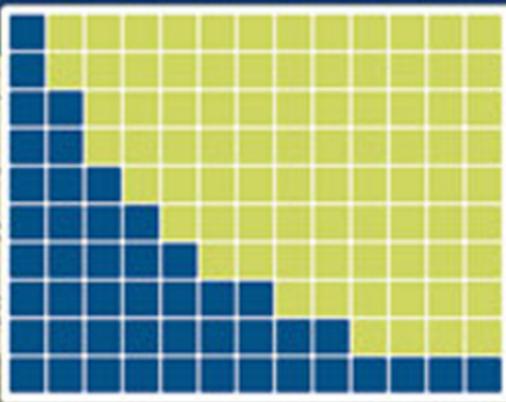


Jose Moreira • Hubert Wekmann

**AN ENGINEER'S GUIDE TO
AUTOMATED
TESTING OF
HIGH-SPEED
INTERFACES**



An Engineer's Guide to Automated Testing of High-Speed Interfaces

For a listing of recent titles in the *Artech House Microwave Library*,
turn to the back of this book.

An Engineer's Guide to Automated Testing of High-Speed Interfaces

José Moreira
Hubert Werkmann



**A R T E C H
H O U S E**

BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data

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

ISBN-13 978-1-60783-983-5

Cover design by Igor Valdman

© 2010 ARTECH HOUSE

685 Canton Street

Norwood, MA 02062

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher. All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

10 9 8 7 6 5 4 3 2 1

Para os meus pais Inês e José e o meu irmão Carlos
— José Moreira

For everyone who supported me getting to the next levels
— whenever, wherever
— Hubert Werkmann

Contents

Preface	xvii	
<hr/>		
1	Introduction	1
1.1	Characterization and Design Verification	2
1.2	Production Testing	4
1.3	Accuracy and Correlation	5
1.4	The ATE Test Fixture	5
1.5	The Future	7
	References	7
<hr/>		
2	High-Speed Digital Basics	9
2.1	High-Speed Digital Signaling	9
2.1.1	Out-of-Band Signaling	10
2.1.2	Data Eye Diagram	11
2.1.3	Differential Signaling	12
2.1.4	Transmission Line Termination	14
2.2	Time and Frequency Domains	17
2.2.1	The Concept of Bandwidth and Its Pitfalls	18
2.3	Bit Error Rate	21
2.4	Jitter	23
2.4.1	Jitter Histogram	25
2.4.2	Jitter Categorization	26

2.4.3	Amplitude Noise and Conversion to Timing Jitter	34
2.4.4	Jitter in the Frequency Domain	36
2.5	Classification of High-Speed I/O Interfaces	39
2.6	Hardware Building Blocks and Concepts	43
2.6.1	Phase Locked Loop (PLL)	43
2.6.2	Delay Locked Loop (DLL)	46
2.6.3	Clock and Data Recovery (CDR)	46
2.6.4	Pre-Emphasis/De-Emphasis and Equalization	49
	References	53
3	High-Speed Interface Standards	55
3.1	PCI Express	56
3.1.1	Application Areas	56
3.1.2	PCI Express Fundamentals	56
3.1.3	PCI Express Details	59
3.1.4	PCI Express Protocol	61
3.1.5	Electrical Specifications	65
3.1.6	ATE Test Requirements	68
3.1.7	Test Support	70
3.1.8	Test Challenges	71
3.2	HyperTransport	73
3.2.1	Application Areas	73
3.2.2	HyperTransport Protocol	74
3.2.3	Electrical Specifications	83
3.2.4	Test Support	85
3.2.5	Test Requirements	85
3.2.6	Test Challenges	91
3.3	XDR DRAM	93
3.3.1	Application Areas	93
3.3.2	XDR Fundamentals	93
3.3.3	XDR DRAM Details	95
3.3.4	XDR Protocol	99
3.3.5	Electrical Specifications	105
3.3.6	ATE Test Requirements	106
3.3.7	Test Support	106
3.3.8	Test Challenges	106
3.4	GDDR SDRAM	107
3.4.1	Application Areas	107

3.4.2	GDDR Fundamentals	107
3.4.3	GDDR5 Details	108
3.4.4	GDDR5 Protocol	114
3.4.5	Electrical Specifications	122
3.4.6	ATE Test Requirements	123
3.4.7	Test Support	124
3.4.8	Test Challenges	124
3.5	Other High-Speed Digital Interface Standards	127
	References	129
4	ATE Instrumentation for Digital Applications	133
4.1	Digital Pin Electronics ATE Card	137
4.1.1	CDR and Phase Tracking	139
4.1.2	Equalization	140
4.1.3	Time Interval Analyzer or Time Stamper	140
4.1.4	Timing Jitter Injection	141
4.1.5	Amplitude Noise and Common Mode Voltage Injection	143
4.1.6	Bidirectional and Simultaneous Bidirectional Support	144
4.1.7	Protocol Engine	146
4.1.8	ATE Loopback Path	146
4.1.9	Parametric Measurements	146
4.2	Sampler/Digitizer ATE Card	150
4.2.1	Aliasing	150
4.2.2	Digitizer	151
4.2.3	Sampler	152
4.3	Parametric Measurements with Sampled Data	153
4.3.1	Undersampling of High-Speed I/O Signals	153
4.3.2	Coherency Equation	155
4.3.3	Capturing Digital Waveforms	156
4.3.4	Special Considerations for Coherent Sampling with Digital ATE Channels	159
4.4	Power Supplies	160
	References	162
5	Tests and Measurements	163
5.1	Bit and Pattern Alignment	163
5.1.1	Bit Alignment	165
5.1.2	Pattern Alignment	168

5.2	Functional Test	170
5.3	Shmoo Tests	172
5.4	Fundamental Driver Tests	175
5.4.1	Rise/Fall Time	175
5.4.2	Data Eye Diagram	176
5.4.3	BER Bathtub Curve	185
5.4.4	Skew	188
5.4.5	Pre-Emphasis and De-Emphasis Measurement	191
5.5	Driver Jitter Tests	192
5.5.1	Jitter Histogram	192
5.5.2	RMS Jitter	193
5.5.3	Peak-to-Peak Jitter	194
5.5.4	Measuring the Jitter Spectrum	195
5.5.5	Random and Deterministic Jitter Separation	197
5.5.6	Measuring the Data Dependent Jitter	204
5.5.7	Jitter Measurement Correlation	205
5.5.8	Driver Amplitude Noise	208
5.6	Fundamental Receiver Tests	211
5.6.1	Setup and Hold	211
5.6.2	Receiver Sensitivity	212
5.7	Receiver Jitter Tolerance	214
5.7.1	Random Jitter Tolerance	215
5.7.2	Sinusoidal Jitter Tolerance	216
5.7.3	DDJ Jitter Tolerance	218
5.7.4	Testing the Receiver Equalizer	220
5.8	PLL Characterization	221
5.8.1	Jitter Transfer	221
5.8.2	Frequency Offset	223
5.8.3	Spread Spectrum Clocking	224
5.9	Other Tests	227
5.9.1	Impedance Tests	227
5.9.2	Return Loss	231
5.10	Measurement Errors	233
	References	234

6	Production Testing	237
6.1	Golden Device	238
6.2	System Level Test	239
6.3	Instrument-Based Testing: At-Speed ATE	239
6.3.1	Physical Implementation	240
6.3.2	Parametric Testing	242
6.4	Instrument-Based Testing: Low-Speed ATE	246
6.4.1	Double Data Clocking	246
6.4.2	Channel Multiplexing	249
6.4.3	Near-End Loopback Testing	249
6.5	Instrument-Based Testing: Bench Instrumentation	261
6.6	Active Test Fixture	261
6.7	Multisite Testing	262
6.7.1	Driver Sharing for Multisite Applications	263
	References	266
7	Support Instrumentation	269
7.1	Oscilloscopes	269
7.1.1	Real-Time Oscilloscopes	269
7.1.2	Equivalent-Time Sampling Oscilloscopes	270
7.2	Bit Error Rate Tester	274
7.3	Time Interval Analyzer	275
7.4	Spectrum Analyzer	276
7.5	Vector Network Analyzer	277
7.6	Arbitrary Waveform and Function Generators	277
7.7	Noise Generators	279
7.8	Sinusoidal Clock Sources	280
7.9	Connecting Bench Instrumentation to an ATE System	282
7.9.1	Signal Integrity	282
7.9.2	Synchronization	284
7.9.3	External Reference Clock Impact on Jitter Measurements	286
7.10	Coaxial Cables and Connectors	287
7.10.1	Coaxial Cables	287

7.10.2	Coaxial Connectors	293
7.11	Accessories	298
7.11.1	Power Splitters and Power Dividers/Combiners	298
7.11.2	Attenuators, Blocking Capacitors, and Terminations	299
7.11.3	Pick-Off T	301
7.11.4	Delay Lines	302
7.11.5	Probes	302
7.11.6	Balun	306
7.11.7	Rise Time Converters	308
	References	309
8	Test Fixture Design	311
8.1	Test Fixtures	313
8.2	High-Speed Design Effects	315
8.2.1	Reflections Due to Impedance Mismatches	316
8.2.2	Conductor Losses	319
8.2.3	Dielectric Losses	320
8.2.4	Crosstalk	328
8.3	Impedance Controlled Routing	330
8.3.1	Microstrip and Striplines	330
8.3.2	Differential Routing	333
8.4	Via Transitions	334
8.4.1	Interlayer Vias	339
8.4.2	Pogo Pin Vias	340
8.5	DUT BGA Ballout	342
8.6	Sockets	346
8.6.1	Socket Electrical Characterization	347
8.7	Relays	349
8.8	Bidirectional Layout	354
8.9	Wafer Probing	356
8.10	Stack-Up	359
8.11	Power Distribution Network	363
8.11.1	Power Planes	370
8.11.2	Decoupling Capacitors	375
8.11.3	Socket Inductance	383

8.11.4	Power Distribution Network Design	384
8.11.5	Power Distribution Network Simulation	384
	References	386
9	Advanced ATE Topics	391
9.1	ATE Specifications and Calibration	391
9.1.1	Accuracy and Resolution	391
9.1.2	Understanding OTA and EPA	392
9.1.3	Linearity and Edge Placement Accuracy	393
9.1.4	Calibration	395
9.2	Multiplexing of ATE Channels	399
9.3	Focus Calibration	401
9.3.1	Skew Calibration	402
9.3.2	Data Eye Height Calibration	402
9.3.3	Jitter Injection	404
9.3.4	Data Eye Profile	406
9.4	Testing of High-Speed Bidirectional Interfaces with a Dual Transmission Line Approach	409
9.5	Including the DUT Receiver Data Recovery in Driver Tests	414
9.6	Protocol Awareness and Protocol-Based Testing	416
9.7	Testing Multilevel Interfaces with Standard Digital ATE Pin Electronics	421
9.8	Signal Path Characterization and Compensation	423
9.8.1	Signal Path Loss Compensation: De-Embedding	423
9.8.2	Characterization in the Frequency Domain	428
9.8.3	Signal Path Loss Compensation: Equalization	430
9.9	ATE DC Level Adjustments	439
9.9.1	Correction of Force Levels for DUT Input Pins	441
9.9.2	Correction of Levels for DUT Output Pins	442
	References	445
A	Introduction to the Gaussian Distribution and Analytical Computation of the BER	449
A.1	The Gaussian Distribution	450

A.2	Computation of the BER for a System with Only Gaussian Random Jitter	453
A.3	Computation of the α (BER) Value	456
A.4	Properties of the Error Function $\text{erf}(x)$ and Complementary Error Function $\text{erfc}(x)$	458
	References	459
B	The Dual Dirac Model and RJ/DJ Separation	461
B.1	The Dual Dirac Jitter Model	461
B.2	RJ/DJ Separation with the Q-Factor Algorithm	465
	References	467
C	Pseudo-Random Bit Sequences and Other Data Patterns	469
C.1	Pseudo-Random Bit Sequences	469
C.2	Pseudo-Random Word Sequences	470
C.3	Other Important Patterns	472
	References	473
D	Coding, Scrambling, Disparity, and CRC	475
D.1	Disparity	476
D.2	8B/10B Coding	478
D.3	Scrambling	481
D.4	Error Detection	484
D.4.1	Parity Bits	485
D.4.2	Checksums	485
	References	488
E	Time Domain Reflectometry and Time Domain Transmission (TDR/TDT)	491
E.1	TDR	492
E.1.1	Measuring the Impedance of a Trace with a TDR	493
E.1.2	Measuring the Round-Trip Delay of a Signal Trace	494
E.1.3	Measuring Discontinuities on a Signal Path with a TDR	495

E.1.4	Measuring the Return Loss with a TDR	495
E.2	TDT	497
E.2.1	Measuring the Step Response	497
E.2.2	Measuring the Insertion Loss with a TDT	498
E.2.3	Measuring Crosstalk Using a TDT and an Extra Sampler	498
E.3	Differential TDR/TDT Measurements	499
	References	501
F	S-Parameters	503
F.1	Simulating and Synthesizing Time-Domain Responses from S-Parameters	509
F.2	S-Parameters of Coupled Differential Pairs and Structures	511
	References	513
G	Engineering CAD Tools	515
G.1	Circuit Simulators	515
G.2	3D EM Field Solvers	518
G.3	2D Planar Field Solvers	518
G.4	Power Integrity	520
G.5	Model Generation	521
G.6	Other Tools	521
	References	524
H	Test Fixture Evaluation and Characterization	525
H.1	Measuring the Test Fixture Performance	525
H.1.1	Test Coupons	527
H.1.2	Test Fixture Socket and Socket Via Field Probing	529
H.2	Measuring the Test Fixture Power Distribution Network	535
	References	540
I	Jitter Injection Calibration	543
I.1	Sinusoidal Jitter Injection Calibration	543
I.1.1	The J_1/J_0 Bessel Approach	544
I.1.2	The RJ Subtraction Approach	548

I.2	Random Jitter Injection Calibration	551
I.3	ISI Jitter Injection Calibration	555
	References	557
	About the Authors	559
	Index	561

Preface

The starting point for writing this book was a discussion between the two of us about our common experience in our ATE test application support work for high-speed digital interfaces and the challenges these interfaces pose on the test engineers we supported. Initially we just planned to write an extended application note about the nature of jitter and jitter testing. This was due to the fact that we identified this topic as the one that all engineers that in the one or other way have responsibility for getting high-quality high-speed devices out of a fabrication line had most questions about.

However, it became clear to us very soon that just another jitter document would not really satisfy the needs of these engineers because their main difficulty is not necessarily to understand the test theory around jitter. It rather is the transfer of this knowledge about the theory into a practical implementation of device test flows and the various cross dependencies between device building blocks, interface architectures, test system interfacing, integration of external instrumentation, and efficient test implementations for production testing.

Thus, we analyzed the real-world test implementation issues we came across in the past years, from the instable test that resulted in production yield loss because of insufficient device training, over the massive device failures that were caused because just a single pulse within millions of clock cycles on a PLL input was missing in the test pattern, to the marginal data eye test due to suboptimal test fixture design.

As a result of this analysis, we saw the need to provide a comprehensive introduction to the characterization and production testing of high-speed I/O digital interfaces using automated test equipment (ATE) that does not just list the test requirements for these interfaces and propose test implementations but also gives some background on why the test requirements are defined the way

they are.

The target audience of this book are mainly test engineers working on design verification, characterization, and production testing of multigigabit I/O interfaces. However, the book provides a general overview over the challenges these test engineers face in their daily work with high-speed devices that can be helpful also for chip designers and product engineers to minimize these challenges using appropriate DfT and device test strategies to ensure sufficient device quality with less test implementation effort.

After introducing the challenges of high-speed digital testing on ATE in Chapter 1, we provide a general introduction to high-speed digital signals and related topics in Chapter 2 to set the basic framework to understand the following chapters.

