "Equalizer design and performance trade-offs in ADC-based serial links," in *Proceedings of IEEE Custom Integrated Circuits Conference*, Sep. 2010, pp. 1–8.
36. M. Motoyoshi, "Through-Silicon via (TSV)," in *Proceedings of the IEEE*, vol. 97, no.1, pp.43–48, Jan. 2009.
37. C. Gunn, "CMOS photonics for high-speed interconnects," *IEEE Micro*, Vol. 26, no. 2, pp. 58–66, Mar. 2006.

This page intentionally left blank

CHAPTER 18

Applications

Chuck Yuan, Dan Oh, and Ravi Kollipara

Throughout this book, the various aspects of modeling and analyzing I/O interface channels are examined. However, the complete analysis of a particular signaling scheme has not been discussed. In this chapter, several signaling schemes are analyzed to illustrate the concepts described in previous chapters with specific examples. In the final section, we speculate about what signaling will look like in the future, and the challenges that it will present.

Rambus' XDR memory channel is the first example. It demonstrates several key innovations that mitigate signal integrity issues in high-speed memory operations. These innovations include the FlexPhase timing adjustment (which solves pin-to-pin timing variations with trace length variation), and dynamic point-to-point (DPP) memory architecture (which supports multiple memory modules without limiting signal quality due to multi-drop bus issues). This memory interface has a wide range of applications including main memory, high-end graphics, and low-end consumer applications. To handle greater capacity, the XDR memory system uses single-ended signaling for the address and command lines, and current-mode differential signaling for the data lines.

The second example is Rambus' Mobile XDR memory system for a low-power memory interface. There are several key attributes of the design. First, it employs an optimized clocking architecture for applications with short channels and fast power-mode transitions. Second, it uses fully differential signaling for both data and address/command lines. Finally, it supports multiple data speeds to optimize power consumption. The first generation of the Mobile XDR system supports 2.4Gb/s~4.3Gb/s, including sub-rates and a separate low-speed mode (<100Mb/s) for maximum power savings. Because the Mobile XDR interface uses purely differential signaling, it provides sufficient headroom for future computing power in mobile applications.

In the last example, some of the key Rambus technologies are applied to single-ended main memory applications. The proposed solutions can significantly extend the bandwidth of the

current DDR-based memory systems. Particularly, FlexPhase and DPP technologies can be implemented with minimum design changes to existing DRAM devices. However, future memory systems must achieve their higher bandwidths by staying within the current-generation memory system's power envelope. The *Near Ground Signaling* (NGS) described in this last example greatly helps in this regard, by significantly lowering the IO power consumption, while advancing the data rates. The next generation of DDR-based memory systems will support data rates in the range of 1.6Gb/s–3.2Gb/s. NGS can achieve these data rates without any equalization, either on the memory controller side or on the DRAM side.

Finally, we close this chapter by predicting the future signaling roadmap. The challenge of future interface signaling is to achieve a much higher data bandwidth within a power and cost envelope that is the same as, or similar to, today's system.

18.1 XDR: High-Performance Differential Memory System

This section introduces the XDR memory channel [1] [2]. All the material presented here can be found on the Rambus website at www.rambus.com. In early 2000, Rambus began to design its successor to the RDRAM memory channel, for high-volume and low-cost applications, such as PCs and game consoles. Memory scalability, in terms of both bandwidth and capacity, was the major design goal. To have maximum bandwidth scalability, combined with the best signal integrity, differential signaling and point-to-point topology for DQ signals are chosen to support data rates from 3.2Gb/s–6.4Gb/s. This was 10x to 20x the DDR data rate in 2000. The command and address (RQ) signals remained single ended, and used the same fly-by (bussed) topology as the RDRAM channel, which supported data rates from 800Mz to 1.6GHz. The inclusion of the FlexPhase timing circuit removed the static timing skew caused by both device- and PCB-trace mismatches. A dynamic point-to-point (DPP) technology is developed to allow module upgrades, while maintaining the point-to-point topology for the DQ signals (to achieve the best signal integrity with connectors). XDR memory is currently in high-volume production, shipping more than 100 million parts for Sony's game console PS3, DLP projectors, and DTVs.

18.1.1 XDR Memory Architecture

Rambus XDR memory architecture, as illustrated in Figure 18.1, is a total memory system solution with performance that is an order of magnitude higher than today's standard memories, while utilizing the fewest ICs. Ideal for high-end computing and consumer electronics applications, a single, 4-byte-wide, 6.4Gb/s XDR DRAM component provides 25.6GB/s of peak memory bandwidth.

The key components that enabled the breakthrough performance of the XDR memory architecture are:

- XDR DRAM is a high-speed memory IC that accelerates standard CMOS DRAM cores with a high-speed interface. The interface is capable of 7.2Gb/s data rates and provides up to 28.8GB/s of bandwidth with a single device.

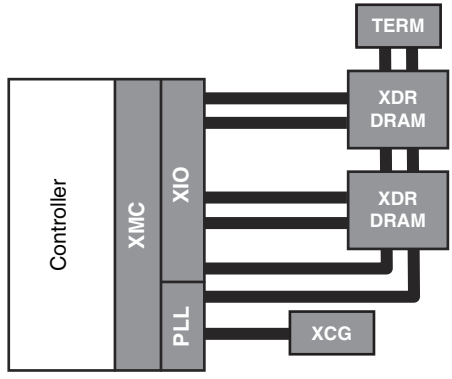


Figure 18.1 XDR Memory Architecture

- The XIO controller I/O cell provides the same high-speed signaling capability found on the DRAM, but adds additional enhancements (such as FlexPhase technology) that eliminate the need for trace-length matching.
- The XMC memory controller is a fully synthesizable logical memory controller optimized to take advantage of innovations like Dynamic Point-to-Point signaling, which provides for capacity expansion while delivering the signal integrity benefits of point-to-point signaling.
- The XCG clock generator provides the system clocks with four programmable outputs, and guarantees that the XIO and XDR DRAM clocking requirements are met.

Figure 18.2 shows the signal summary for XDR memory signaling system. The XDR memory interface architecture consists of four building-block technologies: Differential Rambus Signaling Level (DRSL), Octal Data Rate (ODR), FlexPhase de-skewing circuitry, and Dynamic Point-to-Point (DPP) technology:

- The *Differential Rambus Signaling Level* (DRSL) is a low-voltage, low-power, differential signaling standard that enables the scalable multi-GHz, bi-directional, and point-to-point data buses, which connect the XIO cell to XDR DRAM devices. XDR memory solutions also use the Rambus Signaling Level (RSL) standard, originally developed for the RDRAM memory interface, to enable up to 36 devices to connect to the source-synchronous, bused address and command signals.
- *Octal Data Rate* (ODR) is a technology that transfers eight bits of data on each clock cycle (four times as many as today's state-of-the-art memory technologies that use Double Data Rate [DDR]). XDR data rates are scalable to 7.2Gb/s.
- *FlexPhase* de-skewing circuits eliminate any systematic timing offsets between the bits of an XDR data bus. With a resolution of 2.5ps (at 3.2Gb/s), and a maximum range of more than 10ns, FlexPhase technology eliminates the need to match trace lengths on the

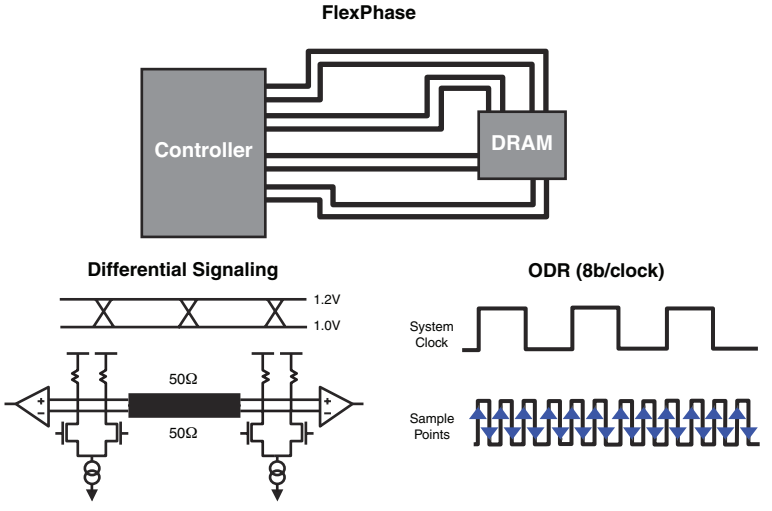


Figure 18.2 XDR Signaling Summary

board and package. FlexPhase also dynamically calibrates out on-chip clock skew, driver/receiver mismatch, and clock standing-wave effects.

- *Dynamic Point-to-Point* (DPP) signaling technology maintains the signal integrity benefits of point-to-point signaling on the data bus, while providing the flexibility for capacity expansions with module upgrades. Memory modules can be dynamically reconfigured to support different data bus widths, allowing a memory controller with a fixed data bus width to connect to a variable number of modules.

18.1.2 XDR Clocking Architecture

Figure 18.3 shows the XDR memory clocking architecture. The system clock provides the input to the XCG Clock Generator, which outputs the clock signal (CTM) necessary to support an XDR memory system. The CTM signal is routed to the XIO PHY area, looped around inside the XIO package, and routed to the DRAM as CFM. The stub, created by the short clock-package trace length, is minimized to avoid degradation of the clock signals. CFM provides the input to the XDR DRAM's PLL to generate the XDR on-chip clock signals. CFM is routed as a bus (as with the RQ signals) to minimize the number of XDR package pins. At a DQ data rate of 3.2Gb/s, the CFM/CTM frequency is 400MHz.

The XCG clock generator is an off-the-shelf solution provided by several leading integrated circuit companies, and supports a broad range of high-performance clocking applications. This is because the two PLLs in the clocking architecture (XIO and XDR devices) form a band-pass filter that filters out most of the system clock jitter. Guaranteed to provide clocks that meet the specifications for both the XIO Cell and the XDR DRAM device, the XCG device provides

four programmable differential outputs, using a reference input clock of 100MHz or 133MHz. It also supports spread spectrum modulation, reducing the EMI generated from the clock distribution network.

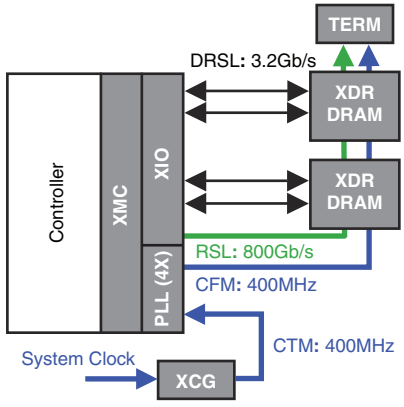


Figure 18.3 XDR System Clocking Architecture

18.1.3 XDR Memory FlexPhase Timing Adjustments

There are two types of timing error in a clocking system, such as the one shown in Figure 18.3. One is the dynamic error (or jitter) caused by noise sources, such as supply noise, thermal noise, crosstalk, and inter-symbol interference. The other type of error is the static error (or skew) caused by data-path and clock-path mismatches in both the device, and PCB/package, physical interconnect routing. Timing calibration, using FlexPhase technology, can remove the static error.

Advanced memory solutions, including XDR DRAM, can use FlexPhase technology to achieve increased per-pin signaling rates. In an XDR system, as illustrated in Figure 18.4, FlexPhase technology anticipates the phase difference between signals on different traces, and manages the transmission of the data bits. The data arrives at the memory device with a known timing relationship, with respect to the command and address signals sent to the memory device. FlexPhase can also enhance conventional DRAM architectures.

FlexPhase improves system data rates by:

- Optimizing I/O signal timing to improve timing margins
- Complementing fly-by command/address system architectures
- Eliminating the requirements for trace-length matching

FlexPhase technology eliminates the need to match trace lengths, both on the circuit boards supporting the memory system and within the packages for the memory devices. This system simplification lowers board and packaging costs. FlexPhase also improves overall system timing

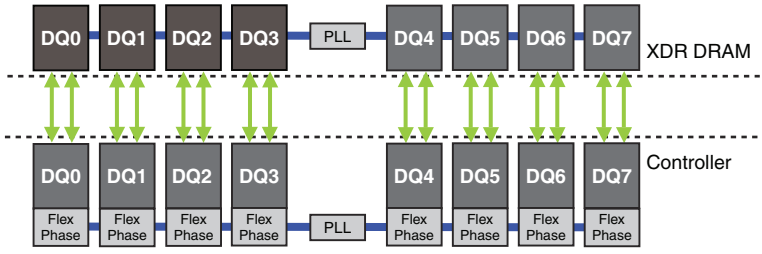


Figure 18.4 FlexPhase Implemented on a Single Byte of an XDR Interface

by eliminating many timing offsets through dynamic compensation for process variations (for example, on-chip clock skew, driver/receiver mismatch, and clock standing-wave effects).