The underlying basis for this book is the high-speed I/O digital standards that evolved in the past years. In Chapter 3, we discuss some of these standards. We selected the standards that are covered with regard to the general test requirements and challenges they represent for ATE implementations. The chosen standards reflect a cross-section of the current high-speed I/O interfaces in the market with respect to test requirements. In this chapter we do not want to give a very detailed description and test implementation examples for the dedicated interface. The main purpose is to extract the general architectural characteristics for each interface that define the type of tests that need to be implemented and can be leveraged to other interfaces that use similar architectural implementations.

After this, we present the basics of ATE systems and instrumentation in Chapter 4 with emphasis on the critical instrumentation for high-speed digital applications like, for example, digital pin electronics cards, sampler/digitizer cards, and power supplies. One of the core chapters of the book is Chapter 5 that covers the details of key measurements for high-speed I/O interfaces and the options a test engineer has to implement these measurements on an ATE system.

Chapter 6 focuses on how high-speed I/O interfaces can be tested efficiently in a production test environment. The key discussion in this chapter deals with at-speed loopback testing. Characterization and test of high-speed digital I/Os often requires complementing an ATE with external measurement instruments for some dedicated tests. Chapter 7 presents some of the bench instrumentation that is relevant for engineers working on high-speed digital applications on ATE and also some of the related accessories.

One of the topics that commonly is recognized as a key bottleneck for the ability to test and characterize high-speed digital interfaces on an ATE is the electrical interfacing between the ATE pin electronics and the DUT. Chapter 8 covers this bottleneck and discusses ATE test fixture design

including associated topics like sockets and wafer probing.

Last but not least, we discuss some advanced topics on high-speed digital testing with ATE in Chapter 9. These topics include the definition of EPA/OTA, the importance of linearity, focus calibration, and the important topics of de-embedding and test fixture loss compensation through equalization.

In the appendices of the book, we cover several additional topics that are not directly related to the test of high-speed devices itself. However, these topics are helpful to understand the background of some of the test requirements and test implementations that are presented in the main chapters of the book.

Appendix A provides a brief overview of the Gaussian distribution and some examples of obtaining analytical results that are relevant for high-speed digital applications.

Appendix B presents a detailed description of the dual Dirac jitter model and the associated random and deterministic jitter separation algorithm.

Appendix C presents an overview of pseudo-random bit sequences and other important data patterns.

Appendix D discusses the topics of encoding, scrambling, disparity, and error correction that are required to understand the implementation of some of the high-speed digital standards.

Appendix E provides an introduction to the topic of time domain reflectometry and transmission that is important for test fixture characterization.

Appendix F introduces the concept of S-parameters that are again important for test fixture characterization and modeling.

Appendix G provides an overview of the different simulation tools that are relevant for high-speed digital applications, especially for test fixture design.

Appendix H presents several approaches to characterize a manufactured test fixture electrical performance, not only regarding the high-speed signal paths but also the power distribution network.

Appendix I discusses the challenge of calibrating the amount of jitter injected by an ATE driver into the device under test (DUT) since receiver jitter tolerance is a key part in some of the high-speed digital standards.

Acknowledgments

We first thank Heidi Barnes, Hideo Okawara, and Bernhard Roth from Verigy and Michael Comai from AMD for contributing to the sections on test fixture design, sampler/digitizer ATE card, data eye profile, active equalization, and

HyperTransport. We also thank Verigy for allowing us to use several pictures we have obtained in the course of our work.

We thank Gert Haensel from Texas Instruments, Wolfgang Maichen from Teradyne, Christoph Zender from DHBW Horb, Istvan Novak from Sun Microsystems, Eric Bogatin from Bogatin Enterprises, Marcus Mueller from Agilent Technologies, and Takahiro J. Yamaguchi from Advantest Laboratories, Ltd., for reviewing the book or parts of it. We also thank all the colleagues from Verigy that have contributed to the review of this book, especially Shinji Fujita, Frank Hensel, Joe Kelly, Bernd Laquai, Clemens Leichtle, Jinlei Liu, Kosuke Miyao, Joerg-Walter Mohr, Jan-Ole Brandt, Roger Nettles, Claus Pfander, Robert Schneider, and Juergen Wolf. Thanks to Andreas Olenyi and Oliver Guhl for their support inside Verigy.

Orlando Bell from GigaTest Labs, and Bob Schaefer and Mike Resso from Agilent Technologies provided help in some of the measurements presented in this book.

The author José Moreira thanks his university mentor Professor Carlos Bispo from the Institute for Systems and Robotics in Lisbon for instilling in him the enjoyment of technical writing.

Finally we thank all the other colleagues that we had the privilege to interact with and who helped shape the authors' knowledge of high-speed digital testing with ATE.

If you have any questions or feedback to the authors please direct them to j.moreira@computer.org or hubert.wermann@arcor.de.

1

Introduction

It is an amazing fact of modern technology that we are able to manufacture integrated circuits (IC) with hundreds of millions of transistors switching at gigahertz clock rates and containing dozens of high-speed I/O terminals running at multigigabit data rates.

It even is more amazing that we are able to guarantee the correct behavior of such devices when delivered to the consumer.

The most amazing matter of fact, however, is that such devices can be delivered not only ensuring the device operation by itself, but their compliance with complex high-speed I/O specifications which guarantee error-free communication with other chips and allow the implementation of powerful yet highly integrated electronic systems we all enjoy (more or less) in our daily life.

These high-speed I/O standards that guarantee the interoperability of devices from different manufacturers and help to facilitate the widespread adoption of multigigabit I/O interfaces into applications such as wired communications, computation, and consumer electronics typically set directions in the industry and serve as a roadmap that pushes IC manufacturers to higher data rates. Figure 1.1 shows the data rates for some of the more common high-speed digital interfaces listing both what is available and what is under development as of the writing of this book.

One of the most critical steps in the manufacturing process of devices that employ such complex high-speed I/O standards is IC testing. While device testing does not usually get the same recognition as the design and front-end process disciplines of the semiconductor industry, IC testing is becoming more important and more complex with the continuous movement to enhanced process technologies and higher clock frequencies or data rates. Besides the technical necessity to guarantee correct device interoperation, the

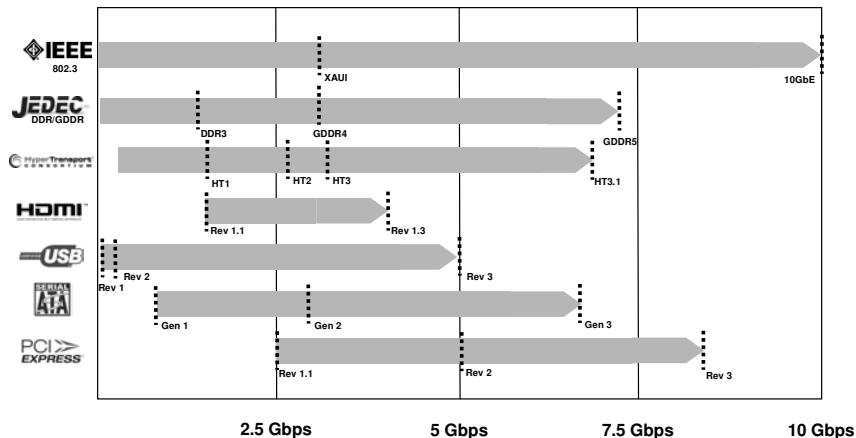


Figure 1.1 Some high-speed digital standards available and under development as of the writing of this book.

implications of IC testing on cost also have to be stressed. Very often device manufacturers only consider the liabilities that are associated with IC testing like capital investment and recurring device manufacturing cost but they neglect the asset that a thorough and reliable test implementation represents because it avoids the delivery of faulty parts to a consumer. It is important to acknowledge that the subject of IC testing is vast and spans areas in multiple dimensions, from IC design to the final application integration into a system. This is reflected in the test-related topics that have to be considered along the manufacturing life cycle of a device. Examples of these topics include design for test (Dft) strategies, design validation and characterization, production testing, system level test, and test cell integration.

The objective of this book is to cover the specific topic of testing high-speed digital interfaces with automated test equipment (ATE). When discussing the testing of high-speed digital interfaces with automated test equipment, there are four main areas a test engineer has to address. These are characterization and design validation of the initial silicon, production testing, accuracy and correlation between bench instrumentation and ATE or between characterization and production, and finally the test fixture used for the high-speed digital testing of the device under test (DUT).

1.1 Characterization and Design Verification

The high-speed digital interfaces embedded in multifunctional integrated circuits have achieved a significant degree of complexity. This complexity

requires the test engineer to have a broad range of knowledge covering several disciplines such as analog design, digital design, probability and statistics, and signal integrity to be able to understand and solve the technical test challenges posed by these interfaces. Figure 1.2 shows a block diagram of a multigigabit I/O cell. Looking at the block diagram, one can estimate the complexity that state-of-the-art I/O cells contain. This complexity in combination with the data rates these cells are operating, both today and in the future, presents additional challenges for design validation, characterization, and production testing.

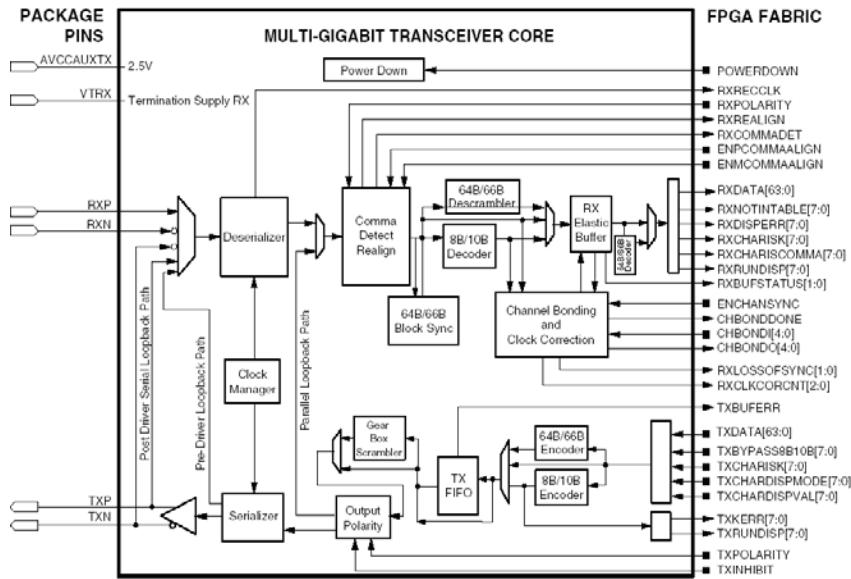


Figure 1.2 Xilinx Rocket I/O cell block diagram (reprinted with permission from [1]).

For the test engineer, the challenge of this movement to higher data rates is not as simple as just using more advanced measurement instruments or ATE systems that support the increase in data rates. To address the timing restrictions that these higher data rates bring, the complexity of modern standards is increasing. State-of-the-art high-speed I/O standards now require additional complex measurements for the test of the I/O cell such as jitter generation, and jitter tolerance. These new challenges require the test engineer to understand and implement a new set of measurements that were not required in the past. Figure 1.3 shows the receiver requirements from the PCI Express standard [2].

Parameter	Description	Min	Max	Units
UI	Unit interval without including of SSC	199.94	200.06	ps
$T_{RX-HF-RMS}$	1.5 – 100 MHz RMS jitter		3.4	ps RMS
$T_{RX-HF-DJ-DD}$	Max Dj impinging on Rx under test		88	ps
$T_{RX-SSC-RES}$	33 kHz Refclk residual	--	75	ps
$T_{RX-LF-RMS}$	< 1.5 MHz RMS jitter	--	4.2	ps RMS
$T_{RX-MIN-PULSE}$	Minimum single pulse applied at Rx	120		ps
$V_{RX-MIN-MAX-RATIO}$	Min/max pulse voltage ratio seen over an time interval of 2 UI.	--	5	
V_{RX-EYE}	Receive eye voltage opening	120		mVPP diff
$V_{RX-CM-CH-SRC}$	Common mode noise from Rx	--	300	mVPP

Figure 1.3 Example of part of the DUT receiver requirements from the PCI Express standard. (From: [2]. ©2009 PCI-SIG. Reprinted with permission.)

1.2 Production Testing

Production testing of high-speed digital interfaces presents a different set of challenges when compared to design validation and characterization. A high-speed digital standard might define a series of measurements that the interface must comply with, but in production testing the cost pressure associated with the target application makes it impractical to implement all of these measurements. The characterization and design verification tests ensure compliance with the standard or design requirements while the lower cost production test focuses on a minimal set of tests to verify that the process parameters for a given device have not shifted, to check basic functionality, and to screen for known failure mechanisms.

At a high level, one can assume that the cost of testing will be proportional to the test time for each device and the capital cost of the measurement instrumentation. This simplified look at cost of testing highlights the importance of faster measurement techniques, a lower cost per measurement ATE system, and the benefit of parallel or multisite testing with a single insertion on an ATE system. Implementing these cost reduction methods can require significant changes in the test methodology and the types of tests compared to those found in high-speed I/O standards. For example, it is not uncommon to reduce cost of testing by using a less expensive ATE system where the measurement instrumentation does not have the data rate or bandwidth to support the device under test. This type of situation requires test methodology such as DfT approaches and at-speed loopback to ensure the full

speed functionality of the high-speed I/O interface.

1.3 Accuracy and Correlation

The ability of an ATE system to measure the performance of a high-speed digital interface depends on the measurement accuracy/repeatability and correlation that can be obtained for a given application. This can be quite challenging with increasing data rates. Basic requirements like the timing accuracy for a state-of-the-art ATE system are now specified in the picosecond range with random jitter values in the hundreds of femtoseconds. This type of timing accuracy demands fast digital edge rates with multigigahertz bandwidth which increases the challenges when trying to maintain the same accuracies over multiple channels in an ATE system.

The narrow timing margins also create a challenge for correlation between ATE systems and bench instrumentation since slight timing variations and test fixture differences between measurement systems can have a significant impact. This correlation is complicated even further for measurements such as jitter separation where the methodology, hardware performance, and algorithms used for the measurement can vary dramatically between instrument manufacturers.

To address these challenges and the trade-offs between measurement systems, the test engineer needs to have a complete understanding of the capabilities and accuracy of the ATE system and how these parameters influence the measurements they need to apply. In addition, the engineer has to have a full understanding of the instrumentation that the ATE system must correlate to, including any specific measurement algorithms these instruments apply.

1.4 The ATE Test Fixture

The design of the hardware and signal path for multigigabit instrumentation is of little use if the signal cannot be transmitted to the device under test (DUT) with at least the performance of the ATE instrument itself [3]. The test engineer now must have an increased understanding of the impact the signal path has on the characterization of multigigabit digital I/O interfaces. In order to assure that this understanding is transferred correctly into the overall test solution, he has to be involved in the design of the test fixture also known as DUT loadboard or device interface board (DIB) which connects the ATE instrumentation to the DUT. In the past, most test engineers were only responsible for providing a schematic or netlist of the needed DUT/ATE pin assignments, leaving most of the test interface design details (e.g., trace

geometry, PCB dielectric material) to the PCB layout team. The success of the test fixture simply meant getting all of the point-to-point connections right and sometimes having additional circuitry on the test fixture to support a particular application test.

This simple pin assignment approach is no longer appropriate for test fixtures that are intended to test high-speed digital interfaces. In this case the actual signal integrity of the routing on the test fixture is critical for the success of the application, and must be engineered to meet the testing requirements of the high-speed I/O channel. This challenge is illustrated in Figure 1.4 where it is shown that the signal performance at the ATE interconnect where the system specifications are set is not the same as that found further away at the DUT. If this interface between the ATE interconnect and DUT is not engineered correctly, it can close the eye in a data eye diagram of a high-speed I/O completely. This in turn will have a direct impact on the measurement accuracy in a device characterization application or on the production yield in a manufacturing test environment.

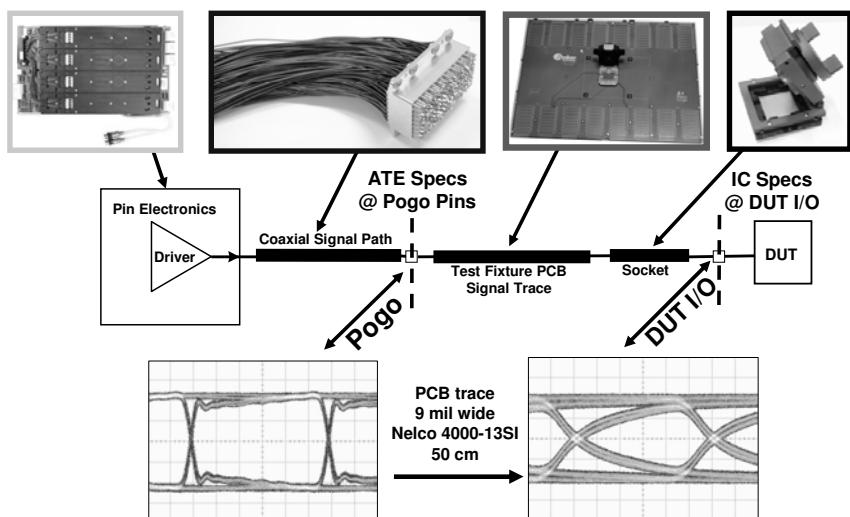


Figure 1.4 The signal integrity challenge in an ATE system. (*From: [4]. ©2010 Verigy. Reprinted with permission.*)

It is also important to take into account that the signal integrity of a test fixture is not restricted to the performance of the connection between the ATE channels and the DUT I/O cells. The generation and detection of the multi-gigabit signal requires a power distribution network that has low noise and fast switching current response characteristics to prevent voltage drop on the supply lines.

1.5 The Future

One indication from looking into the past is that pin counts and data rates will continue to increase in the future pushing the limits of existing electrical interconnects. Technologists have often predicted the need to move to optical technology for short range applications and the demise of the copper interconnect. Not long ago, 2.5 Gbps was considered the limit for copper-based electrical interconnects and that the next generation of chip-to-chip interconnect needs to use optical signal transmission [5]. Although copper-based electrical interconnects have fundamental limits [6], the fact that the majority of the electronics industry is copper-based makes staying with copper interconnects a significant advantage. Because of that, the electronics industry will continue to solve the challenges associated with higher data rates by using new technologies and smarter approaches not only in the design of the IC I/O cells but also in the data transmission.

The 40-Gbps data rates over copper are already possible [7] and devices with above 100-Gbps data rates have been shown in research literature [8]. One can expect that with the continuous need for higher bandwidth in data communication and computation, it is just a matter of time for a new generation of higher data rates to arrive. The ATE industry will have to follow this trend and also continue to provide higher data rates [9]. This means that all of the challenges presented in this book will increase in the near future and new test and measurement techniques and equipment will evolve to meet those requirements.

References

- [1] J. Moreira, B. Sadler, and D. Ferguson, “Parametric Testing of a 10Gbs I/O Cell in Production Through a Parametric Loopback Approach with Retiming,” *IEEE International Design and Test Workshop*, 2006.
- [2] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [3] J. Moreira, M. Tsai, J. Kenton, H. Barnes, and D. Faller, “PCB Loadboard Design Challenges for Multi-Gigabit Devices in Automated Test Applications,” *DesignCon*, 2006.
- [4] B. Roth and J. Moreira, “Equalization in the HSM6800 Pin-Electronics,” *Verigy VOICE Users Conference*, 2010.
- [5] N. Savage, “Linking with Light,” *IEEE Spectrum*, Aug. 2002.
- [6] D. A. B. Miller and H. M. Özaktas, “Limits the Bit-Rate Capacity of Electrical Interconnects from the Aspect Ratio of the System Architecture,” *Journal of Parallel and Distributed Computing*, no. 41, 1997.

- [7] E. Bogatin, "Has the Copper Interconnect Hit Its Speed Limit? Data Rates are Reaching the Unlikely Speed of 40 Gbps, yet Copper Runs on," *Printed Circuit Design and Manufacture*, Jan. 2004.
- [8] Y. Suzuki, M. Mamada, and Z. Yamazaki, "Over-100-Gb/s 1:2 Demultiplexer Based on InP HBT Technology," *IEEE Journal of Solid State Circuits*, vol. 42, Nov. 2007.
- [9] J. Moreira, H. Barnes, H. Kaga, M. Comai, B. Roth, and M. Culver, "Beyond 10Gb/s? Challenges of Characterizing Future I/O Interfaces with Automated Test Equipment," *IEEE International Test Conference*, 2008.

2

High-Speed Digital Basics

This chapter reviews some important concepts that are necessary when analyzing high-speed digital signals and interfaces. These concepts provide a basis for the following chapters. General references on this topic for the interested reader are [1–3].

2.1 High-Speed Digital Signaling

The inputs/outputs of a high-speed digital interface are electrical waveforms (ignoring optical interfaces) that represent information encoded in a digital signal. Several options to encode a digital signal electrically exist as shown in Figure 2.1. Most high-speed digital standards use a nonreturn to zero (NRZ) approach for the signal being transmitted. Return to zero (RZ) signaling is usually not used on high-speed digital interfaces due to its bandwidth requirements. Another type of signaling is pulse amplitude modulation (PAM). In this approach the signal to be transmitted is encoded with different voltage levels, thus, for example, a four level PAM interface can transmit the same information as a NRZ single level interface at half of the clock frequency, as shown in Figure 2.1. Although PAM and RZ signaling are important for certain types of applications, in the remainder of this chapter we will only discuss NRZ type signaling.

Figure 2.2 shows one example of analyzing a digital interface with a commercial ATE system. The figure shows the programmed time domain waveforms for the driver channels on the ATE system that stimulates the respective receiver on the DUT and the digital waveforms measured by the ATE receiver from the appropriate DUT drivers. From these received waveforms it is possible to infer the logic value based on the expected logic voltage thresholds.

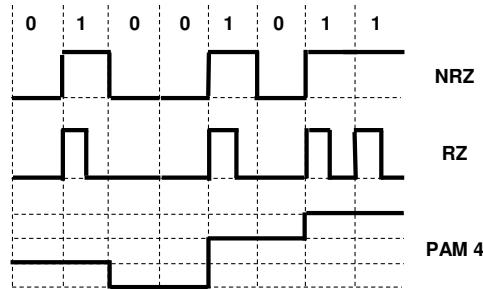


Figure 2.1 Comparison of the different types of signaling approaches for high-speed digital interfaces.

Unfortunately this view is no longer sufficient for characterizing and debugging modern I/O digital interfaces [4]. The increase of the data rate on the DUT I/O data makes the timing and level margins more critical. The question is no longer only whether there are any errors on the data transmission through the digital interface but also how many bits can be transmitted on average before the first error occurs. The reason is that in multigigabit interfaces it is no longer possible to guarantee an error-free data transmission for an indefinite amount of time. This fact requires the use of new techniques to analyze the performance of digital interfaces like the data eye diagram, BER bathtub curve, and jitter analysis, which are discussed in this and the following chapters.

2.1.1 Out-of-Band Signaling

Traditional electrical interfaces typically consist of signals that take care of the data transmission itself (e.g., address and data signals) and signals that are used for control and status information (e.g., reset signal, interrupt signals). These control and status signals are referred to as sideband signals. Besides increased data rates, high-speed I/O interfaces also tend to reduce the amount of sideband signals and integrate the status and control information that needs to be conveyed to partner devices into the regular communication protocol that runs on the signals which are also used for payload data exchange. The most extreme example of this are high-speed I/O interfaces that use serial links. These typically eliminate sideband signals completely to get maximum benefit from the serialization and an optimal ratio of signals that transfer payload data and signals that transfer sideband information.

However, for some of the control and status information that needs to be transferred, integration into the standard protocol that is applied for data exchange is not possible. This, for example, might be the case for serial links

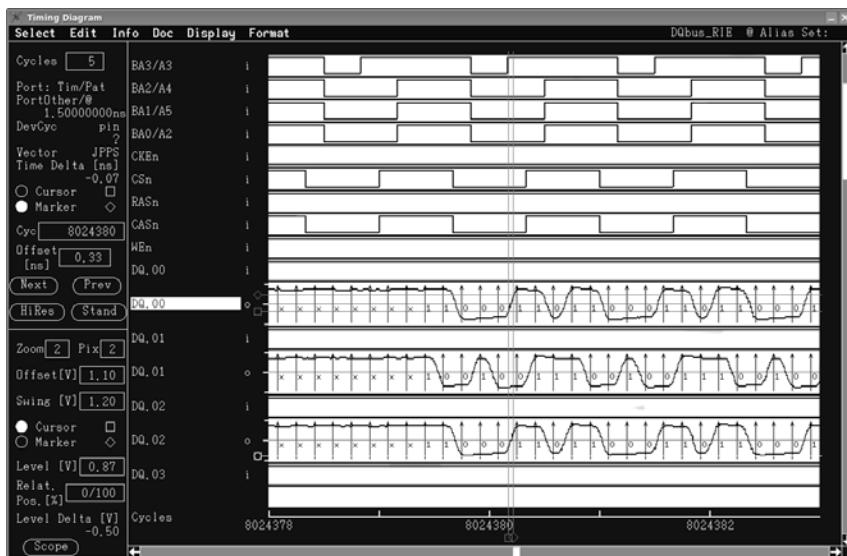


Figure 2.2 Classical view of a digital interface obtained with an ATE system (courtesy of Verigy).

that are in power-saving modes, but still need to transmit few sideband data, or during link (re)initialization. Often signaling mechanisms that do not match the signaling used for normal data exchange of the respective high-speed I/O standard are applied to convey this information in such a case. These non-standard signaling mechanisms are called out-of-band signaling. It has to be stressed that out-of-band signaling can be implemented very differently between the various high-speed I/O interfaces.

2.1.2 Data Eye Diagram

The data eye diagram is a methodology to display and characterize a high-speed digital signal in the time domain. The data eye diagram is derived from the digital waveform by folding the parts of the waveform corresponding to each individual bit into a single graph with a one unit interval (UI)¹ width on the timing axis as shown in Figure 2.3 [2].

Contrary to the classical digital waveform representation of Figure 2.2, the data eye diagram does not allow the identification of the individual bits in the bit pattern. However, showing all the bits folded into a single UI allows the easy visualization of the quality of the digital signal. This is because both the

¹One unit interval (UI) corresponds to the period of one bit and is sometimes also referred to as the bit period. For example, in a 10-Gbps data signal the UI is 100 ps.

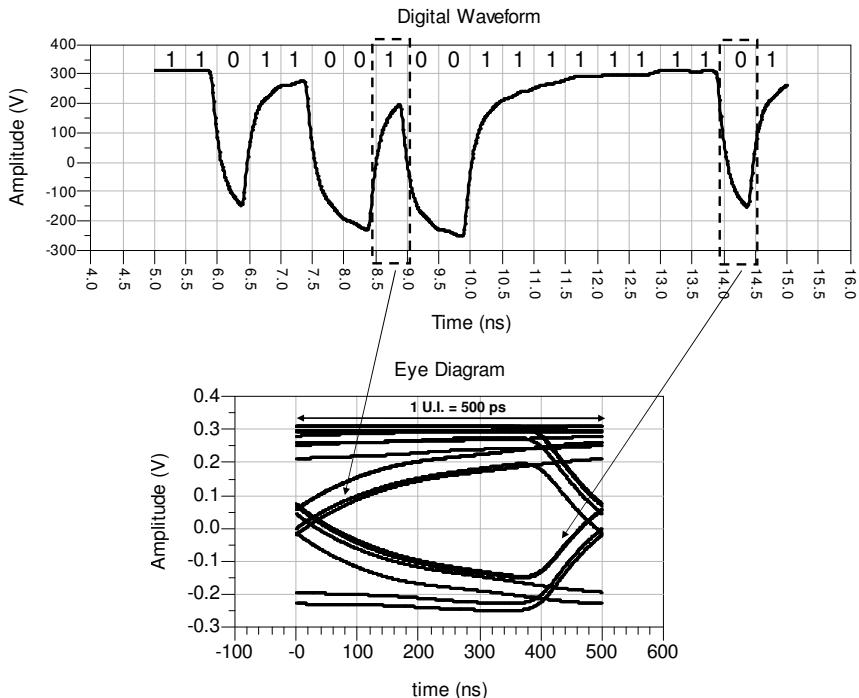


Figure 2.3 Construction of the data eye diagram from a digital waveform.

best and the worst case bit transitions on the waveform will be folded together as shown in Figure 2.4. In the figure two different waveforms are displayed, and although it is not easy to compare the performance of both digital signals when looking at the full waveforms, using the data eye diagram it is obvious that waveform 2 is worse than waveform 1.

Figure 2.5 shows some of the typical nomenclature associated with an eye diagram. One key item is the optimal strobing point in the data eye diagram. This is the point with the maximum timing and level margin. Another important item is the amount of timing jitter (Section 2.4) and amplitude noise (Section 2.4.3) that are present on the data eye that reduce the timing and level margin.

2.1.3 Differential Signaling

Most high-speed digital interfaces use differential signaling for electrical transmission. Differential signaling uses two complementary output drivers (differential legs) to drive two independent signal transmission lines [5]. Figure 2.6 shows a comparison between single-ended and differential

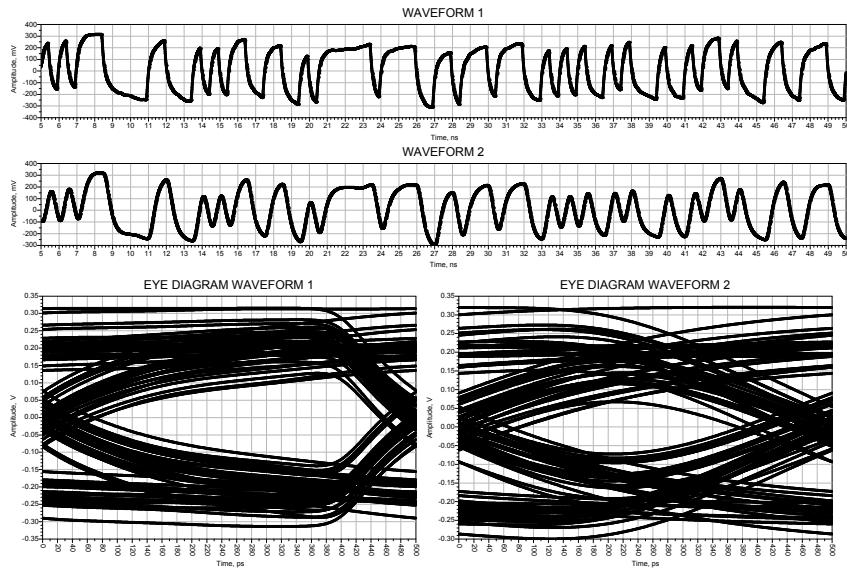


Figure 2.4 Comparing two waveforms on the time domain showing the advantage of a data eye diagram display.