FlexPhase circuit technology brings flexibility, simplicity, and savings to memory system design. FlexPhase circuits can be used in DRAM systems to optimize data and strobe placement. FlexPhase circuits can also fine-tune the timing relationships between the data, command, address and clock signals. In conventional DRAM architectures, FlexPhase circuits can deskew incoming signals at the controller to compensate for uncertainty regarding the arrival times of signals. Additionally, FlexPhase circuits can be used to intentionally inject a timing offset to *preskew* the data, so that it arrives at the DRAM coincident with the command/address or clock signal. FlexPhase minimizes the systematic timing errors in typical memory systems by the adjusting transmit and receive phase offsets at each pin or pin-group.

When using a fly-by architecture (see Figure 18.5), the amount of time required for the data, strobe, command, address, and clock signals to propagate between the memory controller and DRAMs is primarily affected by the lengths of the traces between the controller and the DRAM devices over which the signals propagate. In a fly-by system, the command, address, and clock signals arrive at each DRAM at different times, which, in turn, results in the data signals being transmitted from each DRAM device at different times. FlexPhase can be used at the controller to deskew those data signals, eliminating the offset due to the fly-by architecture, as well as any inherent timing offsets of the system. Similarly, because the command, address, and clock signals arrive at each DRAM at different times, the controller must preskew the data for write operations to the memory devices to account for the difference in when the memory devices expect the write data. FlexPhase can perform this "preskewing," and still eliminate inherent timing offsets in the system.

FlexPhase timing adjustment is a departure from traditional serial link technologies, which use an embedded clock to deskew the timing. Such deskewing techniques (which typically rely on 8b/10b encoding to ensure adequate transition density for clock recovery) require more chip area, have added power consumption, increased latency, and suffer from a 25 percent bandwidth penalty associated with the 8b/10b encoding.

FlexPhase technology includes in-system timing characterization and self-test functionality that enables aggressive timing resolutions in high-performance memory systems. FlexPhase,

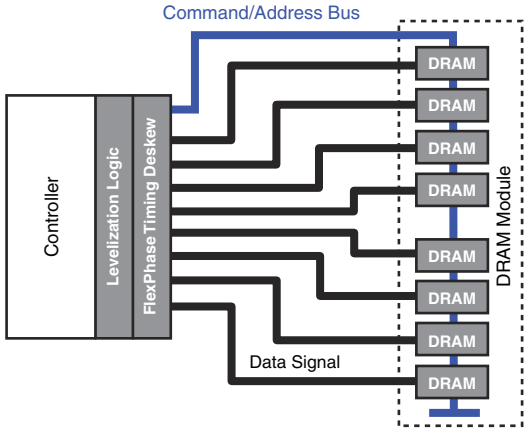


Figure 18.5 Example of Memory Module System Employing "Fly-By" Topology

incorporated within the XDR memory system, results in timing resolutions of 2.5ps, at data rates of 3.2GHz.

18.1.3.1 FlexPhase System Operation

During read access operations, in the example XDR system, a memory controller incorporating FlexPhase technology determines and stores the "receive" phase difference between the transmitted control signals and the data received from each memory device. Subsequently, the phase difference, corresponding to each memory device, is used to deskew the data signals that arrive at the memory controller at different times, thereby allowing for the proper reconstitution of the data accessed from each of the memory devices.

The process is similar during write operations: The "transmit" phase difference is determined for each memory device, and then stored within the memory controller. The transmit phase differences are then used to modify (pre-skew) the timing delay between the transmitted command/address signals and the data sent to each memory device.

18.1.3.2 Device Benefits

At GHz data rates, FlexPhase technology helps to compensate for the manufacturing variations that degrade timing windows and the operational performance of the memory. The FlexPhase approach allows memory interfaces to operate at GHz rates without the power, area, and latency penalties incurred in systems using Clock and Data Recovery (CDR) techniques. FlexPhase also provides for improved testability by using digital phase offsets for margin testing of the high-speed chip interfaces. (Details of margin testing are covered in Chapter 16, "On-Chip Link Measurement Techniques.")

18.1.3.3 System Benefits

FlexPhase technology relaxes PCB trace-length matching requirements by anticipating and calibrating the signaling phase offsets caused by variations in trace lengths and impedances. FlexPhase timing adjustments allow for much simpler, more compact, and more cost-efficient memory layouts. FlexPhase timing adjustments also support the in-system test and characterization of key data signals, enabling performance testing of the high-speed links.

18.1.4 XDR Memory Module: XDIMM

The Rambus XDIMM module (see Figure 18.5) is a high-capacity memory module designed with high-performance XDR DRAM devices. It provides the upgrade flexibility, capacity, and performance essential for servers, consumer electronics, and main memory applications. Operating at data rates of up to 6.4Gb/s, a single XDIMM module provides many times the bandwidth of today's module-based systems, while enabling multi-gigabyte capacities. Using Dynamic Point-to-Point (DPP) technology, XDIMM modules can be installed in single or dual configurations that maintain full system bandwidth and preserve the signal integrity of a true point-to-point topology. The XDIMM module is part of the XDR memory system solution, allowing system and chip designers to integrate the world's fastest memory technology, while eliminating interoperability and signal integrity problems.

18.1.5 XDR Memory Module Upgrade Using DPP Technology

As memory bus speeds continue to increase, maintaining good signal integrity becomes increasingly difficult. Conventional memory buses, in personal computers and workstations, support multi-drop data topologies, which allow more than one device per data signal. These topologies support upgradeability by allowing multiple modules to connect to the bus. However, multi-drop topologies can degrade signal integrity and reduce the speed at which the memory bus can run. In multi-drop topologies, one factor that determines the speed of the memory bus is the worst-case loading characteristics, which occur when memory modules populate all the connectors. Point-to-point topologies (one device at each end of the signal line) have better signal integrity properties, and permit higher bus speeds, but cannot be upgraded, because they do not allow multiple modules. The ability to increase memory system capacity by adding memory modules is such an important feature in computer systems today that traditional main memory systems support multi-drop topologies instead of point-to-point topologies. In the early 2000s, Rambus began investigating ways of combining the benefits of point-to-point signaling with the ability to upgrade memory capacity. The result was Dynamic Point-to-Point signaling technology.

Dynamic Point-to-Point technology combines the benefits of both point-to-point and multi-drop topologies, which allows the creation of memory systems that have point-to-point signaling and the flexibility to add memory capacity through module upgrades. A key benefit of DPP technology is that, by providing capacity expansion, DPP technology allows point-to-point upgrades at full memory system bandwidth. DPP technology can be applied to many different types of memory technologies, including XDR DRAM, SDRAM, DDR SDRAM, and the following

generations of DDR systems (see Section 18.3.4.) Figures 18.6 and 18.7 illustrate how DPP technology is used in an XDR DRAM memory system. As shown in Figure 18.6, the base system configuration has a single memory module, which supplies all the memory bandwidth across the full data-path width. A continuity module occupies the second memory slot, providing electrical continuity that maintains the point-to-point connection across half of the data path.

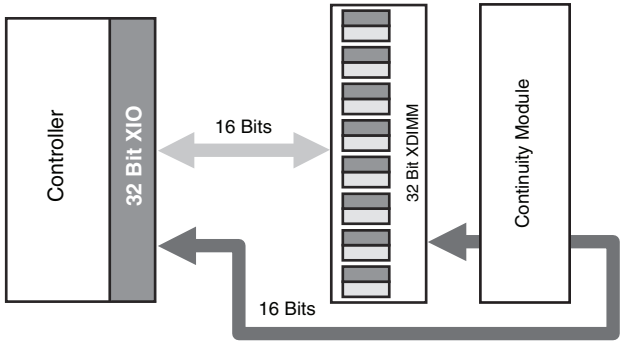


Figure 18.6 Base System Configuration: One 32-Bit Module Installed in 32-Bit XDR System

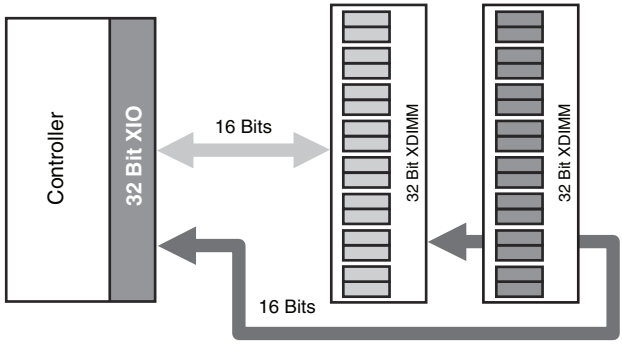


Figure 18.7 Upgrade System Configuration: Two 32-Bit Modules Installed in 32-Bit XDR System

When the continuity module is removed, and an expansion module is added (refer to Figure 18.7), the data path is reconfigured to supply memory bandwidth from both modules. In this example, each module supplies half of the memory system bandwidth, across a different half of the data path, in a point-to-point topology. Using DPP technology, the single 32-bit module is "dynamically rewired" to become a 16-bit module, when the second module is added. XDIMM modules accomplish this by changing the width of the memory devices on the XDIMM module: the XDR DRAMs switch from x4 DRAMs (in the base configuration) to x2 DRAMs (in the upgraded configuration). In the x4 mode, each XDR DRAM supplies four bits of data: two bits

directly to the ASIC and two bits through the continuity module to the ASIC. After inserting an upgrade module, the path through the continuity module is broken, and the devices switch to x2 mode. In x2 mode, each XDR DRAM supplies two bits of data directly to the ASIC.

Point-to-point signaling is maintained both before and after the capacity upgrade, which allows the memory system bandwidth to also be maintained. The dynamic rewiring in DPP technology allows the memory system to retain the signal integrity benefits of point-to-point signaling, while enabling memory-system capacity expansion at full memory system bandwidth. DPP technology, used in conjunction with FlexPhase technology, presents a compelling framework for memory system architecture.

18.1.6 XDR Memory Channel Topology in PlayStation®3

To illustrate the XDR channel characteristics, this section describes a channel similar to the one implemented in the Sony PS3 system. Here, the PCB trace length is less than 5 inches. The XIO interface uses a wirebond package to minimize the system cost. The channel transfer function (or insertion loss) and the single-bit response at 4.8Gb/s are shown in Figure 18.8 and Figure 18.9, respectively. The maximum differential crosstalk is less than 30dB. The attenuation at 2.4GHz is 6dB, and the postcursor is less than 15% of the main bit. Although equalization, such as transmit FIR, is beneficial, it is not required.

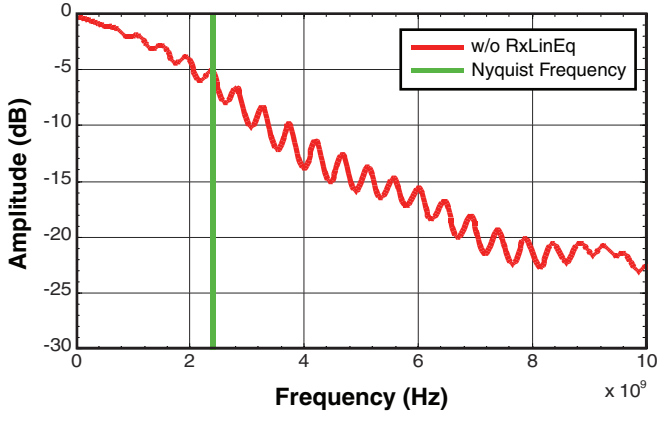


Figure 18.8 Transfer Function of 5-Inch PCB XDR Memory DQ Channel with XIO in Wirebond Package

Figure 18.10 shows the eye diagram, timing bathtub, and voltage bathtub at 4.8Gb/s. The XIO interface can also be implemented in an ultra-low cost package, such as low-profile quad flat packages (LQFP), where there is no ground reference and package inductance is very large [3]. Thanks to differential signaling, the differential impedance of a LQFP package is still close to 100ohm, because the two wires form a coplanar transmission line that eliminates the need for a ground plane. As shown in Figure 18.11, the LQPF XIO eye diagram (at 3.2Gb/s) is similar to the

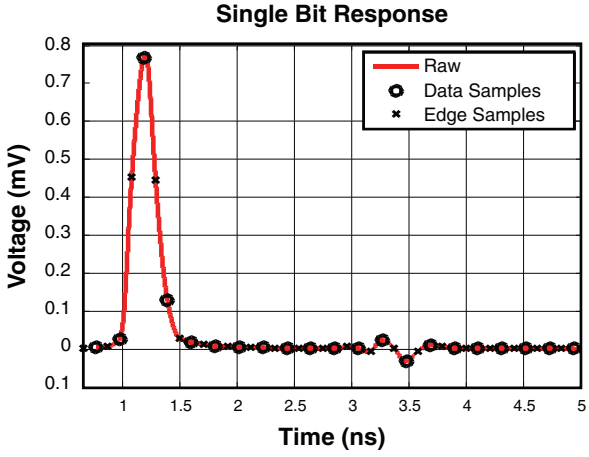


Figure 18.9 Single-Bit Response at 4.8Gb/s of 5-Inch PCB XDR Memory DQ Channel with XIO in Wirebond Package

wirebond XIO 4.8Gb/s eye diagram. While showing some signal degradation, the eye is still large enough for robust channel operation.