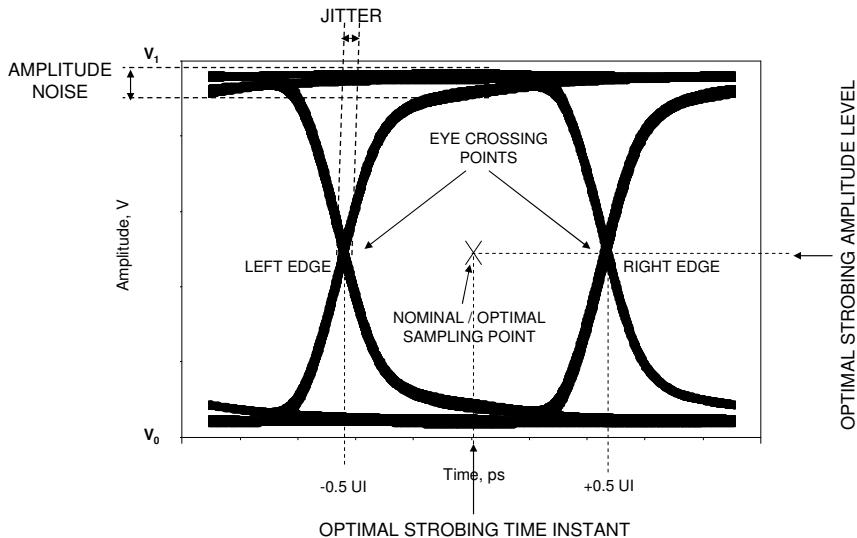


Figure 2.5 Nomenclature typically associated with a data eye diagram.

signaling and how differential signaling is defined. While a single-ended signal is characterized by its high voltage level value V_{HIGH} and its low voltage level value V_{LOW} , a differential signal is usually characterized by its differential amplitude V_{DIF} and the common mode value V_{CM} of the two differential legs.

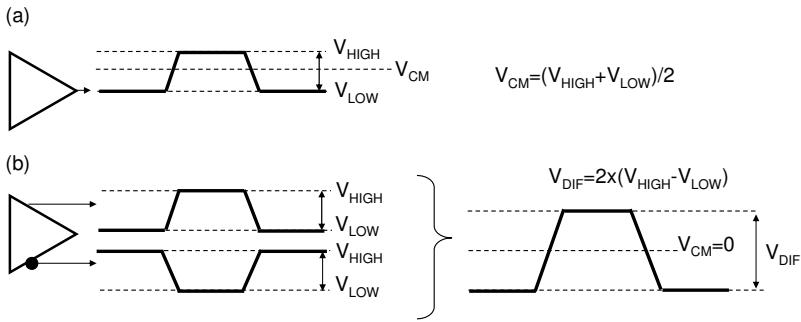


Figure 2.6 Comparison of (a) single-ended and (b) differential signaling.

Differential signaling has several properties that make its use advantageous for high-speed digital interfaces [5]. They include lower dI/dt, lower EMI, higher gain, and more immunity to crosstalk if transmitted through a coupled transmission line. Of course, the drawback is the need to use twice the number of pins and transmission lines compared to a single-ended signaling approach.

2.1.4 Transmission Line Termination

Electrical signals in the context of high-speed digital applications are typically transmitted through coaxial cables or controlled impedance printed circuit board traces with a characteristic impedance of $50\ \Omega$ for single-ended signaling and $100\ \Omega$ for differential signaling as shown in Figure 2.7 ([5, 6] are good starting points for the concepts of impedance and controlled impedance design).

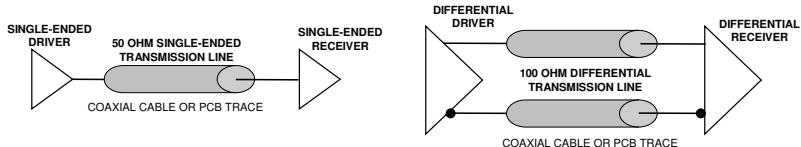


Figure 2.7 $50\ \Omega$ single-ended and $100\ \Omega$ differential transmission system.

The challenge of transmitting high-speed digital signals with fast rise times is that any impedance change seen by the signal while it travels through

the signal path will create a reflection (e.g., due to relays and vias). Some of these points will be discussed in more detail in Chapter 8. In this section we are interested in the impedance discontinuity from a DUT driver/receiver or pin electronics driver/receiver. This is illustrated in Figure 2.8.

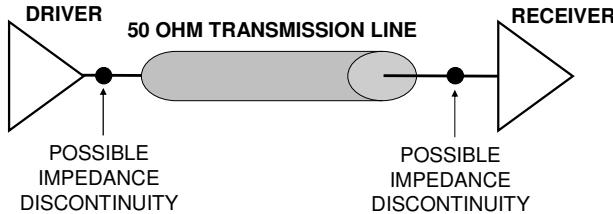


Figure 2.8 Impedance discontinuities at the pin electronics and DUT receiver.

Each time an electrical signal “sees” an impedance discontinuity, a portion of the electrical signal is reflected. The value of this reflected portion can be computed by (2.1) where Z_0 is the transmission line impedance before the discontinuity and Z_1 is the impedance of the transmission line right after the discontinuity that is encountered by the incident wave V_I and results in a reflected wave V_R :

$$V_R = \frac{Z_1 - Z_0}{Z_0 + Z_1} V_I \quad (2.1)$$

This equation shows that to prevent any reflections at the driver or receiver as shown in Figure 2.8, it is necessary that the impedance of the driver and receiver is equal to the impedance of the transmission line. This can be achieved by the use of proper termination resistors as shown in Figure 2.9 for the case of a single-ended 50 Ω transmission line.

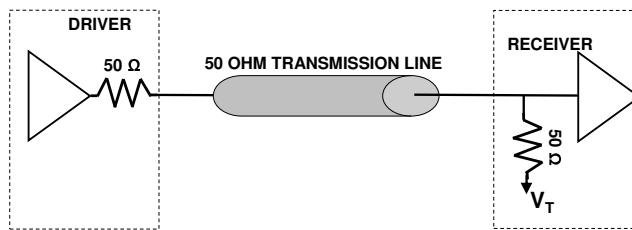


Figure 2.9 Properly terminated transmission line.

Several methods for transmission line termination exist. In this section we will concentrate on the most common structure for high-speed digital applications which is shown in Figure 2.9 and is named series termination

(for the ATE driver side) and parallel termination (for the DUT receiver side). Other possible termination methods are discussed in [6].

It is important to note that the 50Ω resistor on the receiver parallel termination is usually connected to a termination voltage source. This is important to minimize any constant current flowing between the driver and receiver as shown in Figure 2.10. This is one of the values that typically needs to be set when programming the ATE pin electronics receiver.

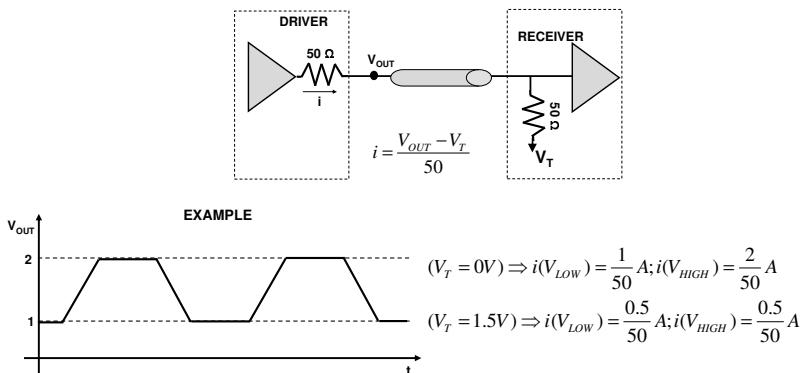


Figure 2.10 The importance of the 50Ω voltage termination on a receiver and its impact on the current flowing between the driver and receiver.

There are some cases where the measurement instrument has a fixed 50Ω to ground parallel termination that cannot be changed. Section 7.11.1 shows a possible approach to address this case when a termination voltage is required by the application. With differential signaling typically two types of termination schemes, named “center tap” and “cross termination” are used on the receiver as shown in Figure 2.11. From an electrical point of view there is no difference between the two configurations, although some standards or receiver implementations might prefer one of them.

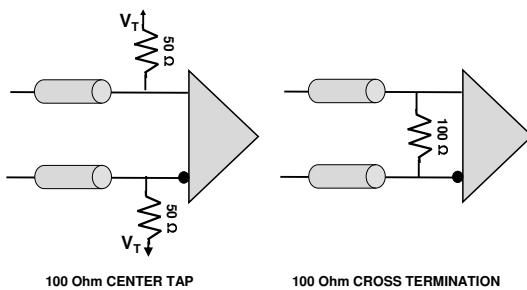


Figure 2.11 Differential signaling receiver termination schemes.

2.2 Time and Frequency Domains

Digital signals can be analyzed in both the time and frequency domains. Digital signals are characterized by the fact that they ideally have a rectangular shape as shown in Figure 2.12 (top). For a random binary sequence with bit period T_B it is possible to compute its power spectrum as shown in (2.2) [7, 8]. It results in the *sinc* function shape which is defined in (2.3).

$$\text{Power Spectrum}(f) = \log \left[T_B \left(\frac{\sin(\pi f T_B)}{\pi f T_B} \right)^2 \right] \quad (2.2)$$

$$\text{sinc}(x) = \begin{cases} 1 & (x = 0) \\ \frac{\sin(x)}{x} & (\text{otherwise}) \end{cases} \quad (2.3)$$

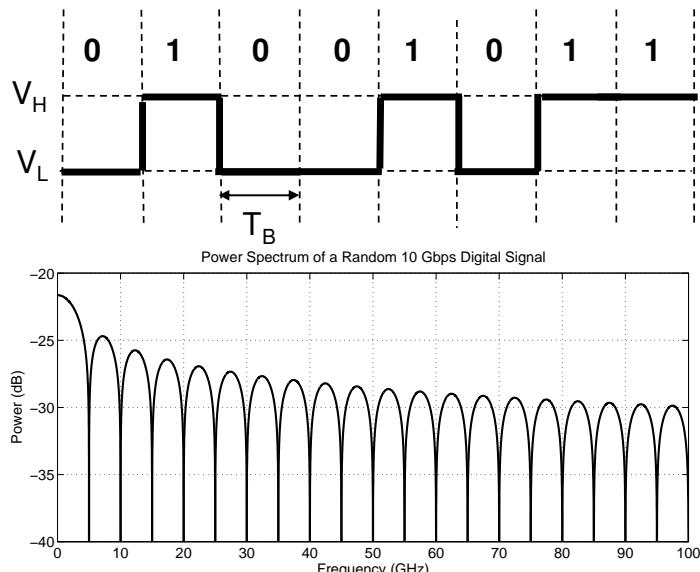


Figure 2.12 An ideal NRZ digital signal (top) and the power spectrum of a perfect NRZ random digital signal at 10 Gbps (bottom).

Figure 2.12 (bottom) shows the computed power spectrum for a 10 Gbps random binary sequence. Note that a null in the power spectrum exists at each multiple of the frequency corresponding to half of the bit period ($T_B/2$). Although Figure 2.12 shows the time and frequency domain views for a “perfect” digital signal, real signals are not perfect. Figure 2.13 shows a real 10 Gbps signal with a PRBS $2^7 - 1$ data pattern (see Appendix C) measured

on the time domain with an equivalent-time sampling oscilloscope and in the frequency domain with a spectrum analyzer (see Chapter 7).

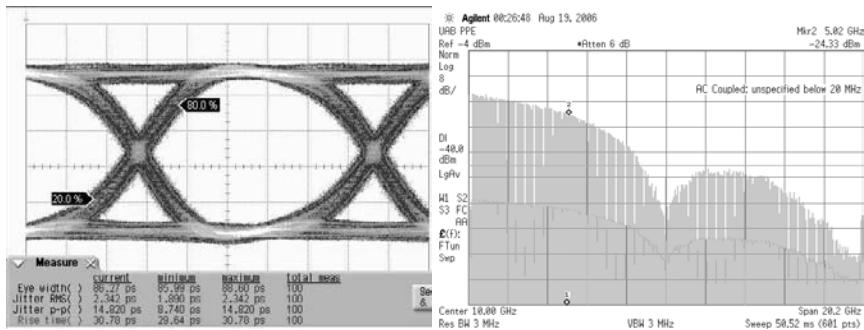


Figure 2.13 Viewing a high-speed digital signal on the time domain (left) and frequency domain (right).

The time domain measurement is in the form of a data eye diagram showing a signal with a fast rise time (30 ps 20/80² with a good “open eye”). The frequency domain measurement shows one important property of high-speed digital signals, which is the fact that they are broadband, having energy from a low frequency determined by the used data pattern until the highest frequency that is determined by the signal rise time. This is a very important factor that makes high-speed digital signals different from high-frequency RF signals that are typically narrow band around a specific frequency point.

The frequency content of the digital signal will depend on the data rate, on the pattern that is used and on the rise time of the driver. Figure 2.14 shows the frequency spectrum of two data signals with the same data rate and data pattern but originating from different drivers. The figure shows that the driver with the slowest rise time (in this case the one with a maximum data rate of 3 Gbps) has almost no energy above 10 GHz while the 10-Gbps driver still has energy components beyond 10 GHz.

2.2.1 The Concept of Bandwidth and Its Pitfalls

The concept of bandwidth for a measurement instrument or for a passive interconnect like a cable or a test fixture is very often used to define the required or expected performance of a measurement instrument or interconnect. Typically the term “bandwidth” refers to the frequency, where

²20/80 refers to the voltage levels used to compute the signal rise time meaning that the 20% and 80% point of the full signal amplitude are used to compute the rise time. 10/90 is also sometimes used for rise time measurements (see Section 5.4.1).

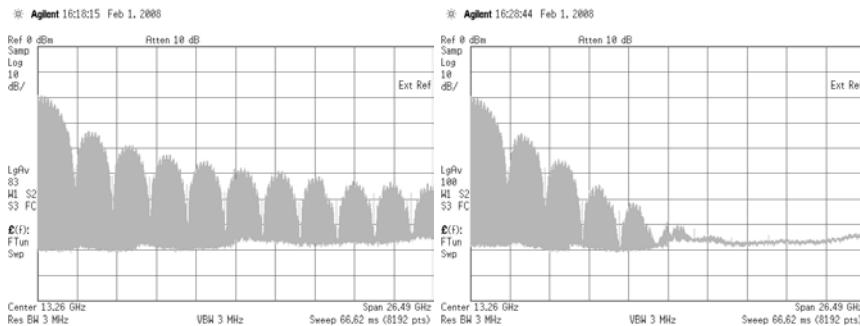


Figure 2.14 Comparison of the measured frequency spectrum of a PRBS $2^7 - 1$ data signal at 2.5 Gbps generated by two different drivers with different rise times (left: a 10-Gbps driver; right: a 3-Gbps driver).

the frequency response of the measurement instrument or interconnect is -3 dB³ below that of the DC point as shown in Figure 2.15. Note that for coaxial cables the term “bandwidth” is sometimes used with another definition in mind. This topic will be further discussed in Section 7.10.1.

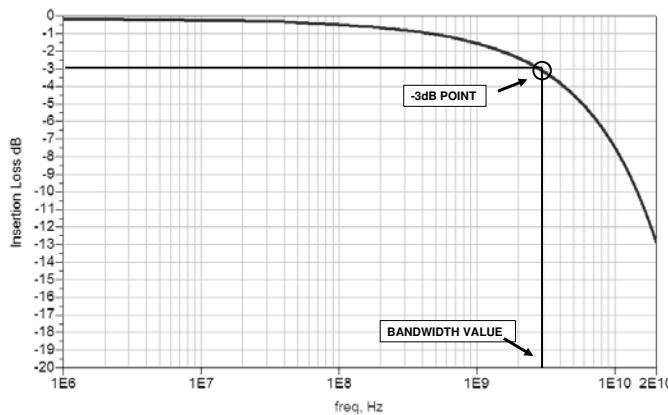


Figure 2.15 Example of the response of a measurement instrument or interconnect and the -3 dB definition of bandwidth.

The problem with this definition is that it is very limited in the sense that systems with the same bandwidth value might have very different frequency responses. Figure 2.16 shows the frequency response of two different systems with the same bandwidth and the simulated data eye diagram for each of the systems.

³ -3 dB corresponds to a reduction of the signal amplitude by a factor of 1.41.