18.2 Mobile XDR: Low Power Differential Memory System

Traditional application categories, such as mobile, consumer electronics, and computing, are converging, with portability becoming a key factor in designing these products. The memory system used in these devices must optimize cost, power consumption, performance, and form factor, all at the same time. For instance, smart phones have become an emerging computing platform. The latest smart phones not only handle context-rich web pages, they also provide HD quality video processing power. Providing the required data bandwidth, using small form factor and low power consumption, is quite a challenging task.

Extending the data rates of the current LPDDR2 memory systems to next-generation mobile applications can be difficult when using existing packaging solutions, such as PoP [4]. It may require advanced packaging solutions, such as Through-Silicon Via (TSV), or expensive fine ball-pitch packaging technologies. Although TSV has gained much attention recently, it has not yet been adapted by high-volume memory systems. This is due to the significant cost overhead, in addition to technical design challenges, such as testing and thermal issues.

To address the increased bandwidth requirement without drastic changes in infrastructure Rambus introduced the Mobile XDR technology. A Mobile XDR system uses many of key innovations in XDR memory, such as FlexPhase and DPP technologies. However, the signaling architecture is designed from scratch to maximize power saving. Because mobile applications use fewer DRAMs than main memory computing applications, all signals (including the address and command lines) use differential signaling. This is in contrast to the XDR system, where only the data signals use differential signaling. Compared to single-ended signaling (which suffers from

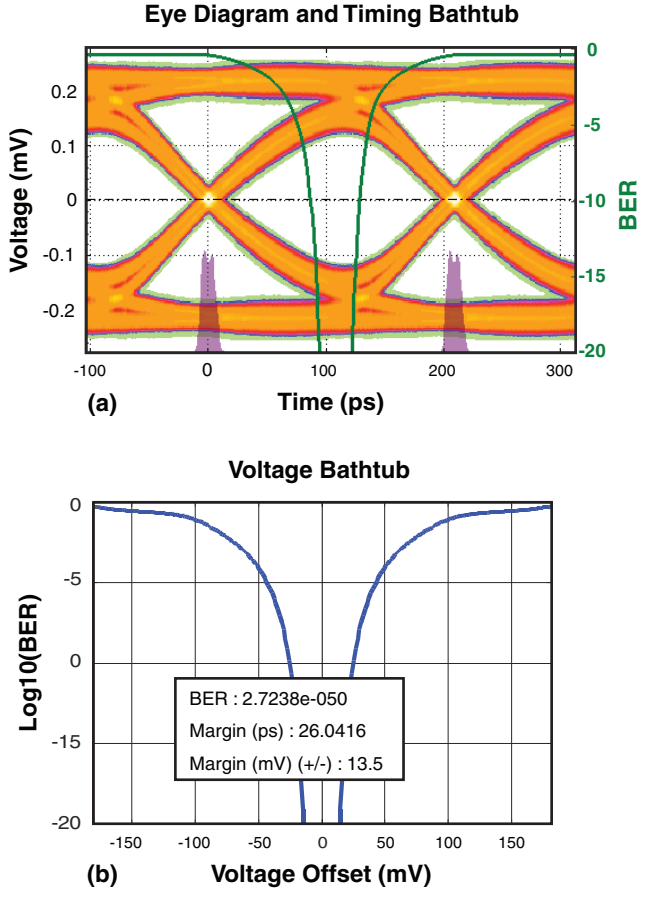


Figure 18.10 5-Inch PCB XDR Memory Channel at 4.8Gb/s with XIO in Wirebond Package: (a) Eye Diagram and Timing Bathtub, and (b) Voltage Bathtub

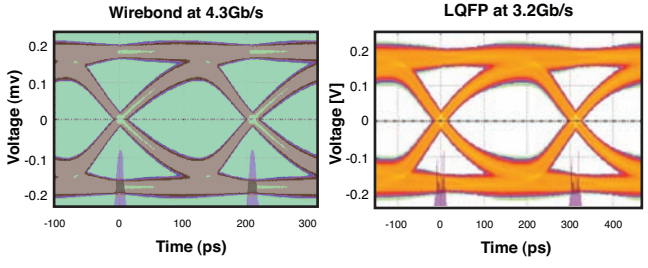


Figure 18.11 Eye Diagram Comparison of XIO in Wirebond at 4.3Gb/s and LQFP at 3.2Gb/s

crosstalk, SSO, and EMI issues [5]), the differential interface provides a clean channel response that enables low-swing signaling. Section 18.2.1 covers the key Mobile XDR–enabling technologies to lower power consumption. An analysis shows that the Mobile XDR system consumes only ~40% of the power consumed by an LPDDR2 system, at the same data bandwidth.

The small form factor requirement in 3D integration presents additional design challenges. Providing a clean power supply for the various interfaces, and the processor, is difficult due to tight integration and power requirements. Consequently, modeling the impact of the various noise sources in 3D packaging is important. Predicting link performance with noise, using traditional SPICE simulation, is limited due to its computational efficiency. In Section 18.2.2, the statistical approach described in Chapters 8, "Link BER Modeling and Simulation" and 9, "Fast Time-Domain Channel Simulation Techniques," is used to predict the link performance. The correlation and simulation results show that the Mobile XDR technology provides large bandwidth headroom, without using expensive 3D packaging solutions.

18.2.1 Low Power Differential Memory Interface Architecture

Figure 18.12 shows the memory interface building blocks for an x16 interface. The link consists of two-byte systems. Each byte consists of eight bidirectional data signals, three unidirectional command/address (CA) signals, and one clock signal. All of these signals are differential. There are additional, low-speed, CMOS signals for power management and other miscellaneous operations. The high-speed links operate from 2.7Gb/s–4.3Gb/s per link, based on 8:1 multiplexing. This operation range accommodates both existing and emerging LPDRAM processes. The rest of this section describes several unique technologies associated with the Mobile XDR system.

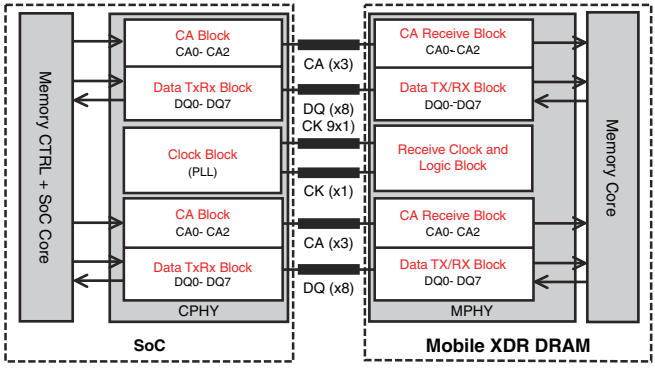


Figure 18.12 Mobile XDR Memory Interface Architecture

18.2.1.1 Asymmetric Clocking Architecture with a Fast Pause Option

Selecting the right clocking topology is one of the most critical design decisions in high-speed I/O interface design. The optimum clocking scheme for a given signaling technology primarily depends on its application. For instance, a CDR-based clocking scheme may be best suited for applications where the interface connects two chips with different clock sources. Examples include a digital cable connecting two systems, such as HDMI, or a backplane interface for SerDes applications. For on-board parallel bus applications, a forwarded clocking architecture is commonly used. In this case, there are separate TX and RX buses, and each bus has its own clock. A PLL often used in the transmitter, whereas either a PLL or DLL is used in the receiver. Such architecture is used for XDR system (not Mobile XDR) for high-speed memory applications. Chapter 10, "Clock Models in Link BER Analysis," describes the pros and cons of various clocking topologies.

FlexClocking technology is an asymmetric clocking scheme that is used in the Mobile XDR system to support fast power mode transitions, which eventually lead to savings in link power consumption. A half-bit-rate clock signal is forwarded to the DRAM, in order to directly sample and transmit data (and avoiding any closed-loop timing circuits on the DRAM, such as DLL or PLL). The clock for the DRAM core is derived from the received clock using a divider. As a result, the DRAM-side clock path can be easily paused and resumed by the controller, with a minimum hardware overhead. Like the XDR system, the Mobile XDR system also supports controller side only per-pin timing calibration. This timing adjustment is especially handy for mobile applications with 3D packages, because performing trace-length matching is very difficult, due to the small form factor.

18.2.1.2 Very Low Swing Differential (VLSD) Signaling

I/O power consumption is reduced by using very *low swing differential signaling* (VLSD) [7] [8]. Figure 18.13 illustrates the signaling interface. The output driver is an N-over-N voltage-mode differential driver, with near-ground 100mV common voltage. An on-chip linear regulator provides a 200mV transmitter supply. Each pin has its own regulator. Because the signaling is referenced to ground, the I/O supply voltage for the controller and DRAM does not need to be common, which simplifies the supply network design. The N-over-N push-pull voltage mode driver uses only one quarter of the power used by a current-mode-logic (CML) style current-mode driver with the same swing and output impedance.

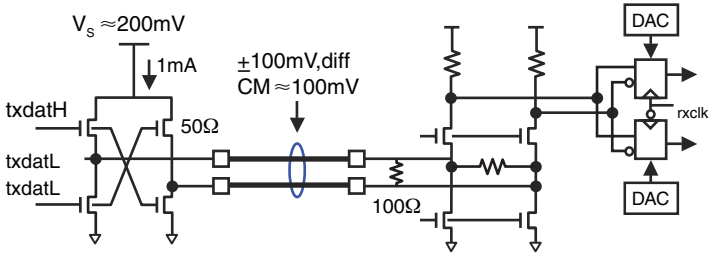


Figure 18.13 Near-Ground Voltage-Mode Differential Signaling Circuitry

The bus turnaround activity in bidirectional memory interfaces poses an additional challenge in regulator design. During the read/write bus turnaround, the transmitter loads a transient current step on V_s. To avoid charging the bypass capacitor (C_S) due to this load current, a combination of open- and closed-loop bias stages is used [6]. The open-loop current source supplies the nominal 1mA driver current, which is easily turned off. An auxiliary closed-loop regulator holds V_s, using a small fraction of the load current when the driver is inactive, enabling a fast driver turn on. Using this hybrid scheme, the driver can be power cycled in less than 5ns, which is smaller than the typical bus turnaround latency without any power recycling.

The signal is terminated differentially at the receiver, reducing the termination power dissipation significantly. The receiver input offset is trimmed with sub-mV resolution, using offset calibration digital-to-analog converter (DAC) in the data slicers. A highly sensitive receiver design is critical in a low-power signaling interface, because it allows a lower transmitter swing. This, in turn, reduces the driver size and its parasitic capacitance, allowing further reduction in the swing requirement.

18.2.1.3 Low Power Modes and Transitions

To save power during idle periods, the interface features clock pause and fast shutdown of the regulated power and PLL circuitry. As a result, the architecture supports four power modes: the *active mode* (where the I/O is fully operating), the *clock pause mode* (where the clocks in the controller and DRAM are paused), the *power down mode* (where additional bias circuits are disabled), and the *deep power down mode* (where all interface circuits, including the controller PLL, are disabled).

18.2.1.4 Multiple Data Rate Support

One of the major differences in mobile computing versus PC applications is variable data-rate support. The optimal data rate depends on the application's bandwidth requirements, to lower power consumption. Data rates can be lowered using two approaches: native low frequency transactions, and the combination of high frequency transactions and buffering. The power efficiency of the second approach depends on the buffer size and its ability to perform fast power cycling. The Mobile XDR system supports both modes. Table 18.1 shows the native supported data rates. The high-speed mode requires receiver termination and periodic timing calibration to compensate for any slow timing drift due to temperature or the power supply. The medium speed typically does not require receiver termination. Depending on the system, periodic calibration may be also disabled. For most systems, periodic calibration is not necessary for low speeds, such as High/4 and High/8. However, if a fast transition is needed to the high-speed mode, then periodic calibration must be maintained for even the medium-speed cases. The lowest data-rate mode (<200Mb/s) is directly controlled by the input reference clock frequency, so it can support any data rate below 200Mb/s. Because power saving is the purpose of this mode, no termination is used. In addition, periodic timing calibration is not recommended, because it requires unnecessary power consumption. As a result, the transition from the low-speed to the high-speed data rate has a longer latency, when compared to the other transitions.

Table 18.1 Data Rates and the Corresponding Configurations Supported by the Mobile XDR System

Speed	Representative Data Rate	Timing Calibration	Termination
High	2.7Gb/s~4.3Gb/s	Yes	Yes
Medium	High/2, High/4, High/8	Yes	Not required
Low	<200Mb/s	No	No

18.2.2 Mobile XDR and LPDDR2 Systems Comparison

LPDDR2 DRAM is one of the most commonly used DRAMs for mobile applications. The PoP package is widely used for mobile memory devices, because it provides a small form factor using a proven mass production package technology. Figure 18.14 illustrates a PoP package in mobile systems. The bottom controller chip represents the application processor, and is packaged as either a flip-chip or a wirebond. (Typical high-end application processes use flip-chip technology.) The top memory devices are packaged using PoP technology. Due to the vertical profile limitation in mobile phones, stacked devices are typically limited to two. The major drawback of this PoP package is the availability of the number of package balls.

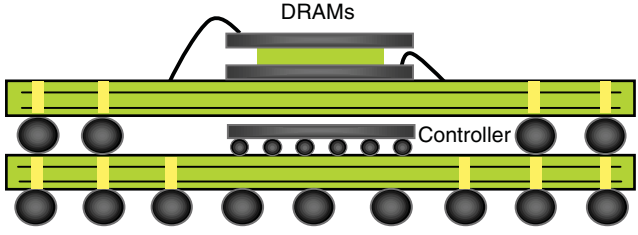


Figure 18.14 Two Stacked Dies, PoP System

To compare the number of balls required for the LPDDR2 and Mobile XDR systems, a commonly used 12mm x 12mm (0.4mm pitch, 216 balls) package is considered. Two x32 LPDDR2 memories, running at 800Mb/s per pin, can achieve a total of 6.4GB/s. Figure 18.15 shows the ball assignment. Increasing the data rate is difficult, due to the limited number of balls. On the other hand, using Mobile XDR DRAM, 12.8GB/s can be achieved using two x16 Mobile XDR DRAMs.

Two x16 DQ differential memories, operating at 3.2Gb/s per pin, can achieve a total data bandwidth of 12.8GB/s. Figure 18.16 shows the ball assignment for our differential interface. The LPDDR2 interface (at 6.4GB/s) will consume about 2.6 times more power than the differential case at the same data bandwidth, and about 1.3 times more power at 12.8GB/s.

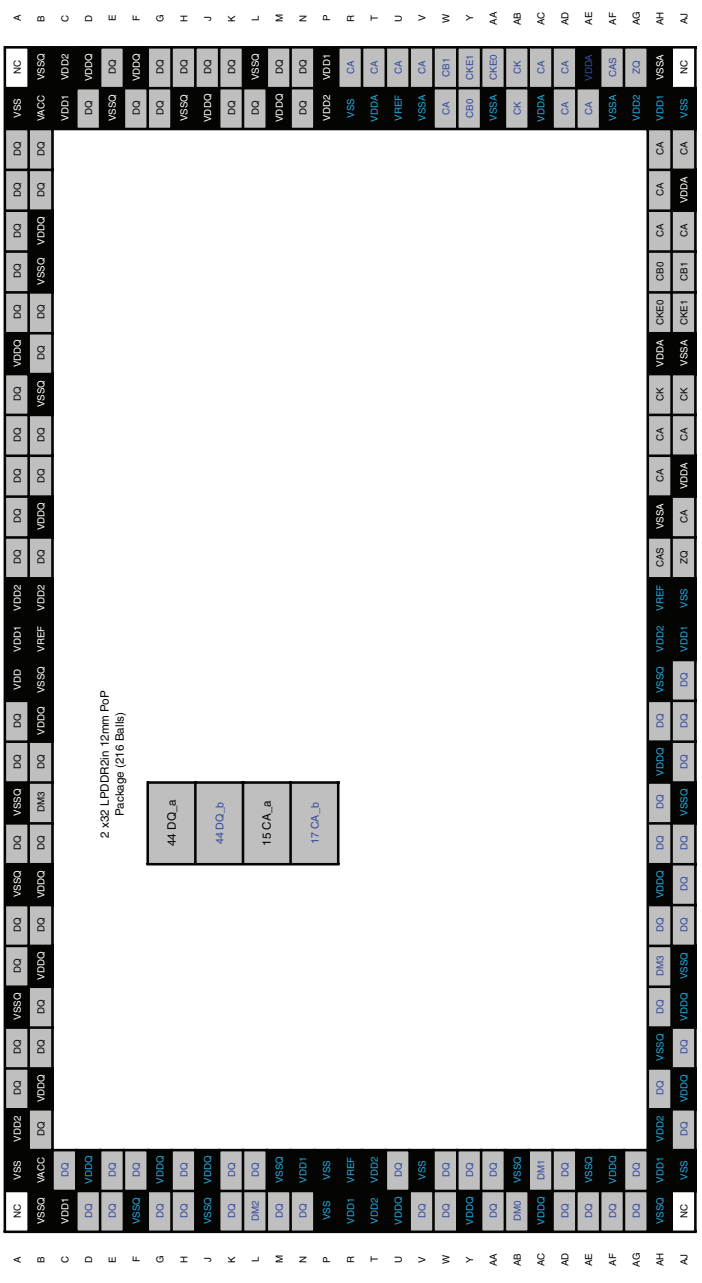


Figure 18.15 LPDDR2 Ball Assignment—JEDEC Proposal Shown for Demonstration (Actual Ball Assignment May Be Different)

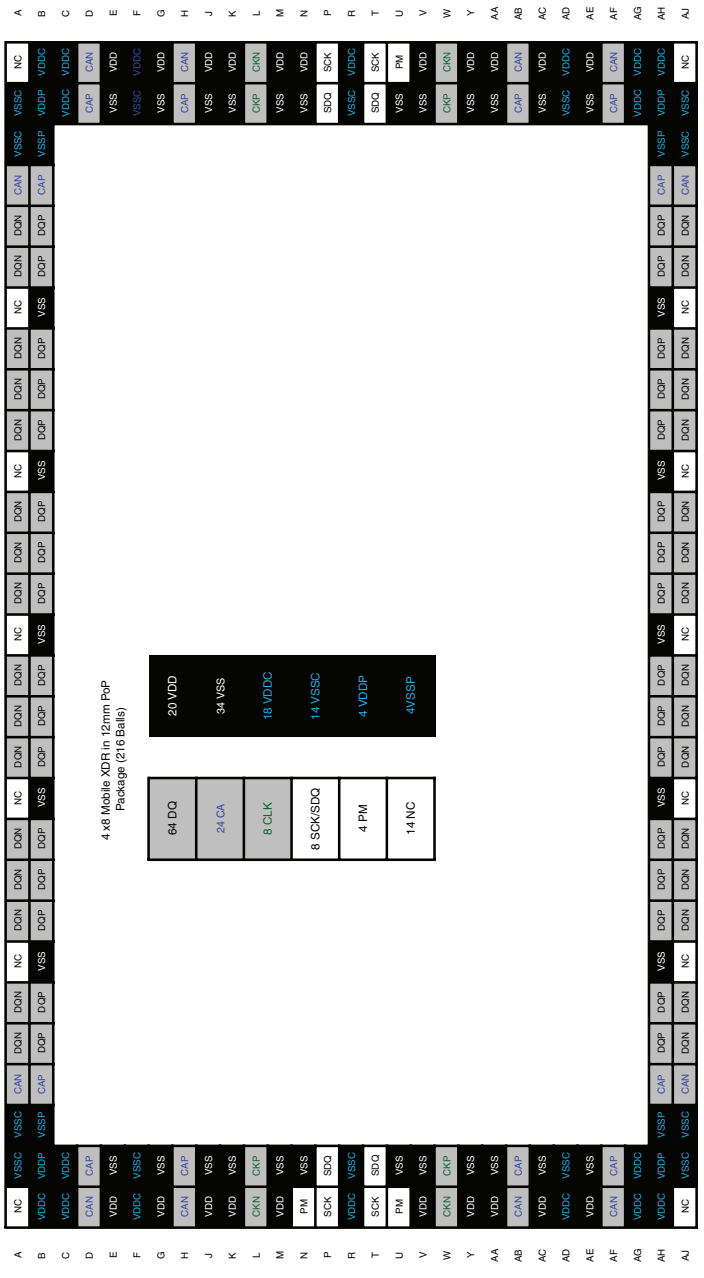


Figure 18.16 DQ x32 Ball Assignment for Mobile XDR DRAM Shown for Demonstration (Actual Ball Assigment May Be Different)

Table 18.2 compares LPDDR2 and Mobile XDR systems, in terms of signaling challenges. Due to the long wirebond, and limited routing area in the PoP package, crosstalk can be quite significant and can limit the performance of the single-ended LPDDR2 signals. For the same reasons, SSN (simultaneous switching noise—the reference voltage and power noise due to package inductance) is also large in LPDDR2. Because the package form factor is small, there is a very limited area in which to perform trace-length matching. The Mobile XDR system solves this issue using FlexPhase timing adjustments. To increase capacity, additional LPDDR2 devices can be stacked. The additional devices share the existing signal lines, causing the signals to be point-to-two-points. This multi-drop topology deteriorates the signal quality. The Mobile XDR system uses dynamic point-to-point (DPP) technology to increase the capacity, while maintaining a point-to-point topology.

Table 18.2 Comparison of LPDDR2 vs. Mobile XDR System

Issue	Details	LPDDR2	Mobile XDR
Crosstalk	Both wirebond and DRAM package trace coupling	High	Minimum (differential)
SSN	High inductance due to wirebond and package trace	High	Minimum (differential)
Power Noise	Self noise, core power coupling, package power coupling	High	Minimum (differential, on-chip regulation)
VREF Noise	Coupled power noise from supply or SSN	High	Not Applicable (differential)
Trace Length Mismatch	PoP package has limited area for trace length mismatch	High	Minimum (FlexPhase)
Multi-drop	Capacity increase causes significant ISI	Difficult	DPP
PSIJ	PSIJ modeling is a MUST for PoP system	Medium	High
In-situ Characterization	3D package interaction requires in-situ testing	N/A	Macro available

As described in Chapter 14, "Supply Noise and Jitter Characterization," power supply noise induced jitter (PSIJ) is one of the dominant timing error components in mobile systems. The majority of this jitter is due to on-chip clock routing. The Mobile XDR system has a higher sensitivity to this jitter, because it uses a higher clock frequency and has a shorter bit time. The channel budget for the Mobile XDR system accounts for the impact of this PSIJ. Finally, in-situ testing features are very important for 3D packaged systems, and the Mobile XDR system provides several on-chip measurement features, such as those described in Chapter 16.

18.2.3 Link Performance Modeling and Analysis

As shown in Table 18.2, the Mobile XDR system has several features that can effectively mitigate most of the channel-related issues. As a result, the passive channel impact, which is often the dominant timing error in typical channel analysis, is no longer the case with the Mobile XDR system, providing a clear path for future roadmaps as device processes improve. Figure 18.17 shows a PoP test vehicle for a Mobile XDR system. The test vehicle consists of two ASIC devices, based on the TSMC 40nm LP process. Figure 18.18 illustrates the channel topology. The transfer function and the single-bit response plots are shown in Figure 18.19 and Figure 18.20, respectively. Very small attenuation, crosstalk, and intersymbol interference are observed as expected.

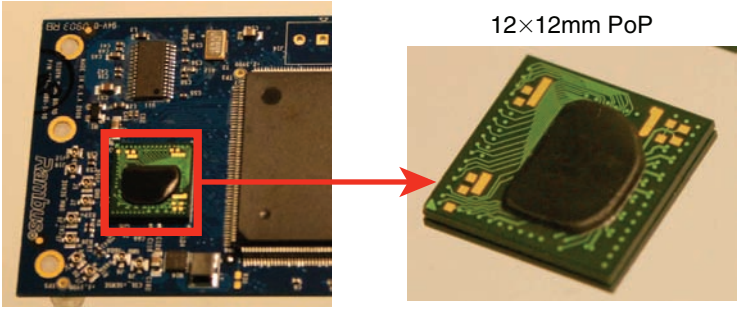


Figure 18.17 A PoP Test Vehicle for Mobile XDR System

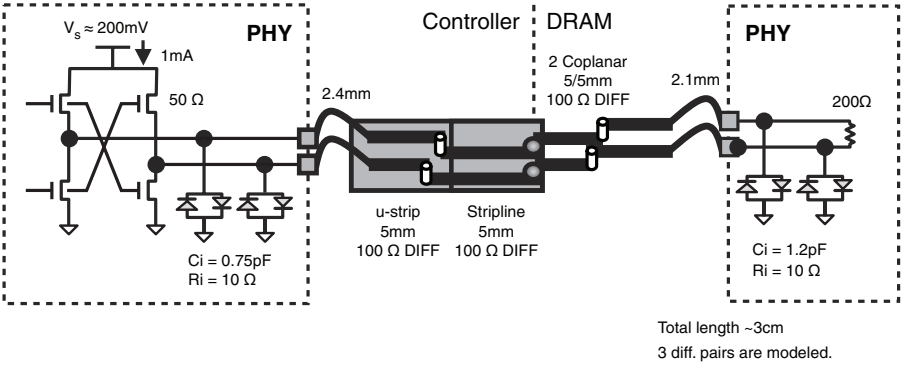


Figure 18.18 A PoP Test Vehicle's Channel Topology

Figure 18.21 shows the eye diagram without any device jitter. The overall link model is generated based on the component-level measurements to compare the link bathtub curves. (A detailed description of the correlation procedure is available [9].) Figure 18.22 shows the correlation of the timing bathtub curves. In addition to a good correlation, a wide timing margin is

observed. Due to the Mobile XDR system's short channel length, and forwarded clocking architecture, modeling jitter tracking between the data and clock signals is important (refer to Chapter 10).