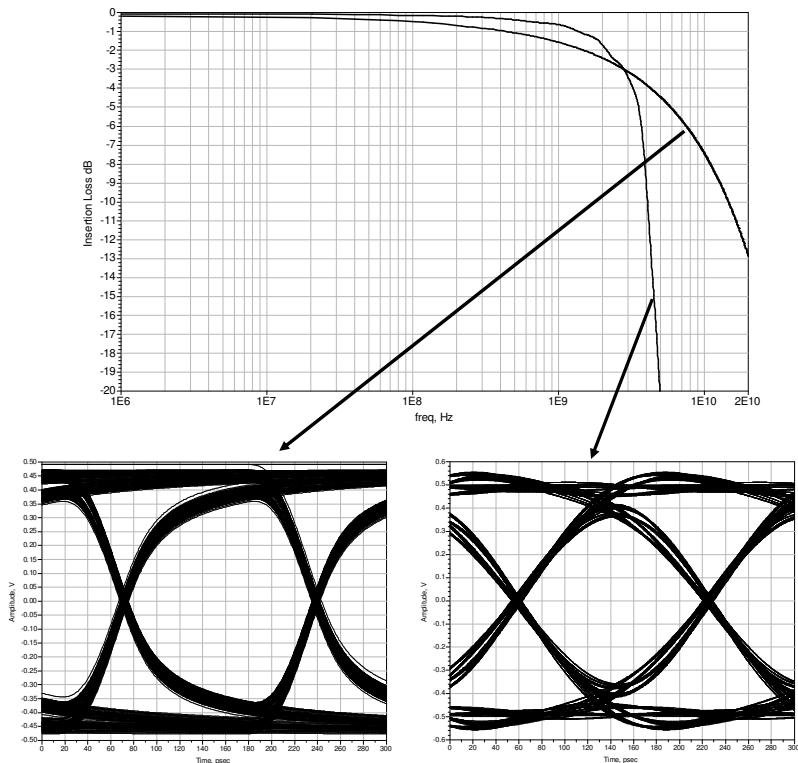


Figure 2.16 Frequency response of two different systems that have the same bandwidth value to a 6-Gbps PRBS.

The resulting data eye diagrams show a significant performance difference between the different frequency responses demonstrating the pitfalls of relying on the bandwidth value for evaluating the time domain performance of an instrument or interconnect. The main point is that knowing where the system frequency response crosses the -3 dB point is not as important as knowing how the system behaves in the frequency domain (i.e., does it roll off very fast after the -3 dB point or does it roll off slowly and in this way not attenuating so strongly the frequencies after the -3 dB point?). Also note that the flatness of the frequency response before the roll-off is also very important.

Although the complete frequency response of a measurement instrument or interconnect might not be available in several situations, it is important to know the frequency response not only for the -3 dB point but also the -6 dB and -10 dB point.

2.3 Bit Error Rate

The concept of bit error rate (BER) is the cornerstone for analyzing the performance of a high-speed digital link. Figure 2.17 shows a basic block diagram of a unidirectional digital interface. Errors due to noise are generated in different parts of the digital interface by physical processes. Depending on the magnitude of the noise and the margins of the digital interface, it might happen that a given noise event will generate an error on the data transmission. The probability of a transmission error occurring is given by the bit error rate (BER).

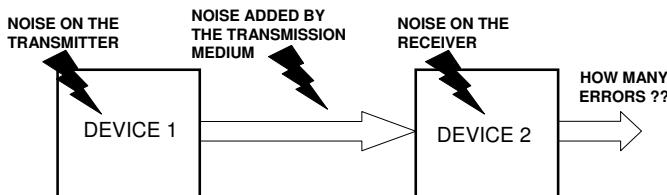


Figure 2.17 Simplified block diagram showing several sources of noise in a digital interface.

The bit error rate⁴ is defined as the ratio of the number of failed bits to the total number of transmitted bits as shown in (2.4).

$$\text{BIT ERROR RATE (BER)} = \frac{\text{NUMBER OF FAILED BITS}}{\text{NUMBER OF TRANSMITTED BITS}} \quad (2.4)$$

A BER of 0.5 means that on average, half of the transmitted bits will suffer from logical errors.

Note that the BER definition depends on the number of transmitted bits (i.e., typically when performing a BER measurement, only a limited number of bits are transmitted and measured). Since errors can be generated by random and deterministic events, for a perfect BER measurement an infinite number of measured bits would be required. Since this is not practical, a limited number of bits is used.

Another important point is the fact that a BER measurement result will also depend on the applied pattern, especially when using a limited size pattern. Figure 2.18 shows a very simplistic example that demonstrates this

⁴Bit error rate is also referred to by some authors as bit error ratio. In fact, as described in [2], bit error rate refers to a timing related parameter (e.g., number of errors per second) while bit error ratio could be considered the most appropriate term for (2.4). But since most engineers and technical publications use the term bit error rate when referring to (2.4), we will also use it through this book, although the reader should be aware of this possible distinction.

dependency. In the figure, the noise of the receiver moves some of the strobing edges to wrong positions that might generate bit errors. The measurement setup uses a 10-bit pattern to compute the BER of the system. With the first pattern a BER of 0.1 is measured, but with the second pattern which is different only in a single bit, the measured BER jumps to 0.2.

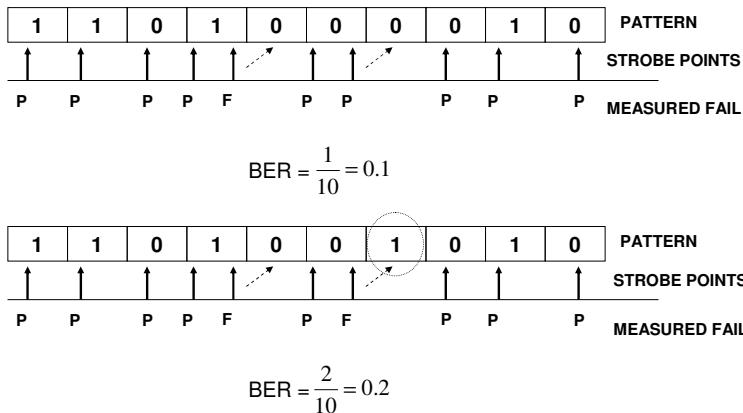


Figure 2.18 Simplistic example showing how the measured BER can be different from the “real BER” due to the used pattern.

For measuring a DUT BER, one needs to be able to transmit bits to the DUT, find out if the transmitted bit was correctly received by the DUT, receive bits from the DUT, and find out if the received bits correspond to the intended bits to be transmitted by the DUT. Figure 2.19 shows a block diagram of two possible measurement setups.

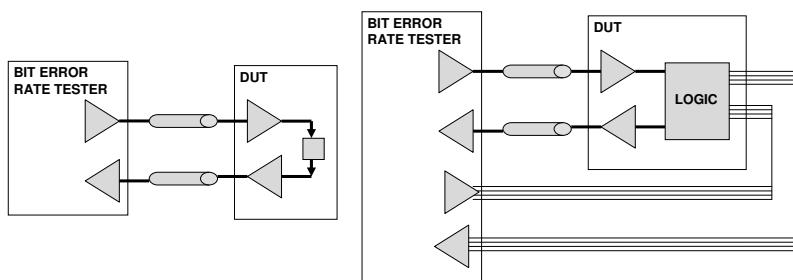


Figure 2.19 Block diagram of bit error rate measurement setup for a digital I/O cell in a DUT (left: The DUT I/O cell can be programmed to a loopback mode; right: the data must be obtained after the internal logic of the device in some lower speed pins).

In the first setup (Figure 2.19, left) a loopback configuration is available between the receiver and driver in the I/O cell. This is a common DFT technique in modern I/O cells [9]. In this way a known pattern can be sent to the DUT I/O cell receiver which in turn is transmitted by the I/O cell driver after going through some internal circuitry without any modification on the bit pattern (the exact circuitry will depend on how the driver/receiver loopback path was implemented). This approach is discussed in detail in Section 6.4.3.

Another more complex setup is shown in Figure 2.19 (right). In this case there is no DfT mechanism available to help with measuring the I/O cell BER. The BER must then be measured by analyzing all relevant inputs/outputs of the DUT. This can be a prohibitive task and that is why DfT is typically used on high-speed I/O cells, although simpler devices like multiplexers/demultiplexers have their BER rates sometimes measured in this fashion (e.g., [10]).

Although the previous examples only concentrated on measuring the BER of a single I/O cell in an IC, it is also possible to measure the BER of a complete digital link including the driver, receiver, and the link medium (e.g., a copper trace on a backplane printed circuit board) as shown in Figure 2.20. Typically this is more of a system test than an IC test performed with an ATE system.

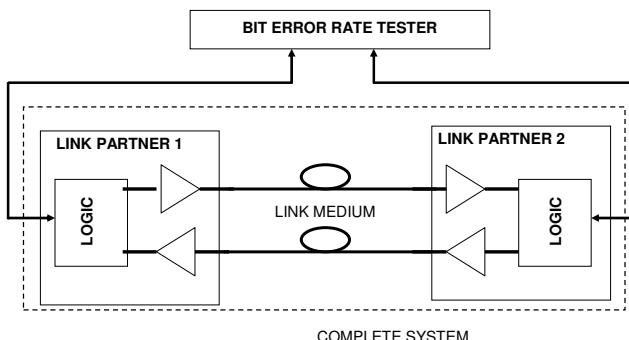


Figure 2.20 Block diagram of a bit error rate setup measurement for a digital link.

2.4 Jitter

Jitter and jitter testing are very important but sometimes complex subjects. There exists a myriad of resources spread through different books, articles, and applications notes. Some of the available references on this topic include [1, 2, 11–14]. This section addresses some of the important points regarding jitter in high-speed digital interfaces.

The International Telegraph and Telephone Consultative Committee (CCITT) has defined jitter as “short-term non-cumulative variations of the significant instants of a digital signal from their ideal positions in time.” A significant instant for a digital signal can be defined as the rising or falling edge from a bit transition crossing a voltage threshold level. To make this definition easier to understand, Figure 2.21 presents the most common jitter definition which is the time interval error (TIE), also sometimes referred to as the absolute jitter.

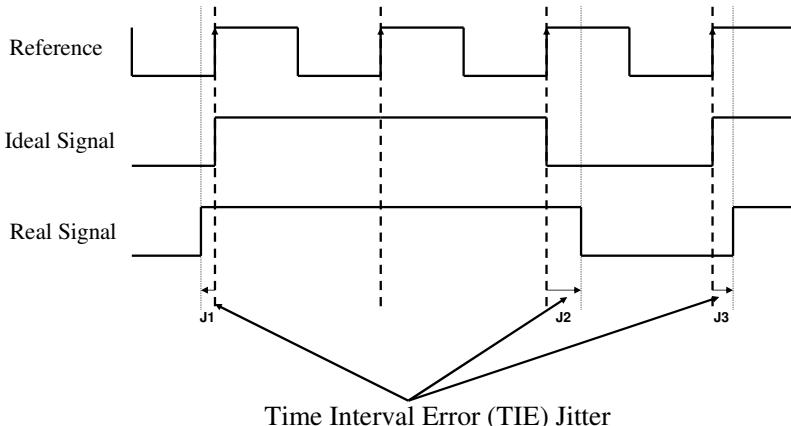


Figure 2.21 Standard definition of jitter on a digital signal, also referred to as time interval error (TIE) or absolute jitter.

In Figure 2.21, the correct timing (i.e., the ideal positions for the rising and falling edges of the signal) is defined by a perfect reference clock. An ideal signal should have the rising and falling edges timing aligned with this clock for each bit. A real signal in turn will have an imperfect timing and the timing difference between the real signal edge and the ideal position is the instant jitter value at that point or the time interval error. One important consequence is that jitter is only defined or measured when there is a transition (e.g., from a logic 1 to a logic 0). This means for a digital signal, only one jitter value is measured at each transition resulting in an array of jitter values.

The TIE jitter definition, although the predominantly used definition of jitter, is not the only one. Another possible definition for jitter in a digital signal, especially for clock signals, is the cycle-to-cycle jitter definition as shown in Figure 2.22 [15]. In this case the jitter value is measured by looking at the variation of the adjacent periods in regard to the expected clock period.

For clock signals it is also typical to define the maximum deviation of each single period of the clock signal under measurement from that of the ideal clock as the period jitter [15].

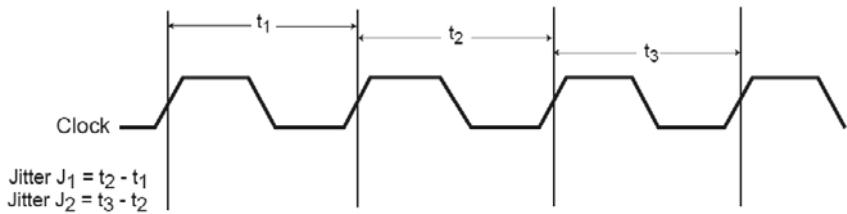


Figure 2.22 Cycle-to-cycle jitter definition.

2.4.1 Jitter Histogram

The jitter histogram is one of the most commonly used ways of analyzing the jitter of a digital signal. The idea is to measure the time interval error of each edge of the signal and generate a histogram of the measurements. The exact way the measurement array of values required for the histogram is generated depends on the specifics of the measurement methodology and instrument (to be discussed with more detail in Chapter 5). Typically an array of values for the measured jitter at each transition is generated. From this array a histogram can be computed. Figure 2.23 shows an example of an array of jitter values.

26.84	15.59	17.50	5.44
-19.77	7.69	27.91	29.79
36.36	13.93	6.42	28.74
-7.49	-2.25	32.47	-0.55
-29.03	-0.78	21.25	18.48
-12.37	1.76	4.28	-6.43
18.69	-15.79	17.54	13.22
21.12	28.46	3.89	38.31
3.20	0.13	-8.30	3.14
5.75	13.73	7.17	-6.01
12.66	-17.10	0.72	-10.00
-29.18	-21.50	-7.29	14.33
-11.63	-1.82	35.42	26.75
-36.60	-5.06	4.43	42.51
-8.98	23.90	54.61	1.08
18.99	12.12	-5.92	3.26
14.35	10.81	11.29	-12.65
45.76	-28.90	31.65	32.24
3.33	-19.35	54.58	-1.51
-43.13	4.04	6.07	-9.46
33.79	-6.96	-15.81	43.68
25.65	25.80	16.07	16.20
-11.65	26.82	-26.40	14.33
4.45	-11.62	-5.48	-20.11
-4.52	-21.84	10.40	8.68

Figure 2.23 Example of a set of 100 measured TIE jitter values (in ps).

To obtain the histogram from the array, a series of time interval buckets or bins (e.g., 1-ps-wide buckets) are created and then the number of jitter measurements that are inside each bucket is counted. The histogram can then be normalized to a percentage by dividing the number of samples in each bucket by the total number of measurement samples used to generate the histogram.

Figure 2.24 shows the plotted histograms (nonnormalized and normalized) for the jitter measurement samples of Figure 2.23 where a resolution of 500 fs was used for each bucket with a total span of -100 ps to 100 ps.

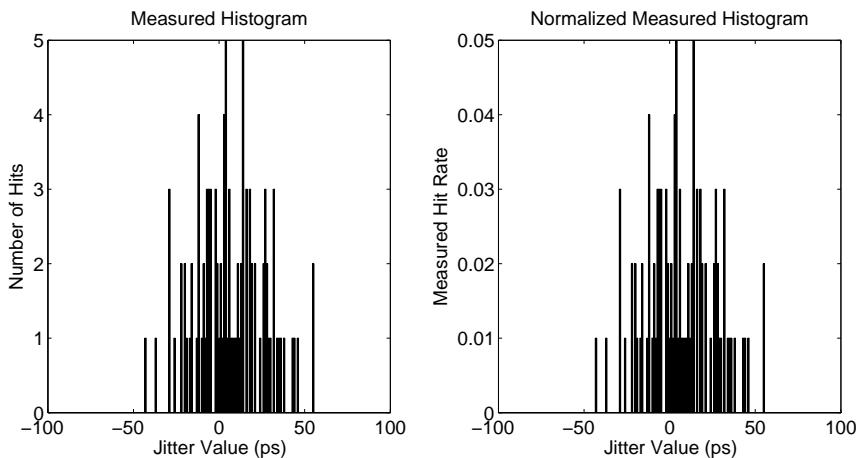


Figure 2.24 Jitter histogram plots (left: not normalized; right: normalized).

The jitter histogram provides a visualization on the amplitude of the jitter in the measured signal and might provide some clues on what type of jitter is present by analyzing its shape. In Section 5.5, two important measures (peak-to-peak and RMS jitter) that can be obtained from the jitter histogram are discussed in detail.

2.4.2 Jitter Categorization

Jitter is typically divided in several subcategories depending on its properties [12, 16, 17]. These properties sometimes provide clues to the origin of the jitter allowing the designer to more easily find the jitter source in the design. Figure 2.25 shows a typical categorization of jitter. Sometimes slightly different categorizations are used by different authors. The two main categories of jitter are random and deterministic jitter.

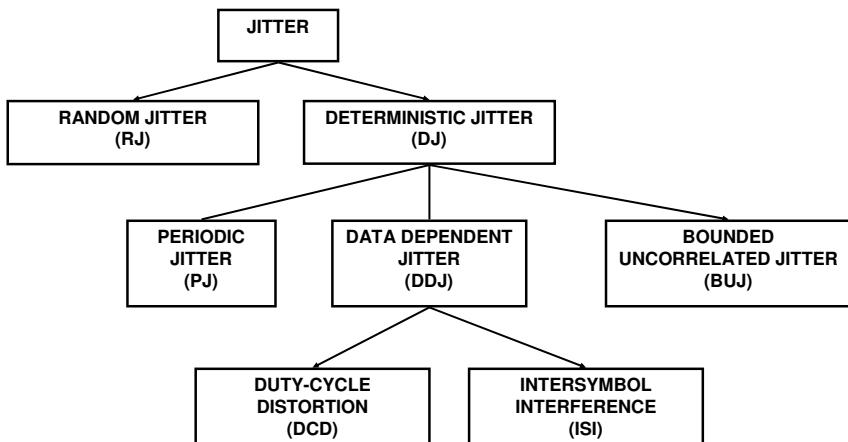


Figure 2.25 Categorization of jitter into different components.

Random Jitter

Random jitter (RJ) is defined as being unbounded in the sense that there is always a probability (although it can be very small) of the jitter value reaching any value. Random jitter is usually modeled using a Gaussian (also known as normal) probability distribution. One important reason for using a Gaussian distribution to model random jitter is due to the central limit theorem (see Appendix A). The theorem states that the sum of an infinite number of arbitrary probability distributions is a Gaussian distribution. Since in any semiconductor device there are multiple noise sources that contribute to the generated random jitter (shot noise, flicker noise, and Johnson noise [18, 19]), using a Gaussian distribution to model the random jitter behavior is a reasonable approach. Figure 2.26 shows a simulation example of a digital data signal at a data rate of 5 Gbps that contains random jitter (with a 10 ps RMS⁵ value).

In Figure 2.26 it is not easy to discern the jitter on the signal waveform. Only the representation in a data eye diagram makes the presence of jitter clearly obvious. The jitter histogram shows a distribution that is not exactly matching the expected Gaussian distribution due to the limited number of samples. If the number of samples to be acquired would be increased, the histogram would converge more closely to the expected Gaussian distribution.

⁵The root mean square (RMS) value of a distribution is defined in Section 5.5.2. For a Gaussian distribution the RMS values correspond to the standard deviation σ (see Appendix A).

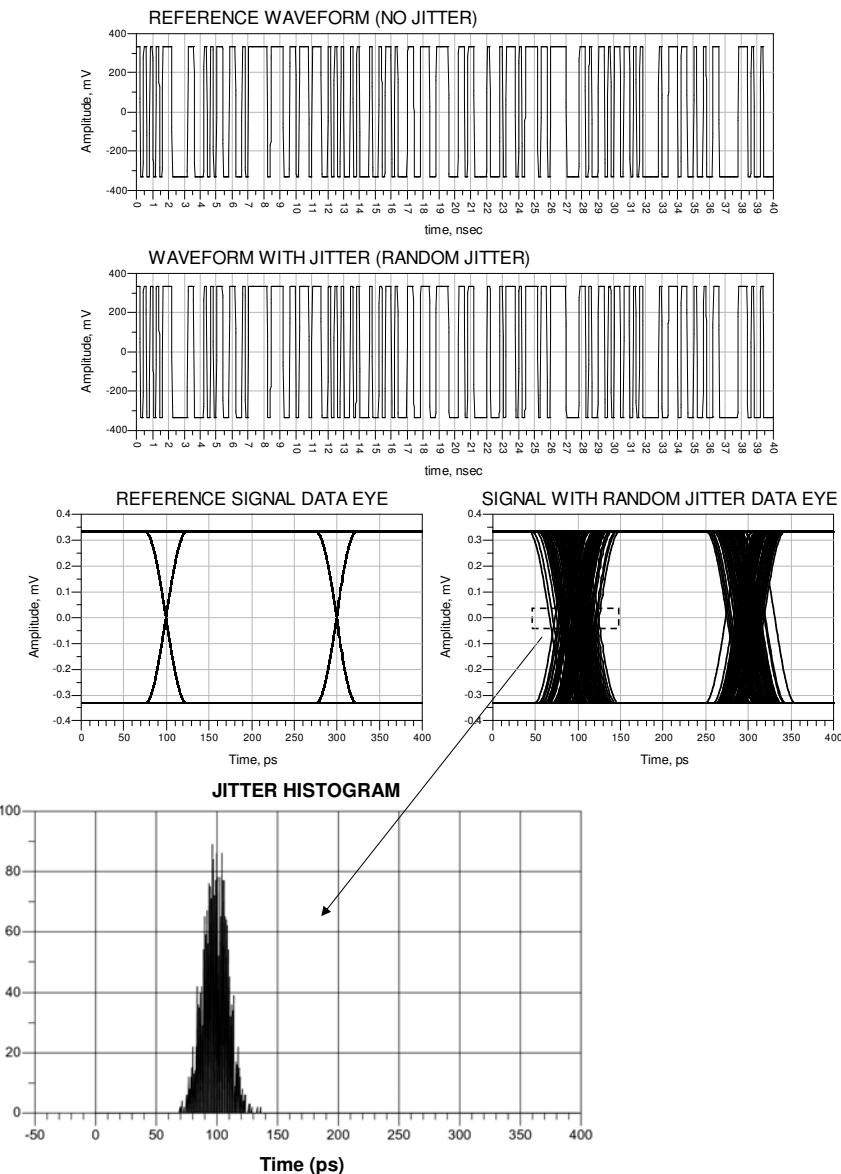


Figure 2.26 Example showing the simulation of a digital signal with only random jitter.

Deterministic Jitter

Deterministic jitter (DJ) is mainly characterized by being bounded (i.e., the maximum and minimum jitter values are limited unlike with random jitter). Deterministic jitter can be further divided into several subcategories depending on the underlying physical cause: periodic jitter (PJ), bounded uncorrelated jitter (BUJ), data dependent jitter (DDJ) in the form of duty-cycle distortion (DCD) and intersymbol interference (ISI).

Periodic Jitter

Periodic jitter corresponds to jitter that has a periodic nature but is not correlated to the signal data pattern. One example is the crosstalk from an adjacent noncorrelated clock signal into the data signal being measured. One typical model for periodic jitter is a sinusoidal waveform, which will also be important for the jitter tolerance test discussed in Section 5.7.2. Other examples of periodic jitter waveforms, like triangular waveforms, are used, however with limited practical use. Figure 2.27 shows an example of the simulation of a 5-Gbps data signal with sinusoidal periodic jitter.

The histogram shows the typical shape expected from a data signal with sinusoidal deterministic jitter with the time difference between the two peaks at the extremes corresponding to the periodic jitter amplitude.

Data Dependent Jitter

Data dependent jitter (DDJ) as the name indicates is correlated with the data pattern. It is typically subdivided into duty-cycle distortion (DCD) and intersymbol interference (ISI).

Duty-Cycle Distortion Jitter

Duty-cycle distortion jitter refers to the jitter generated in a clock signal with different widths of the logical high and the logical low bit. Figure 2.28 shows an example of the added DCD jitter for a bit clock with a duty cycle unequal to 50%.

In the context of DCD jitter, there is another type of jitter referred to as F/2 jitter. In the case of F/2 jitter the width of even bits is different from the width of odd bits which also results in a similar jitter histogram distribution but in a different data eye diagram [20].

Some possible causes of DCD are imbalance in the driver source and sink current, nonlinear loads, marginal timing of output drivers, and common mode voltage in differential signals.

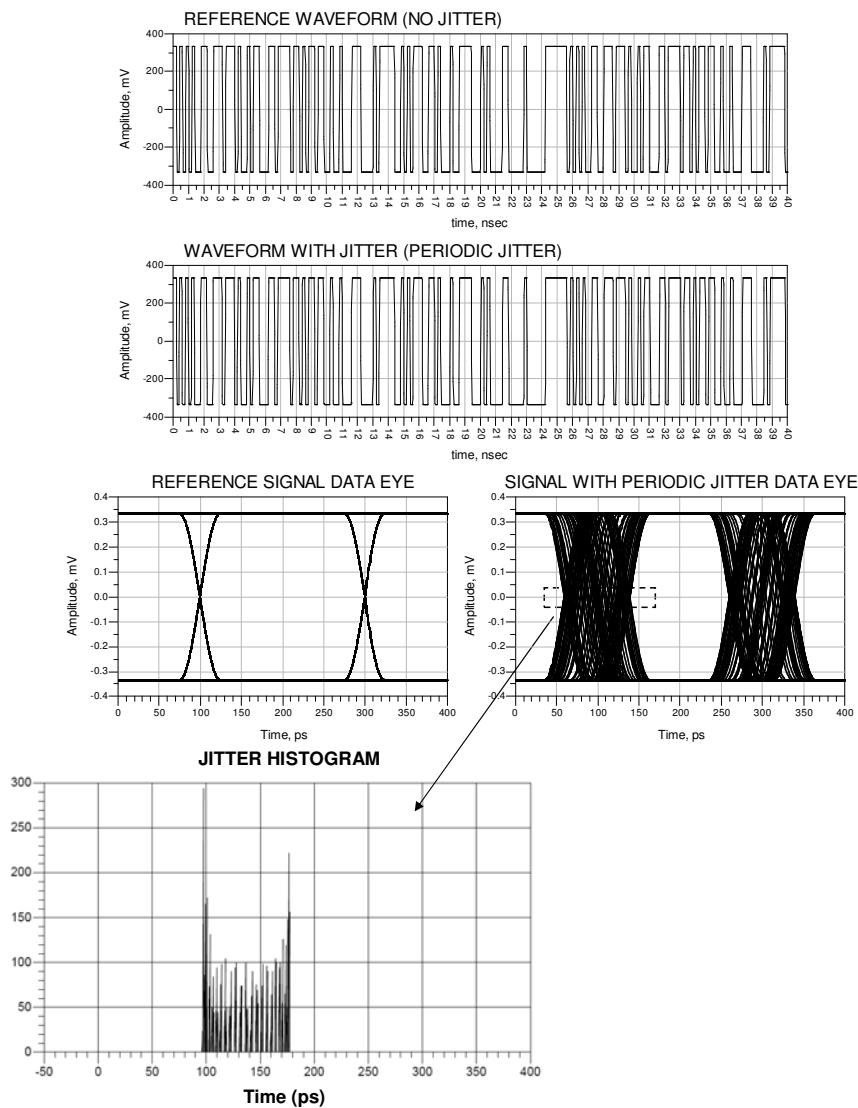


Figure 2.27 Example of periodic jitter.

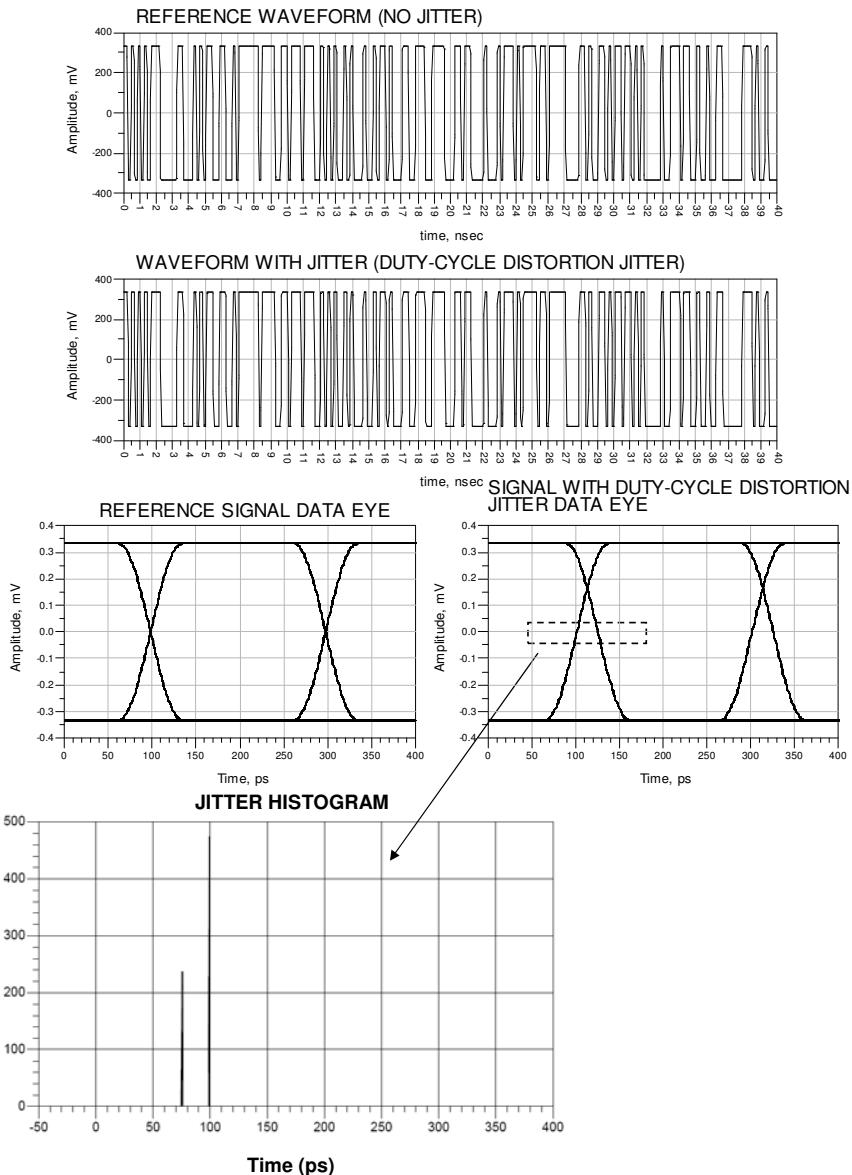


Figure 2.28 Example of data dependent jitter in the form of duty-cycle distortion jitter.

Intersymbol Interference Jitter

Intersymbol interference jitter is based on the influence of the preceding bits on a transition. It can be observed when a digital waveform passes through a bandwidth limited channel like a lossy signal path in a printed circuit board trace. This is exemplified in Figure 2.29.