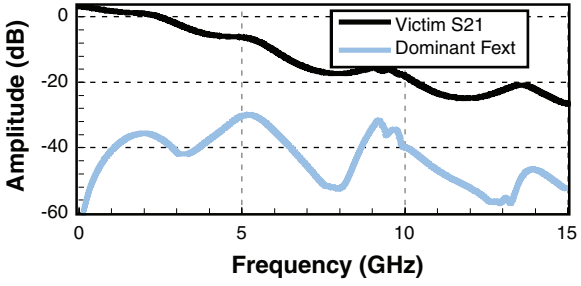


Figure 18.19 Transfer Function and XTK (FEXT/NEXT) of PoP Test Vehicle

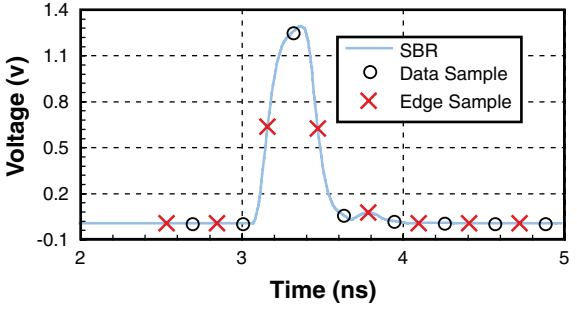


Figure 18.20 Single-Bit Response at 3.2Gb/s of PoP Test Vehicle

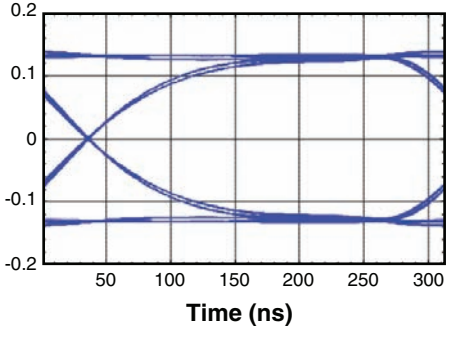


Figure 18.21 Channel Eye Diagram of PoP Test Vehicle Without Device Timing Jitter

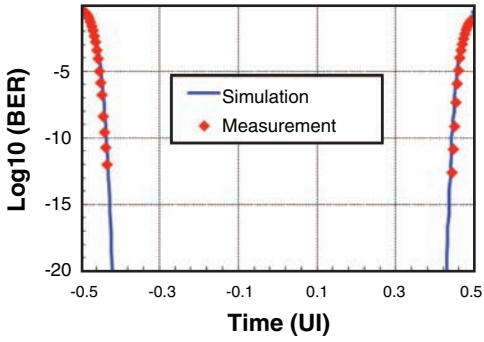


Figure 18.22 Bathtub Correlation for Read Channel

18.3 Main Memory Systems beyond DDR3

Main memory systems are used in servers, desktops, laptops, and note/netbooks. This section discusses the challenges facing these systems, and proposes solutions to those challenges. First, the trends in main memory systems are reviewed, and then the requirements for the next generation are established. Next, the current DDR3 architecture, topology, and I/O signaling are reviewed. Finally, the changes to meet the needs of main memory systems beyond the DDR3 generation are proposed [10].

DDR memory-system data rates have doubled approximately every four years, from one generation to the next, as shown in Figure 18.23. The plot shows the bandwidth of a 64-bit memory module. Module bandwidth has been steadily increasing due to the higher data rates. This pace needs to continue (or even accelerate) to meet the increasing bandwidth needs of multi-core computing, virtualization, and processor integration trends. This is especially true for high-end server systems, where the next-generation main memory data rates are expected to be in the 2GB/s–4Gb/s range.

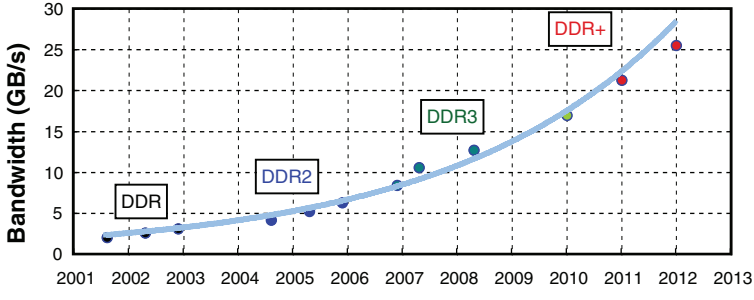


Figure 18.23 64-Bit Main Memory Data Rate and Bandwidth Trends

Low power consumption continues to be a crucial factor. For portable devices (like laptops and netbooks), lower power consumption increases the battery runtime. Desktop and server

systems need lower power consumption to comply with green initiatives or to receive energy star ratings. Finally, a lower cost of ownership is important to the server system operators. Therefore, while the data rate increases, the power envelope will remain the same (or even decrease) in the main memory systems of the future.

Figure 18.24 shows the maximum number of dual in-line memory modules (DIMMs) that a 64-bit memory controller can handle as a function of data rate. Regrettably, the memory capacity per channel has been declining, as the data rates have been increasing. This is due to signal-integrity issues, such as reflections and crosstalk, which worsen at higher data rates, especially for the stub bus systems. As a result, the memory controller can only handle a single module at the higher end of the DDR3 data rate, which effectively reduces the data path to a point-to-point topology, instead of the traditional multi-drop stub topology. A recent analysis, using the statistical approach (as opposed to the linear or simple root sum of squares approach), shows that, at best, the two-modules-per-channel limit can be pushed to 1700Mb/s [11]. However, in addition to higher bandwidth, the server systems also need higher capacity. This higher capacity, in the form of more modules per channel, is required for server systems and desired for the other systems. Consequently, more loadings are desirable as far as capacity is concerned.

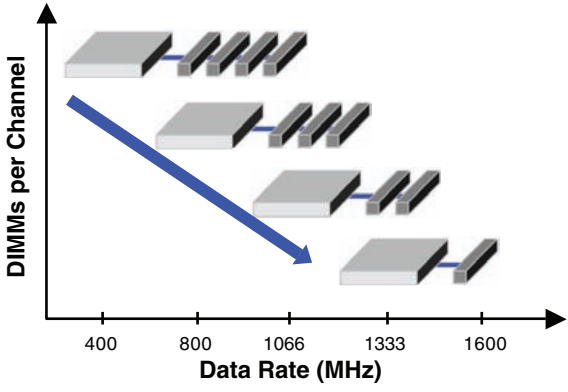


Figure 18.24 Maximum Number of Supported DIMMs Per Channel

Memory access efficiency has been decreasing. As the data rates increase, the access granularity suffers due to the increasing multiple between the interface and the core access speeds (8x to 16x). This results in an increase in the core pre-fetch time, and a sub-optimal minimum access size, due to decreased memory access efficiency and higher core power consumption.

To summarize, the four main requirements for the main memory systems of the future are as follows:

- Keep doubling the data rates (from one generation to the next), while keeping single-ended signaling, to ensure backward compatibility and low pin count. At the same time,

maintain the bus topology of the point-to-point data path (to minimize reflections/crosstalk), as well as the multi-drop command/address bus topology (to minimize pin count).

- Lower power consumption, which benefits both portable and the stationary systems. Other benefits of lower power consumption include smaller capacity heat sinks and cooling fans, which reduce cost and noise.
- Higher memory capacity, by supporting more than one module per channel. This also allows memory upgradability.
- Increase memory access efficiency, to improve throughput and lower core power consumption. This requirement can be met using module threading; however, this concept is not explored here (interested readers should refer to the white paper released by Rambus [10]). Additionally, the wide I/O solutions, mentioned later in this chapter, are not addressed because the designs are not sufficiently mature.

This section describes the features needed to meet the first three requirements. Figure 18.25 illustrates the current DDR3 architecture. The strobe, in both the Write and Read directions, is used to detect the signals in the data path. The Write data is output with the 90-degree phase shift, relative to the strobe, so that the DRAM can directly use the strobe signal to detect the signal on the data path. This makes it necessary to match the delays of the strobe and the data signals, between their respective DRAM pads, and the data sampler. The delay elements, introduced to match these two path delays, contribute to power consumption. The DRAM outputs the data in-line with the strobe signal, and the controller introduces the 90-degree phase shift needed to detect the Read data. Whereas the clock on the controller is derived from a REF_CLOCK using a PLL, the clock on the DRAM is derived from the CA_CLOCK, using a DLL. The DLL and the clock buffers are running, even in stand-by mode, resulting in DRAM clock power consumption. In server systems, where many DRAMs are in stand-by mode, this stand-by power consumption can add up to 60% of the total memory-system power consumption. So, a clocking architecture that reduces the stand-by power is needed to reduce the overall power consumption, especially for the server systems.

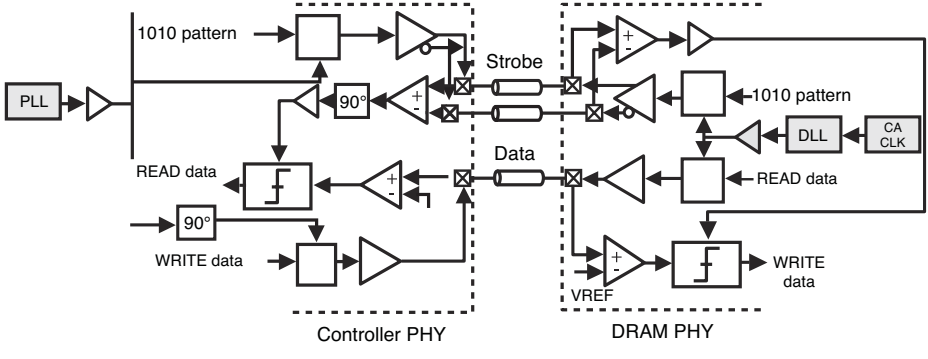


Figure 18.25 DDR3 Architecture

To reach higher data rates, the timing uncertainty between the clock and the data needs to be narrowed. The two sources of timing uncertainty in the clocking system are listed at the beginning of Section 18.1.3. As the data rates increase, the timing uncertainty between the clock and the data must be reduced in proportion to the bit time. This could be challenging for the future main memory systems; a solution is needed that automatically scales the timing uncertainty in proportion to the bit time.

Figure 18.26 shows a dual-rank, two-DIMM DDR3 stub bus system. As mentioned previously, the DDR architecture requires matching the delays of the strobe path and the data path. This means that designers must match both the on-chip path delays, and the off-chip path delays. This delay-matching requirement forces all the data traces to have the same length. The data signals are routed through multiple package, motherboard, and module PCB layers. Signals routed in multiple package or PCB layers have skews, even with matched trace lengths, when impedance variations and material parameter variations are taken into account. Even in a specific layer, some skew may be present between the data signals, due to the FR-4 fiber weave. These skews do not scale with the data rate, but they may place an upper limit on the data rate that can be achieved with the current architecture. Additionally, the length-matching requirement increases routing density, which results in higher crosstalk and reflections. Single-ended signaling systems are highly vulnerable to crosstalk, which limits the data rate. The far-end crosstalk is proportional to the coupled-trace length. Consequently, if all the data paths have the same trace length, the coupling is at its maximum. On the other hand, if the trace length-matching requirement is eliminated, designers can decrease the coupling lengths, or space the longer coupled-length traces farther apart, by taking advantage of the decreased routing density. An architecture that removes the need for delay matching the off-chip data and strobe paths will minimize the crosstalk and increase the data rates.

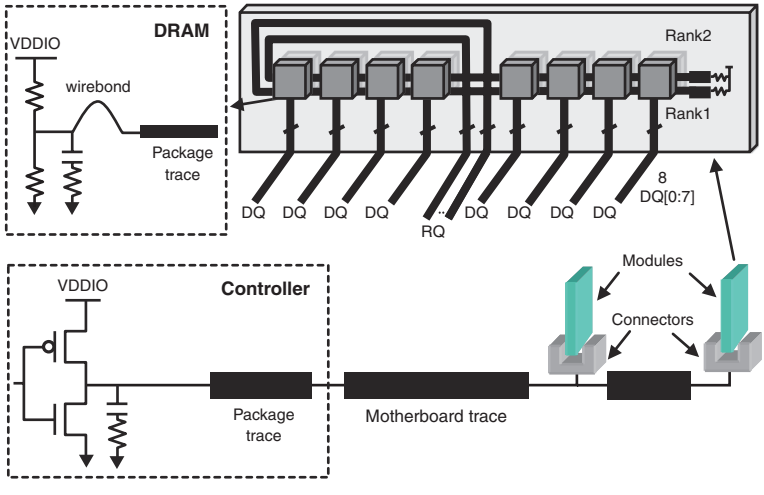


Figure 18.26 DDR3 Link Topology

Figure 18.27 illustrates DDR3 I/O signaling and power consumption. The stub series terminated logic (SSTL), used in the current DDR system, forces VDDIO to be the same on both sides of the link. This is not optimal, because the DRAM process needs a higher VDDIO voltage to achieve higher data rates, while the controller ASIC process may achieve the same data rates at a lower VDDIO voltage. Additionally, the need for a common VDDIO may necessitate thick-oxide transistors, or stacked transistor output stages, both of which are less power efficient than the thin-oxide devices. Finally, SSTL signaling requires termination to both the VDDIO and the ground rails. This uses I/O power in both the logic states. Achieving lower power consumption requires an I/O signaling method that does not require matched VDDIO on both sides of the link, and that does not use power in one of the logic states.

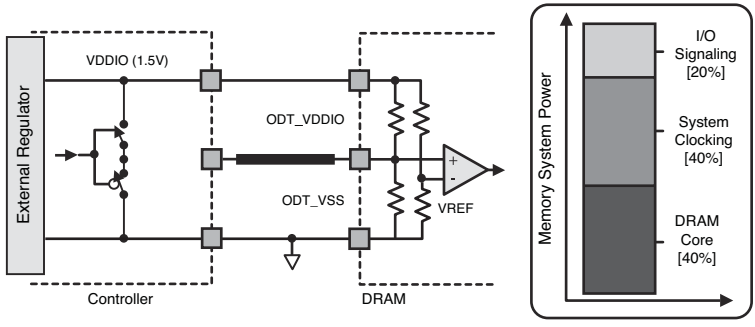


Figure 18.27 DDR3 I/O Signaling and DRAM Power Consumption in Active Mode

The two principal power modes for main memory are active Read/Write, and stand-by Idle. As mentioned previously, the stand-by power consumption of the current DDR architecture is non-zero, and constitutes a major portion of the total power consumption in the case of server systems. In active mode, the DRAM consumes power in all three areas: I/O signaling, system clocking, and DRAM core access. The bar chart in Figure 18.27 shows the approximate contribution from each area. As memory bandwidth increases, the power required for higher data rates, increased rate of core accesses, and larger number of memory channels overshadows the power savings from lower supply voltages. Consequently, reducing active memory power is an important consideration for all current compute platforms [10]. Although the I/O power only accounts for about 20% of the total active power, any reduction is helpful in reducing the overall power consumption.

18.3.1 FlexPhase Timing Adjustment to Enable Higher Data Rates

FlexPhase technology (described in Section 18.1.3) is used on the controller side to account for PVT variations and to eliminate the need for delay matching the strobe and data signals. Figure 18.28 shows the DDR architecture with FlexPhase timing adjustment. The static error can be

greatly reduced by performing the per-pin timing calibration, using the FlexPhase timing adjustment circuitry. FlexPhase technology uses timing adjustment circuitry in the memory controller to account for pin-to-pin timing variations in both the Write and the Read directions. This keeps the DRAM design simple and keeps its cost low. During a write operation, the data is sent with different delays, but it reaches the DRAM at the same time. During a read operation, the DRAM sends the data without any delay adjustment, and the controller manages the timing mismatch by sampling the data at different sampling times. The timing adjustment can be performed during the system initialization stage, and periodically update it during the DRAM refresh periods.

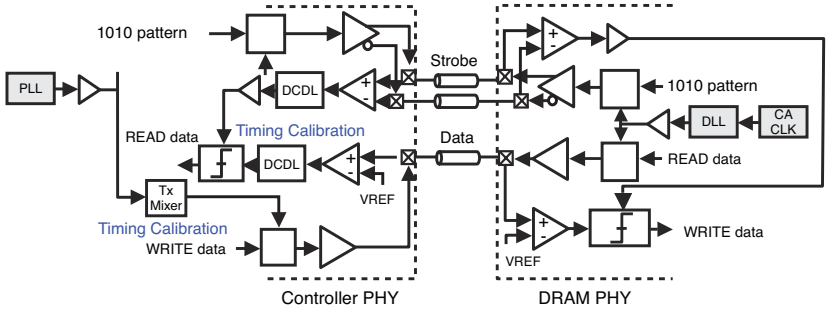


Figure 18.28 DDR Architecture with Implementation of FlexPhase Timing Adjustment

Using FlexPhase technology simplifies trace routing and minimizes crosstalk, because trace-length matching is no longer required. Figure 18.29(a) shows a typical motherboard design with matched trace lengths. Figure 18.29(b) shows the design without trace-length matching. The eye diagrams for both motherboard designs (with and without trace matching) are simulated, and are shown in Figure 18.30 and Figure 18.31, respectively. FlexPhase significantly reduced crosstalk noise.

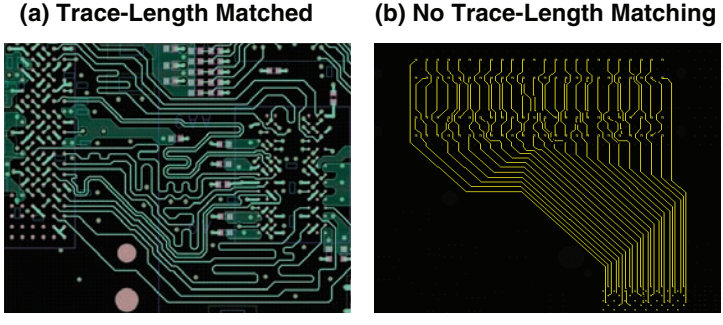


Figure 18.29 Routing Example with and without Trace-Length Matching

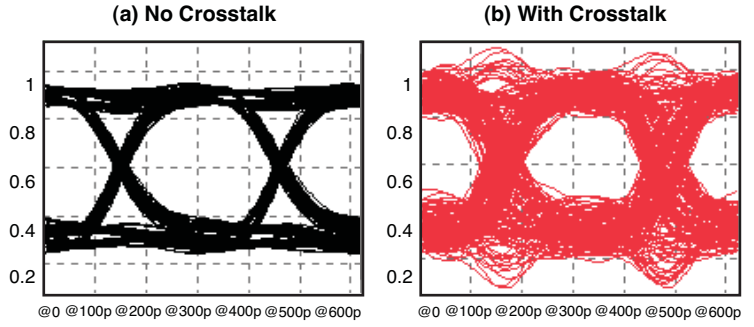


Figure 18.30 Eye Diagram with Trace-Length Matching in a One-DIMM System

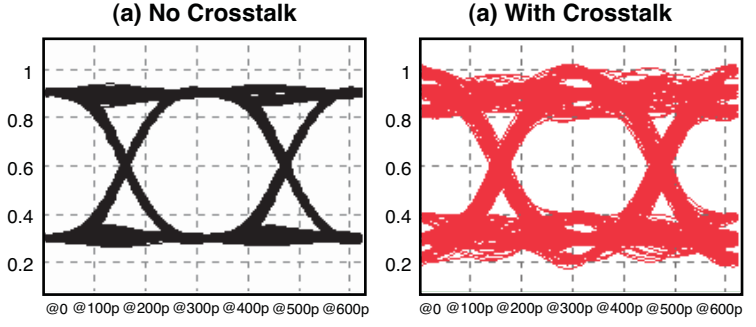


Figure 18.31 Eye Diagram without Trace-Length Matching in a One-DIMM System

18.3.2 FlexClocking Architecture to Reduce DRAM Stand-By Power

The clocking architecture shown in Figure 18.32 incorporates the FlexClocking technology, which is also used in the Mobile XDR system. The key features are very low stand-by current with fast turn-on times. The FlexClocking architecture enables data alignment, without the use of a DLL or PLL on the DRAM device. FlexClocking technology adjusts for timing variations between the clock and data signals, and does not consume stand-by clocking power between data transactions.

To minimize power when the DRAM is not actively transmitting or receiving data, the controller transmits a high quality, differential clock (called "Data Clock") to the DRAM module. The Data Clock is routed using a topology similar to the Command/Control/Address (CCA) bus, and its associated clock. The Data Clock and Timing Reference Signals are active only during data transactions and are enabled with fast turn-on times. The Data Clock oscillates at the Nyquist frequency of the data rate. When combined with the FlexClocking architecture using a separate data clock signal, running at the Nyquist rate (rather than the C/A clock) can enable high data rates without a DLL or PLL on the DRAM.

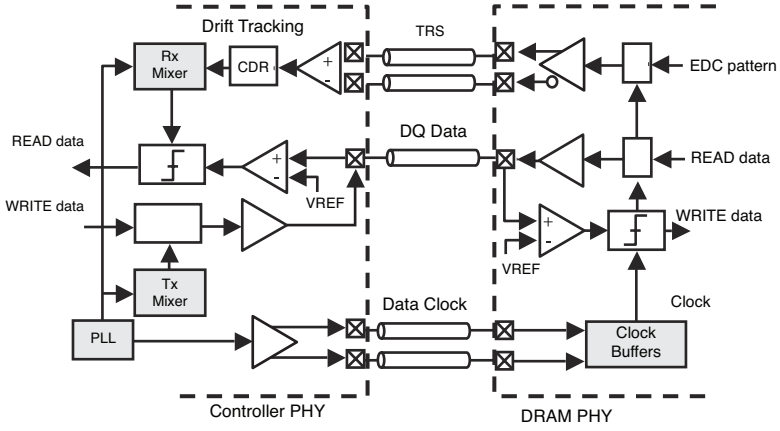


Figure 18.32 FlexClocking Architecture

The clock buffer circuit on the DRAM, which captures "Data Clock" and distributes it to the I/O bit-slices, must have a very low timing variation due to power supply noise. Power supply noise can be as high as 30mV–50mV on a DRAM device, even with excellent isolation and bypassing. Clock buffering with a differential Current-Mode Logic (CML) buffer improves immunity to power supply noise by distributing the clock signal differentially across the DRAM PHY. Though a CML buffer consumes more power than a CMOS buffer for the same fan out, the CML buffer is only activated when data transactions are active, and can be disabled by a signal sent by the controller to the active DRAM module.

Future main memory DRAM devices could implement an optional mode for read operations, in which a read capture signal is output from the TRS pin to the memory controller, in the same manner that DDR3 uses the Data Strobe today. With per-pin calibration, the DRAM's TRS would no longer be required to be a strobe signal and could carry other useful information, such as EDC, for high-reliability systems. This signal can also be periodically edge-tracked by the memory controller to maintain the calibrated timing integrity.

18.3.3 Near Ground Signaling to Reduce I/O Power

In this section, Near Ground Signaling (NGS) is proposed as a way to reduce the active I/O power consumption, as shown in Figure 18.33. By using Near Ground Signaling technology, the memory controller's I/O voltage (VDDIO) can be well below the maximum voltage of thin-oxide devices (typically 0.9V to 1.0V for 45nm and below process technologies). Meanwhile, the DRAM, on the other side of the channel, can continue to utilize the higher voltage (1.2V–1.35V) that its on-chip circuits require to achieve reliable high data-rate signaling.

Near Ground Signaling has a reduced signal swing when compared to traditional SSTL or POD, which substantially lowers I/O power on both sides of the link (compared to DDR3). An internally regulated (500mV DC) supply powers the I/O driver circuits to provide a low I/O signal swing. I/O signals swing from the ground rail (low) to a high of between 250mV–300mV.

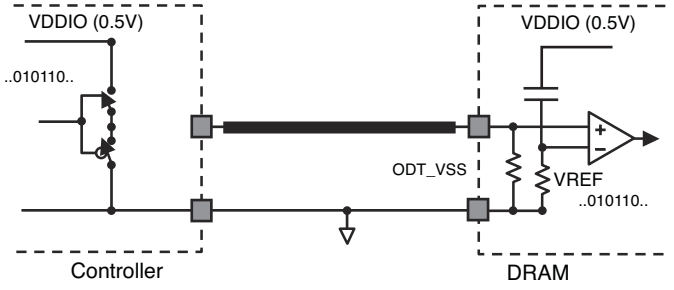


Figure 18.33 Near Ground Signaling (NGS)

Besides requiring lower signaling power, the lower signal swing reduces the size of the I/O driver circuit, enabling even further power reduction in the pre-driver and clock distribution circuits. From a cost perspective, Near Ground Signaling also eliminates the need for thick oxide transistors on the memory controller, potentially saving at least two additional masks and two additional semiconductor-processing steps.