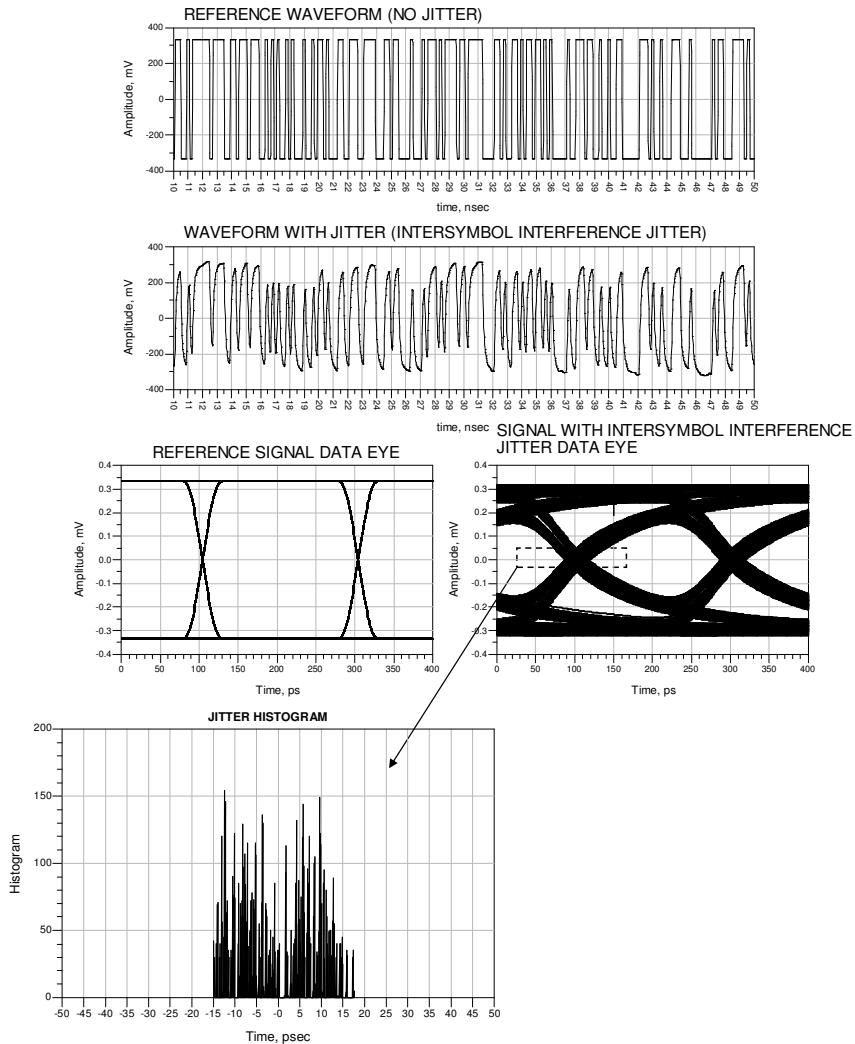


Figure 2.29 Example of data dependent jitter in the form of intersymbol interference jitter.

From the figure it is possible to see the effect of a lossy channel on the data signal waveform (more on this in Chapter 8) and how the previous bits have the effect on the current bit transition and in this way add jitter. In case of a printed circuit board trace, the shape of the bit rising edge will depend on how the line was charged by the previous bits, and it is this dependence that creates the data dependent jitter. Interestingly, this can be seen in the example of Figure 2.30 where a pattern consisting of two repetitive symbols is plotted in the frequency and time domain showing the contribution of each symbol for the ISI. Some more analytical work on predicting DDJ is presented in [21].

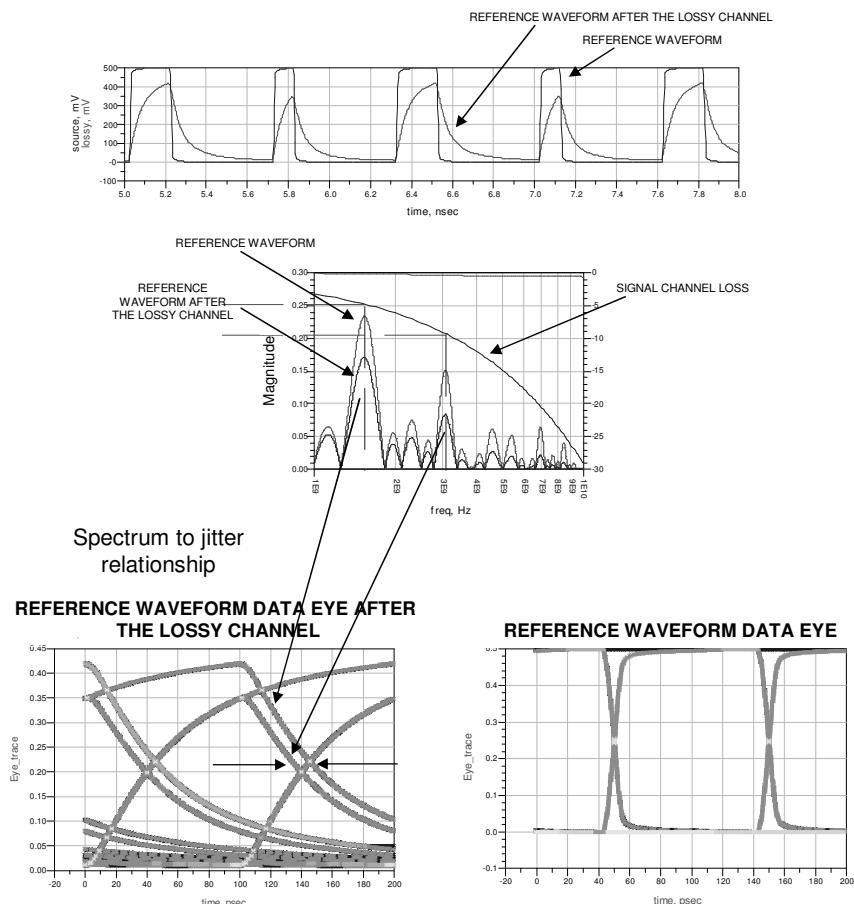


Figure 2.30 Signal path loss effect on the spectral power density and data eye for a two-symbol alternating pattern 11000001000001100000100000... [22].

Bounded Uncorrelated Jitter

Bounded uncorrelated jitter is defined as deterministic jitter that is bounded but uncorrelated with the data pattern. One example of a mechanism that causes this type of jitter is the crosstalk from an adjacent data line belonging to a different bus with a different frequency. In fact, periodic jitter can be considered a form of BUJ but due to historical and practical reasons, PJ, especially sinusoidal jitter, has a separate classification.

To demonstrate how BUJ looks on the time domain, the simulation setup in Figure 2.31 was used where a 5-Gbps data line is coupled (i.e., physically very close in the PCB board) to a 3.2345-Gbps data line belonging to a different bus. The crosstalk from this lower speed data line will create BUJ on the 5-Gbps data signal as shown in Figure 2.32.

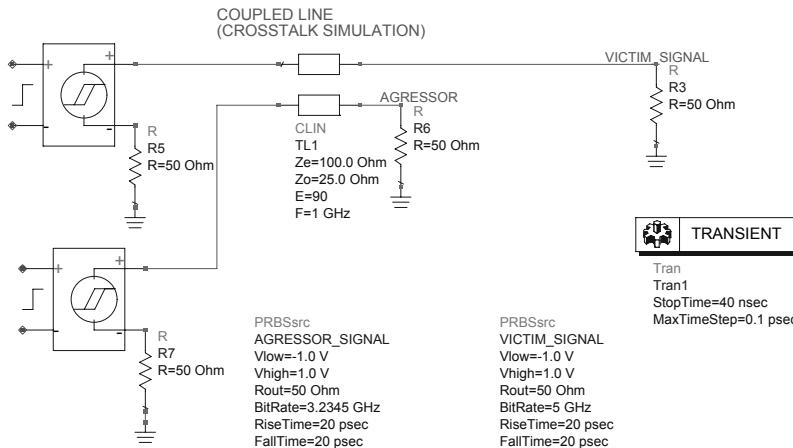


Figure 2.31 Simulation setup for demonstrating bounded uncorrelated jitter.

In this example the crosstalk is very strong in order to clearly show its effect. It is easy to see the degradation of the signal on the time domain waveforms and on the data eye diagram. Note also that the jitter histogram does not have a shape that would allow one to clearly conclude that the jitter was of the BUJ type due to the fact that the crosstalk source is uncorrelated.

2.4.3 Amplitude Noise and Conversion to Timing Jitter

The previous sections have only discussed timing jitter, but a digital signal also contains noise in the voltage levels. While timing noise is referred to as jitter, voltage level noise is usually referred to as amplitude noise. One critical point is that any amplitude noise in a signal is translated into timing jitter as shown in Figure 2.33.

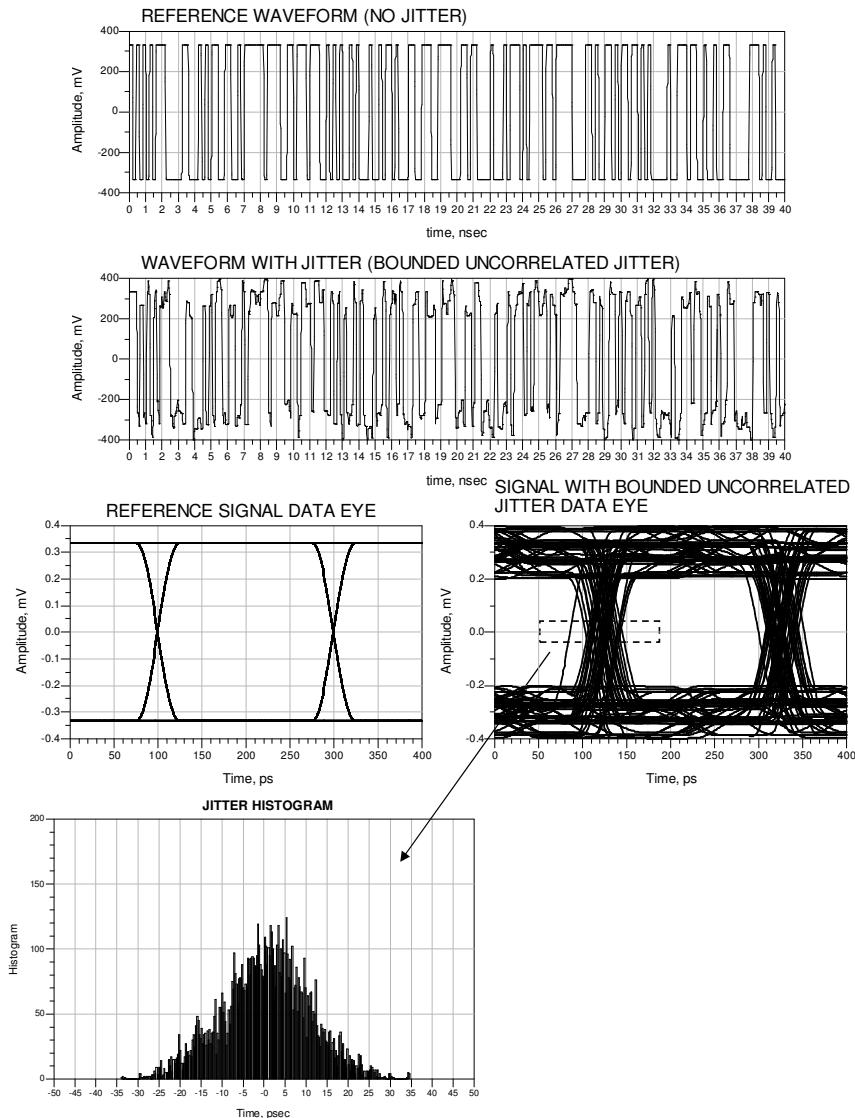


Figure 2.32 Example of bounded uncorrelated jitter.

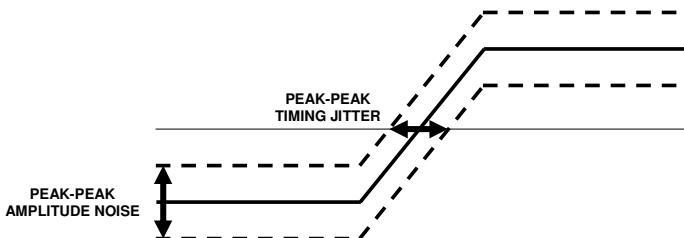


Figure 2.33 Amplitude noise to timing jitter conversion using a linear model.

Using a linear model, the timing jitter peak-to-peak value can be computed from the amplitude noise peak-to-peak value using the following equation:

$$\text{Timing Jitter (peak-to-peak)} = \frac{dV}{dt} \times \text{Amplitude Noise (peak-to-peak)} \quad (2.5)$$

where dV/dt is the slew rate of the digital signal.

2.4.4 Jitter in the Frequency Domain

The previous sections looked at jitter in the time domain (e.g., the jitter histogram) but it is also possible to analyze jitter in the frequency domain. This approach can provide significant insights on the causes of the measured jitter. How to compute the jitter spectrum is directly related to the methodology used to measure the jitter.

The straightforward way to analyze jitter in the frequency domain is to compute its spectrum (i.e., the jitter frequency spectrum). This can be easily done by performing the discrete Fourier transform (DFT) on the time domain jitter signal. The challenge is how to obtain the time domain waveform of the jitter embedded on the data signal waveform. For example, if the measured data is in the form of a real-time sampled waveform, then the jitter waveform can also be obtained by comparing the transition times of the waveform with the ideal transition times. One challenge is that only during logical transitions the possibility is given to calculate a jitter value point for this time instant. A solution to obtain the missing jitter values during intervals without logic transitions is to apply an interpolation algorithm.

Figure 2.34 shows an example of a jitter spectrum comparison between a digital signal with only random jitter and a digital signal with a periodic sinusoidal jitter component at 100 MHz. Note that the data eye diagram and measured histogram in both cases are very similar. Only the analysis in the

frequency domain clearly reveals the presence of a periodic jitter component at 100 MHz. One reason why the frequency domain view is so powerful is that the random jitter is spread across the frequency range allowing the deterministic jitter components to become more visible when compared to a time domain approach like a histogram.

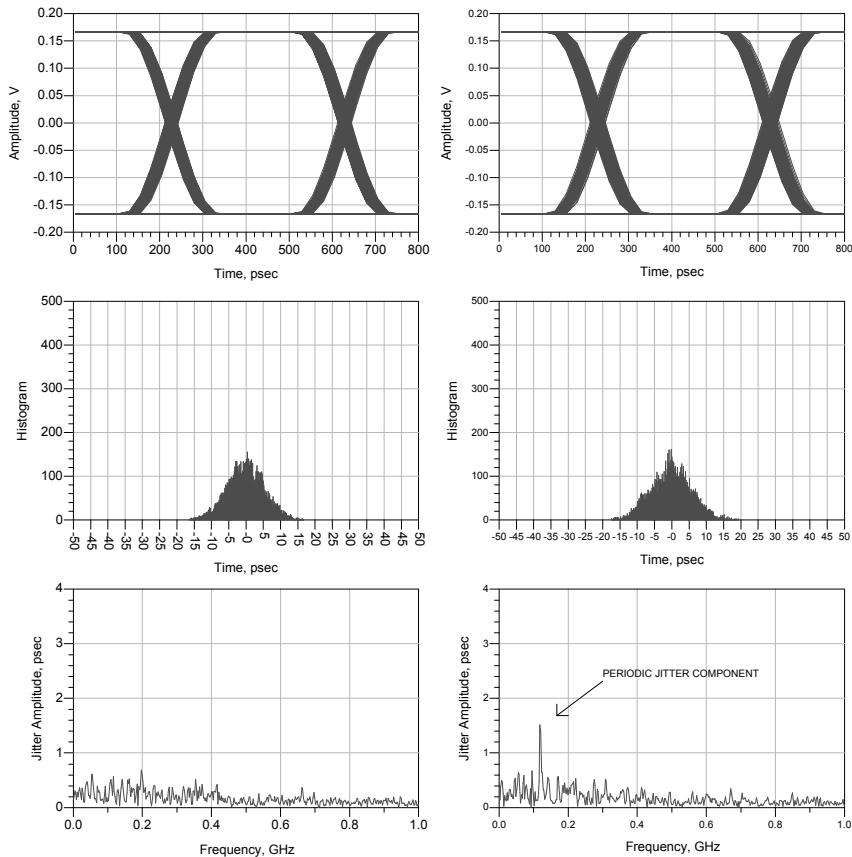


Figure 2.34 Comparison of the jitter spectrum obtained from a digital signal with only random jitter (left) and a signal with random jitter and a periodic jitter component in the form of sinusoidal jitter at 100 MHz (right).