Table 18.3 summarizes the I/O power simulation results using typical operating voltages for SSTL and NGS signaling. NGS signaling offers significant power savings, even when the power consumption of the 0.5V on-die regulator is considered. To further reduce power consumption, NGS can also use data bus inversion (DBI) coding (described in Chapter 12, "SSN Modeling and Simulation"). In conjunction with DBI coding, NGS can lower the worst-case SSO noise by 40%. The termination to ground used by the NGS signaling can also help reduce simultaneous switching output noise (SSO), as the ground rail typically has the lowest-impedance.

Table 18.3 Comparison of SSTL and NGS Power Consumption

	VDDIO	VDD (Controller/DRAM)	Power-I/O
SSTL-1.5	1.5V	0.9/1.5V	22.8mmW
SSTL-1.2	1.2V	0.9/1.2V	14.7mW
NGS	0.5V	0.9/1.2V	1.9mW

NGS requires a sensitive receiver with offset calibration, which can lead to some DRAM testing issues. Another drawback of Near Ground Signaling is the level-shifting circuits that are required to be in the receive path of the DRAM. One solution is to use high-speed, common-gate NFET level-shifters, which consume low power and can be disabled when there are no active data transactions [10].

Figure 18.34 illustrates the NGS signaling performance under the worst-case configuration of the DPP topology [12]. Data is transmitted to Rank1 device, which is unterminated, and Rank2

device is terminated with 40Ω. The received eye diagrams at both devices are shown in Figure 18.34. To examine the impact of SSO noise on the voltage and timing margins, the channel model (shown in Figure 18.33) with both ideal and non-ideal power supply models are simulated [4]. Figure 18.34 shows the additional margin loss caused by the worst-case SSO noise, while the channel simulation with the ideal power supply shows only crosstalk and ISI effects. The eye opening of the open-terminated Rank1 is significantly more degraded by SSO than that of the 40Ω-terminated Rank2.

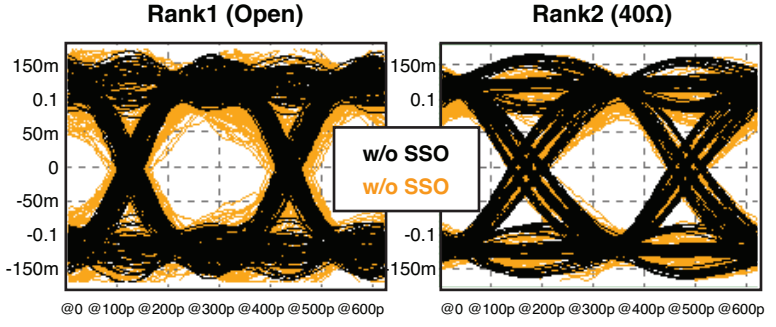


Figure 18.34 Signaling Performance and SSO Impact at 3200Mb/s

18.3.4 Memory Capacity Increase with Dynamic Point-to-Point Link Topology

Dynamic Point-to-Point (DPP) topology (described in Section 18.1.5) can be used to extend capacity without sacrificing the bandwidth of the memory system. Figures 18.6 and 18.7 demonstrated the DPP concept. Figure 18.35 is a comparison of DQ channel performance with DPP topology, at 3.2 GB/s, with one populated DIMM versus two populated DIMMs. The worst-case signal integrity is observed when only one module is loaded, and the signals must cross a continuity module, adding two extra connector crossings (see Figure 18.6), which results in additional connector crosstalk and reflections. With both modules loaded in a DPP configuration, the signaling improves, because the electrical path is shortened and the extra connector crossing is eliminated (see Figure 18.7). With DPP, two modules can be supported per channel, because there is enough margin to sustain the data rate, even with the extra connector transitions. The worst-case eye opening for the RQ bus occurs at the last DRAM (see Figure 18.36) with a data rate of 1600 Mb/s. Note that there is still sufficient eye opening for reliable operation of the RQ bus at this data rate. There is no additional degradation of the RQ channel due to the DPP topology, as each RQ channel supports only one active module, and no additional connector transitions are involved.

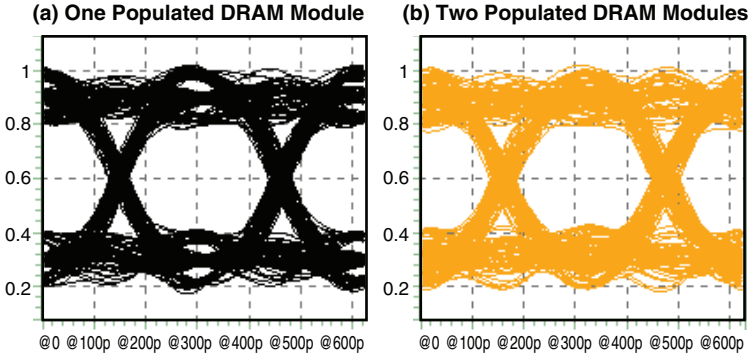


Figure 18.35 DQ DPP Eye Diagram Comparison at 3200Mb/s

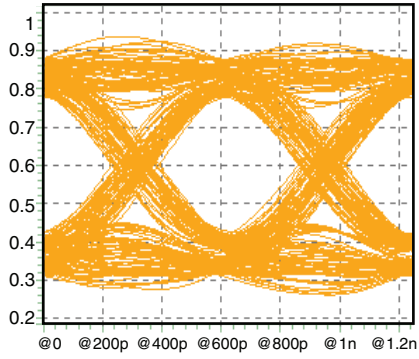


Figure 18.36 The RQ Eye Diagram at the Last DRAM at 1600 Mb/s

18.4 Future Signaling Systems

Driven by the convergence of computing, communication, and consumer devices, the need for higher data bandwidths will continue to increase, with no end in sight. To put this into perspective, Figure 18.37 provides a summary of the data bandwidth of various I/O interface designs over the past decade and includes a prediction for the future. In the near future, one can expect high-end graphics applications to require more than 1TB/s of data bandwidth. However, three fundamental limits, or "walls," stand in the way of ever-increasing bandwidth; this section covers each of them.

First, there has been a dramatic increase in data bandwidth for chip-to-chip communication, and for memory interfaces, over the last decade. As of this writing, the differential signaling data rate is approaching 25Gb/s, even for a backplane channel with a 12-inch PCB trace [13] [14]. To push data rates higher, one must be able to recover signals from a channel with a 30dB+ loss. A major question is: Can we reach 50Gb/s when the bit time is only 20ps? Even if this were

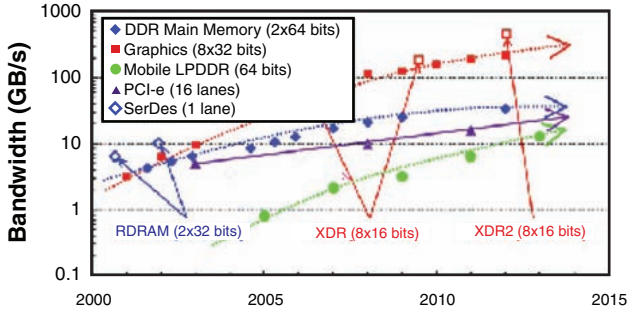


Figure 18.37 I/O Bandwidth Requirements

possible, producing such a device in high volume would be very difficult, because a 1-ps loss in time margin amounts to 5% of the bit time.

Although it is generally believed that single-ended signaling reached its end-of-life at about 6- to 8-Gb/s (due to crosstalk, SSO, and reference voltage noise), an advanced single-ended signaling scheme has been recently demonstrated that it can be extended to 12.8Gb/s [15]. However, even with this potential solution, it is not clear how much single-ended signaling data rates can be improved. This is the "data-rate wall."

Second, as process shrinks below 28nm, additional DFM restrictions are placed on the circuit design. Furthermore, device variability becomes very difficult to control. In particular, the DRAM transistor's performance is not improving as much as those in the ASIC process are. The general belief is that DRAM process cannot support data rates much higher than 16Gb/s. This is the "process-scaling wall."

Third (and most importantly), power has become a dominant concern: not only for mobile devices, but also for the servers used in datacenters. High-end graphics cards already consume more than 500W, and a memory interface can consume as much 100W! The power consumption of modern I/O interface designs has reached the limit where the designs can be cooled using "reasonable" solutions. For practical and economical reasons, we can only improve data bandwidth within the same power envelopes of previous generations. Consequently, power efficiency must be improved to make this possible. This is the "power wall."

What will it take to scale these three "walls?" To address the "data-rate wall," the first (and most obvious) requirement is great innovations in system architecture, circuit architecture, and interconnects. In addition to increasing data rates to achieve higher bandwidth, the number of links (or channels) could be increased. To overcome channel loss, one could explore low-loss materials, and reduce the channel length by using different interconnect technologies.

Second, to overcome the "process-scaling wall," we could explore disaggregated DRAM architecture, which uses an ASIC buffer for the faster (or primary) chip-to-chip interface, and a wider and slower (or secondary) interface for the DRAM interface. This approach also enables us to achieve as much as 2TB/s of memory bandwidth for graphics, with the same power envelope

used by today's high-end graphics card. For low-power designs, radically different signaling and circuit architectures must be developed that keep interface power proportional to the amount of data transmitted. In addition, we must remove the redundancy in the design that guarantees performance at the cost of power consumption.

Two emerging technologies that may address the issues of bandwidth and power consumption are wide I/O technology and optical links. The following sections briefly introduce these two technologies.

18.4.1 Wide I/O

Perhaps, the easiest way to increase the I/O bandwidth is by simply adding more channels. This brute-force approach quickly runs out of steam due to the limited number of package pins and PCB routing area. Wide I/O technology allows a massive number of connections that use a relatively low data rate. The signaling aspect of wide I/O is less challenging, because it operates at a considerably lower data rate than serial and narrow I/O. It also uses less power, because it requires no complex timing circuitry or data-path design.

Because wide I/O is based on a large number of connections, the interconnect structures (including balls, traces, and vias) must be small. Only 3D packaging solutions, such as direct fact-to-face die attachment, multi-chip packaging (MCP) modules, and through-silicon vias (TSV), support wide I/O interfaces. 3D packaging solutions are not widely used in computing application yet, due to expensive packaging costs and limited flexibility. The recent popularity of smart phones has fueled the usage of 3D packaging, based on package-on-package (PoP), which still provides a limited number of pins. However, it has opened the door for more expensive 3D packaging solutions, because high packaging costs are justified by a smaller form factor, and better bandwidth and power performance. In addition, flexibility is not a critical requirement in mobile applications.

TSV is a great future packing technology because it could provide a massive number of interconnects. However, its adaptation by I/O interfaces has been slow due to thermal management, reliability, testing, integration, as well as economic issues among different IC manufactures. While the ultimate goal of 3D integration is the direct attachment of all dies using TSVs, the likelihood of this event is low, because it requires a company that can vertically integrate all the components. A more likely solution for the near future is the silicon or substrate interposer. Figure 18.38 shows a 3D memory system with a silicon interposer. TSVs are only used at the interposer. Because of the extra thickness/area required by the interposer, it is unlikely that this solution will be used in the mobile space. However, this configuration is ideal for graphics systems, because it can provide large bandwidth with minimum power consumption. In a graphics application, the short channel length of the 3D system also helps to boost the per-pin data rate.

Samsung introduced the first wide I/O memory product for mobile applications in 2011 [16]. It consists of a 512-bit I/O with a 200-MHz clock that achieves a bandwidth of 12.8GB/s. It uses microbumps with a $20\times17\mu m^2$ size and 50μm pitch. The power consumption is only 0.78mW/Gb/s, which is 4.5% of an LPDDR device. The memory capacity could be increased from 1Gb to 2Gb by stacking two wide I/O DRAMs, using TSVs. TSV stacking was demon-

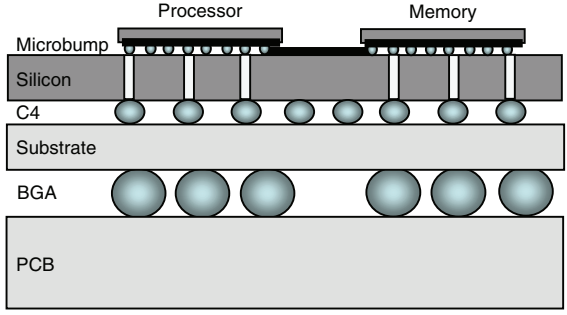


Figure 18.38 Wide I/O Memory System with Silicon Interposer

strated using a 75μm diameter. However, the yield loss, due to the TSV process, was roughly 70%. TSV is potentially a great technology with which to increase memory capacity; Samsung demonstrated its usage in mobile or graphics systems [16]. NEC used a separate interface device to buffer memory transactions and eliminate the high-end interface circuits from the DRAM devices, which are stacked on a silicon interposer [17] (see Figure 18.39). Samsung applied this concept to PC and server memory systems [18]. Four 2-Gb DDR3 devices are stacked, using TSVs, to make an 8-Gb 3D DDR3 system that supports four ranks. Fifty percent redundant TSVs were used to increase the yield to >98% from 15%.

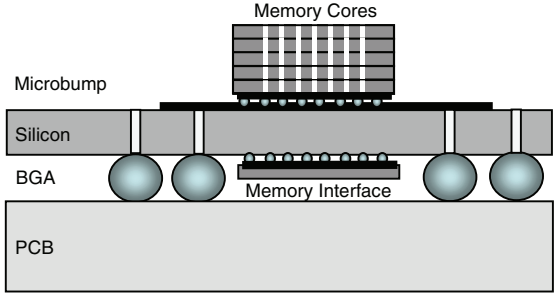


Figure 18.39 3D Stacked Memory Interface Module with a Silicon Interposer

18.4.2 Optical Link

Electrical signaling over copper inherently suffers from serious signal-integrity issues as data rates reach the Gb/s region. The distance of electrical signaling decreases as data rates go higher. Compared with electrical signaling over copper, optical communication over fiber channel is much less sensitive to distance and great for long-haul applications. Historically, the transition from electrical links to optical links happens at a bandwidth-distance product of $100\text{Gb/s} \cdot \text{m}$

[19]. For example, optical communication, now at 25Gb/s, has penetrated into rack-to-rack communication in data centers (about 1–10 meters).

In addition to the bandwidth-distance product, which has historically been the driver behind the transition from electrical signaling to optical signaling, currently two other important drivers may accelerate the transition: bandwidth-density and power envelope. ITRS projects 10TB/s I/O bandwidth for multi-core chips by 2015 [20]. Assuming that the power limit for a high-performance multi-core chip is 150W (with half of that used for I/O with 10-Gb/s signaling and 1-mW/Gb/s power efficiency—an optimistic estimation); it requires 15K signal pins to provide a 10-TB/s bandwidth! ITRS predicts that the pin count by 2015 will be approximately 4K for high-performance chips [20]. If we scale the per-pin bandwidth, and use high-speed signaling, (for example, 50Gb/s with 10-mW/Gb/s efficiency—a very aggressive estimation), then the I/O power envelope limits the total bandwidth to 1TB/s.

There is no clear roadmap for electrical signaling to support high-performance processor scaling. The wide I/O solutions discussed earlier only partially alleviate the problem, because there is a limit to how many chips can communicate efficiently over wide I/O. In contrast, multiple optical communication channels are readily available over a single optical fiber, via *dense wavelength division multiplexing* (DWDM). This provides yet another scaling dimension that is not available in electrical signaling.

Over the past decade, the industry has addressed the problem of high-performance processor support, and many efforts were directed towards making optical links a viable solution for short-distance communication, in terms of integration density, cost, and power-efficiency. Traditionally, optical links consist of separately packaged discrete optical components that are manually assembled for optical alignment. This has prevented optical communication from benefiting from cost and power scaling. Recently, the integration trend that occurred in silicon technology has started in optics. For example, Avago Micropod technology [21] integrates 12 VCSELS (or photo-detectors) on a single die, providing 12 parallel optical links. Infinera takes this one step further, and integrates hundreds of photonic devices on one die in their Photonic Integrated Circuit (PIC) [22]. Silicon photonics technology takes yet another step, and integrates not only photonic devices, but also transistors, on a single die, based on a CMOS-compatible process with minimal changes. Luxtera [23] ships 4x10Gb/s active optical cables based on silicon photonics. Recently, they have highlighted their 4x25Gb/s link at SC 2010 as a disruptive technology. Others firms, such as IBM [24] and Intel [25], are also investing heavily in developing silicon photonics technology. IBM demonstrated a test chip consisting of six WDM channels, with integration density as high as $0.08mm^2$ –$0.5mm^2$ per transceiver, including the electronic backend and all photonic devices except the laser source [24]. Intel demonstrated a 50Gb/s photonic link in July 2010, in which they integrated only photonic devices (including a hybrid laser source) on one die [25].

With the ability to print photonic devices similar to transistors, silicon photonic technology provides scalability, along with the possibility of lowering the cost of optical communication over the years. Additionally, when photonic devices are integrated with an electronic back-end, the power consumed by modulating photonic devices is minimized due to less parasitics.

Luxtera's products offer ~20mW/Gb/s power efficiency at a cost of $1~$2/Gb/s [23]. By 2016, optical links may consume as low as 5mW/Gb/s power and cost as little as $0.17/Gb/s [26]. This level of power efficiency and cost will then be comparable to current electrical signaling solutions. Currently, electrical signaling offers 1 mW/Gb/s~20mW/Gb/s power efficiency, depending on data rates, and costs ~$0.1/Gb/s for short-reach applications.

Even with these exciting developments in silicon photonics, the technology is still immature. Reliability and yield are still under investigation. The biggest upside of optical signaling, compared to electrical signaling, is DWDM. However, a power- and cost-efficient solution for DWDM with silicon photonics is yet to be developed. Furthermore, current manufacturing and packaging technologies are optimized for electrical signaling. For silicon photonics to have a low enough cost to allow them to compete with electrical signaling for short-reach applications, a cost-effective system-level optimization has to take place. On the other hand, wide adoption drives system-level optimization. This is a typical chicken-and-egg problem. Nevertheless, it is evident that, in the near future, optical signaling is going to be part of the solution space for short-reach applications that are less cost-sensitive.

References

1. K. Chang, S. Pamarti, K. Kaviani, E. Alon, X. Shi, T. Chin, Jie Shen, G. Yip, C. Madden, R. Schmitt, and C. Yuan, F. Assaderaghi, and M. Horowitz, "Clocking and circuit design for a parallel I/O on a first generation CELL processor," in *International Solid-State Circuits Conference Digest of Technical Papers*, 2005, pp.526–615.
2. W. Beyene, A. Torres, N. Cheng, A. Vaidyanath, J. Kizer, H. Nguyen, and C. Yuan, "Characterization and hardware correlation of multi-gigahertz parallel bus with transmit pre-emphasis equalization," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, Oct, 2004, pp. 177–180.
3. J. Kim, R. Schmitt, D. Oh, W. Beyene, M. Li, A. Vaidyanath, Y. Lu, J. Feng, C. Yuan, D. Secker, and D. Mullen, "Design of low cost QFP packages for multi-giga memory interface," in *Proceedings of IEEE Electrical Performance of Electronic Packaging Conference*, May 2009, pp. 1662–1669.
4. R. Schmitt, J.-H. Kim, J. Feng, D. Oh, and C. Yuan, "Analyzing signal and power integrity limitations for mobile memory systems in 3D packaging environments," presented at the IEC DesignCon, Santa Clara, CA, 2009.
5. D. Oh, W. Kim, J.-H. Kim, J. Wilson, R. Schmitt, C. Yuan, L. Luo, J. Kizer, J. Eble, and F. Ware, "Study of signal and power integrity challenges in high-speed memory I/O designs using single-ended signaling schemes," presented at the IEC DesignCon, Santa Clara, CA, 2008.

6. R. Palmer, J. Poulton, B. Leibowitz, Y. Frans, S. Li, A. Fuller, J Eyles, J. Wilson, M. Aleksic, T. Greer, M. Bucher, and N. Nguyen, "A 4.3GB/s mobile memory interface with power-efficient bandwidth scaling," in *Symposium on VLSI Circuits Digest of Technical Papers*, 2009, pp. 136–137.
7. K.-L. J. Wong, H. Hatamkhani, M. Mansuri, and C.-K. K. Yang, "A 27-mW 3.6-Gb/s I/O transceiver," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 4, pp. 602–612, Apr. 2004.
8. J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz, "A 14mV 6.25-Gb/s transceiver in 90nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2745–2757, Dec. 2007.
9. D. Oh, S. Chang, C. Madden, J.-H. Kim, R. Schmitt, M. Li, C. Yuan, F. Ware, B. Leibowitz, Y. Frans, and N. Nguyen, "Design and characterization of a 12.8GB/s low power differential memory system for mobile applications," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct. 2009, pp. 33–36.
10. "Challenges and solutions for future main memory," Rambus white paper, May, 2009.
11. S. Chaudhari, J. A. McCall, and J. Salmon, "Proposal for BER based specifications for DDR4," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, Oct 2010, pp. 121–124.
12. J.-H. Kim, D. Oh, R. Kollipara, J. Wilson, S. Best, T. Giovannini, I. Shaeffer, M. Ching, and C. Yuan, "Challenges and solutions for next generation main memory systems," in *Proceedings of IEEE Electrical Performance of Electronic Packaging and Systems Conference*, pp. 93–96, Oct. 2009.
13. A. Healey, C. Liu, and W. Jin, "FEC for high-speed SerDes link system of 25-28Gb/s," presented at the IEC DesignCon, Santa Clara, CA, 2011.
14. M. Shimanouchi, M. Li, and S. Gabriel, "Architecture/configuration explorations for 25-28Gb/s interconnects," presented at the IEC DesignCon, Santa Clara, CA, 2011.
15. A. Amirkhany et al, "A 12.8Gb/s/link tri-modal single-ended memory interface for graphics applications," in *Symposium on VLSI Circuits Digest of Technical Papers*, 2011, pp. 232–233.
16. J.-S. Kim et al., "A 1.2V 12.8GB/s 2Gb mobile wide-I/O DRAM with 4x128 I/Os using TSV-based stacking," in *International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2011, pp. 496–497.
17. M. Kawano, N. Takahashi, Y. Kurita, K. Soejima, M. Lomuro, and S. Matsui, "Three-dimensional packaging technology for stacked DRAM with 3-Gb/s data transfer," *IEEE Transactions on Electron Devices*, vol. 55, no. 7, pp. 1614–1620, Jul. 2008.
18. U. Kang et al., "8 Gb 3-D DDR3 DRAM using though-silicon-via technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 111–119, Jan. 2010.

19. A. V. Krishnamoorthy, "Photonics-to-electronics Integration for Optical Interconnects in the early 21st century," *Optoelectronics Letters*, vol. 2, No. 3, 163–168.
20. ITRS Roadmap, see http://www.itrs.net/Links/2007ITRS/Home2007.htm
21. http://www.avagotech.com/pages/en/press/micropod
22. http://www.infinera.com/technology/PIC.html
23. http://www.luxtera.com
24. http://www.research.ibm.com/photonics/
25. http://download.intel.com/pressroom/pdf/photonics/50G_Silicon_Photonics_Link.pdf?iid=pr_smrelease_vPro_materials1
26. A. Benner, "Cost Effective Optics: Enabling Exascale Roadmaps," *Hot Interconnect*, 2009.

This page intentionally left blank

Index

Numbers

A

B

C

D

E

F

G-H

I

J

K-L

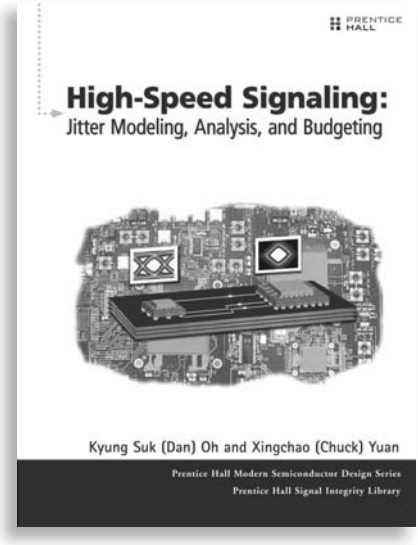
M

N

Q–R

S

T



FREE Online Edition

Your purchase of ***High-Speed Signaling*** includes access to a free online edition for 45 days through the Safari Books Online subscription service. Nearly every Prentice Hall book is available online through Safari Books Online, along with more than 5,000 other technical books and videos from publishers such as Addison-Wesley Professional, Cisco Press, Exam Cram, IBM Press, O'Reilly, Que, and Sams.

SAFARI BOOKS ONLINE allows you to search for a specific answer, cut and paste code, download chapters, and stay current with emerging technologies.

Activate your FREE Online Edition at www.informit.com/safarifree

- **STEP 1:** Enter the coupon code: YUAEWFA.
- **STEP 2:** New Safari users, complete the brief registration form. Safari subscribers, just log in.

If you have difficulty registering on Safari or accessing the online edition, please e-mail customer-service@safaribooksonline.com

Safari Books Online

Addison Wesley · Adobe Press · Alpha · Cisco Press · FT Press · IBM Press · lynda.com · Microsoft Press · New Riders · O'Reilly · Peachpit Press · Prentice Hall · Que · Redbooks · Sams · SAS Publishing · Sun Microsystems · Wiley