Figure 2.35 shows an example of a jitter spectrum comparison between a digital signal with random and deterministic jitter in the form of ISI from a lossy signal trace and a digital signal with random jitter and deterministic jitter in the form of BUJ obtained through crosstalk from another digital signal using the approach presented in Figure 2.31.

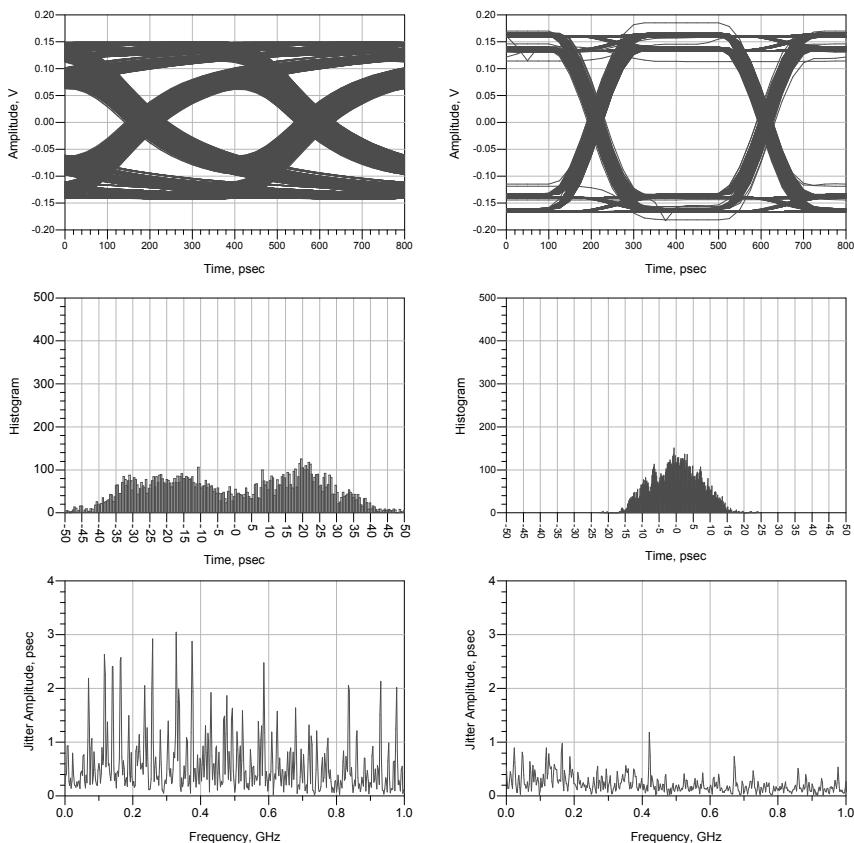


Figure 2.35 Comparison of the jitter spectrum obtained from a digital signal with random jitter and deterministic jitter (left) in the form of ISI and a signal with deterministic jitter in the form of BUJ (right).

In this case it is easy to observe that for ISI type deterministic jitter, the jitter spectrum is composed of several discrete lines at multiple frequencies. In the case of BUJ jitter, the spectrum is not that different from a random jitter spectrum making it more difficult to separate the BUJ from the random jitter noise floor. Note that jitter components having very low frequencies (e.g., Hertz) are typically not called jitter but wander.

As mentioned before, jitter frequency decomposition techniques depend on the measurement approach. In the case where the measurement instrument is a digital pin electronics, the technique to obtain the frequency jitter decomposition is not straightforward, since we do not have a real-time sampled waveform available. One possible technique is using the error density as described in [23]. This technique is discussed in detail in Section 5.5.4.

2.5 Classification of High-Speed I/O Interfaces

Digital data transmission interfaces can be classified according to the clocking architecture they use. Lower speed interfaces typically deploy common clock architectures, where devices communicating with each other derive their data generation and sampling timing from a centrally provided system clock. This approach has proven to not be feasible with high-speed I/O interfaces. The distribution of a central clock with the required timing accuracy especially with regard to clock skew over a whole system easily results in high system design efforts and leads to high system cost. Thus, high-speed interfaces typically use source synchronous timing concepts where each device derives its own timing from a local or central system clock, but the data transmission timing reference is determined by the sending device that transmits this clock information together with the sent data. The receiving device derives the phase of this transmitted clock information and uses it to correctly latch the incoming data.

It is important to mention that the term “source synchronous” as used in this book has to be taken literally for any interface that transmits phase information about the transmitted signals together with the transmitted data, regardless if it is on the same or separate signals. Sometimes this term is misleadingly used to describe interfaces that explicitly transmit a clock signal together with data signals. Using our more general interpretation of source synchronous interfaces, all important high-speed I/O interface standards today use a source synchronous data transmission concept. Source synchronous interfaces can be subclassified in embedded clock interfaces where the transmission clock phase information is sent together with the transmitted data on the same physical signal trace. The other representatives of the source synchronous group are at-cycle source synchronous interfaces and forwarded clock interfaces which leads to the overall classification shown in Figure 2.36. All source synchronous interfaces have the advantage that the clock information is originating from the same source as the data signals. Thus, transmitter timing variations such as thermal drifts of certain jitter components are present on both, the data content and the clock. This widens the margins to latch the data correctly at the receiving device.

The architectural topologies for the different interface classes are shown in Figure 2.37.

Common Clock Interfaces

As mentioned, although common clock interfaces are the classical clocking architecture for digital interfaces this interface type usually is not found in the high-speed I/O interface arena. Correct data transmission using a common

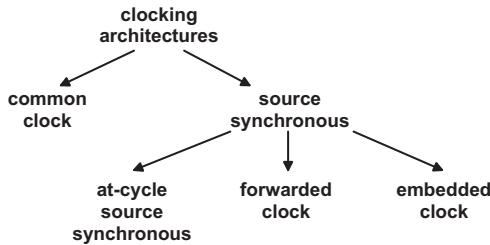


Figure 2.36 Clocking architectures for high-speed interfaces.

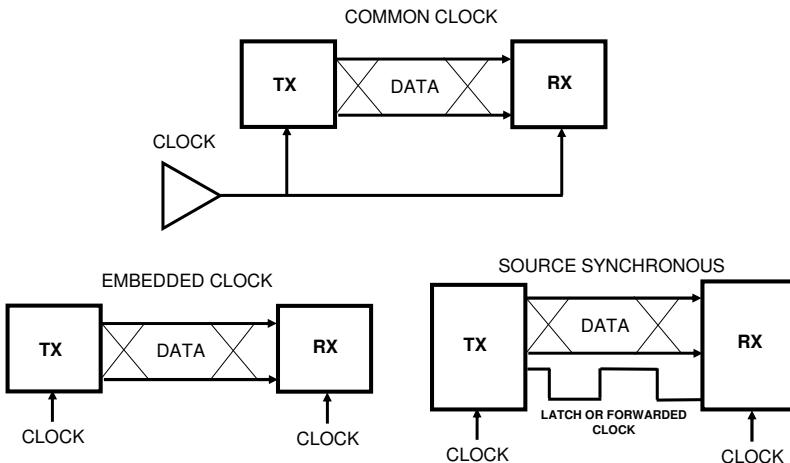


Figure 2.37 Different clocking models for high-speed digital transmission.

clock interface relies on accurate clock distribution with well defined relative phase conditions between clock and data for each device. System design parameters that impact this phase relationship are skew introduced by the clock routing paths, skew within the data connections for a bus, as well as skew between these data signals and the distributed clock signals. Since for high-speed I/O interfaces bit time intervals are small, the allowed margins for these skew components are so tight that high-speed I/O system design is not possible anymore in an economical way using common clock architectures.

At-Cycle Source Synchronous Interfaces

At-cycle source synchronous interfaces transmit a clock signal together with the signals of the data bus. For wide data busses, at-cycle source synchronous interfaces usually divide the bus into subbusses and one clock signal is transmitted together with the signals of each subbus. At the receiving

device, the received clock signal is directly used to latch the associated data signals. For at-cycle source synchronous interfaces, there is a one-to-one phase relationship (isochronous timing) between the clock and the data signals. Thus, clock and signal connections have to keep the same electrical length from the transmitter up to the latching circuit in the receiver. The isochronous timing conditions for at-cycle source synchronous interfaces have the advantage that even high-frequency jitter components caused by a transmitter can be tracked by this kind of interface as long as they affect clock and data in the same way. On the other hand, for very high-speed interfaces with short bit times, the margins for the allowed setup/hold times at the receivers become small. This in fact can cause very similar challenges for designing systems using this kind of interfaces as in the case for common clock based systems. Even small variations in the electrical length between data and clock signals or nonuniform timing variations between data and clock can cause a violation of the allowed setup/hold times.

Forwarded Clock Interfaces

In contrast to at-cycle source synchronous interfaces, the clock signal of forwarded clock interfaces serves as a relative phase reference only. The difference between the at-cycle source synchronous and the forwarded clock concept is shown in Figure 2.38. For at-cycle source synchronous interfaces an isochronous timing concept is used where a fixed relative timing relationship between a clock edge and a corresponding data bit has to be kept for the complete transmission path. On the other hand, forwarded clock interfaces use a mesochronous concept where this fixed clock edge to data bit phase relationship is not existent anymore for the data transmission. Of course, a valid relative timing relationship that does not violate setup/hold time margins between clock edges and data bits needs to be achieved at the receiving end of the data transmission path. However, for forwarded clock interfaces it is not required that a distinct clock edge at the source keeps its phase relationship with the corresponding data bit while being transmitted. Valid setup/hold time conditions at the receiving end of the transmission path are typically set up using training steps during device initialization and potentially during device operation if this is required. The benefit of this training concept is that PCB designers do not need to keep phase matched trace lengths between the clock and data connections as well as within the data connections. Static phase mismatches on these connections are compensated by the device training.

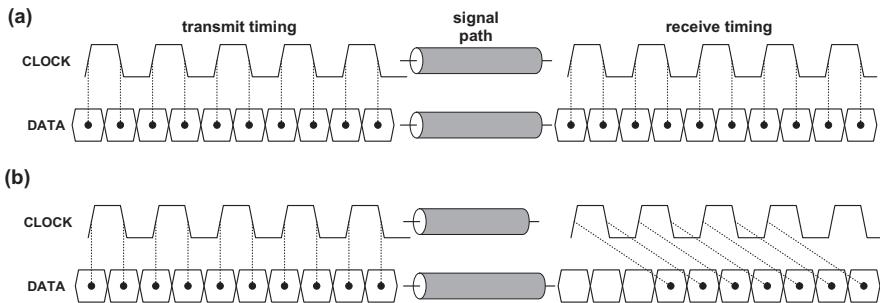


Figure 2.38 Comparison of (a) at-cycle source synchronous interface versus (b) forwarded clock interface.

Embedded Clock Interfaces

Embedded clock interfaces combine the clock information and the data content within the same signal. On the receiver side, a so-called clock data recovery (CDR) circuit separates the clock and data information from this signal and reconstructs the data content. The benefit of combining the data and clock phase information on the same signal is that both are affected equally by signal disturbances. This allows embedded clock interfaces to transmit data at very high data rates on a PCB. In order to allow the extraction of clock phase information and the reconstruction of the data in the receiver, embedded clock data transmission has to fulfill requirements such as having enough data transitions on the transmitted signal. This usually is fulfilled by deploying appropriate coding or scrambling (see Appendix D). Also, the methodology used to extract data and clock information (see Section 2.6.3) has influence on which kind of timing variations can be tolerated on embedded clock interfaces and which kind is likely to cause data transmission problems.

Digital interfaces can be further divided into unidirectional (also referred to as simple duplex) and bidirectional interfaces (also referred to as full duplex). Unidirectional interfaces only allow transmission of data in one direction, meaning that each I/O cell requires a separate transmitter and receiver pin. A bidirectional interface uses the same media or pin for the transmitter and receiver parts of the I/O cell as shown in Figure 2.39.

Clearly full duplex interfaces have the advantage of requiring fewer pins since both the transmitter and receiver share the same pins. The challenge with this type of approach is the I/O cell design. Typically bidirectional interfaces are behind unidirectional interfaces regarding the maximum achievable data rate.

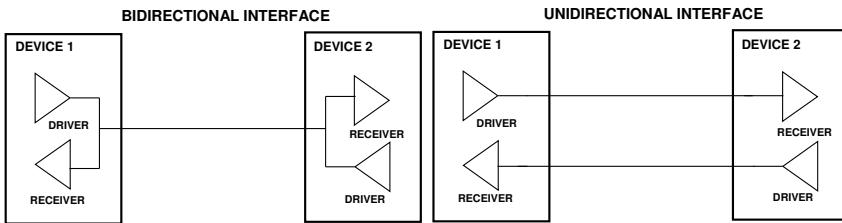


Figure 2.39 Comparison of unidirectional and bidirectional interfaces.

2.6 Hardware Building Blocks and Concepts

This section provides a high-level description of the important building blocks of high-speed I/O cells. In a high-speed I/O interface, the most important building blocks are phase locked loops (PLL), delay locked loops (DLL), clock and data recovery (CDR), and equalization circuits. These components of an I/O cell define the timing behavior of the high-speed signals under measurement and it is important for a test engineer to understand the characteristics of these circuits.

2.6.1 Phase Locked Loop (PLL)

A phase locked loop is a circuit that aligns the signal phase of an oscillator contained in the PLL with the phase of a reference input signal [24]. Besides this phase alignment or clock synchronization functionality, another application area for PLLs is to derive clock signals that are synchronized to the provided reference signal but run at different frequencies than the reference signal. In high-speed I/O circuits, this functionality of a PLL is used to generate a high-speed bit clock from a lower speed system clock. The derived high-speed clock is used to determine the final I/O data rate and among other things triggers the bit transitions of the high-speed transmitters of a device. Figure 2.40 shows a basic block diagram of a PLL.

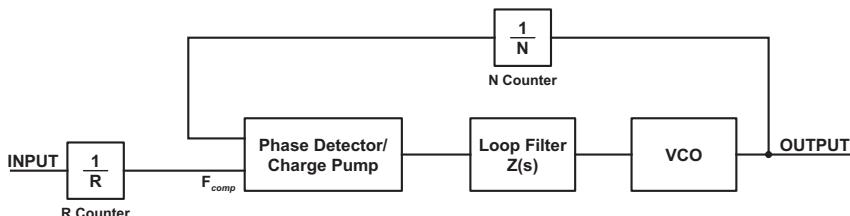


Figure 2.40 Basic block diagram of a phase locked loop.

PLLs are implemented as closed loop systems. The (potentially divided) input frequency serves as comparison frequency F_{comp} input for a phase detector with an integrated charge pump. This phase detector with the help of the integrated charge pump generates a current on its output that represents the instantaneous phase difference between the comparison frequency and the PLL output frequency that is fed back via a divider to the second input of the phase detector. The generated current is converted via the loop filter impedance $Z(s)$ into an average voltage that is fed into a voltage controlled oscillator (VCO). The VCO converts the voltage at its input into a proportional output frequency that serves as output of the PLL as well as input to the feedback loop of the PLL. The N counter (divider) shown in this figure provides the capability to derive other frequencies than just the one defined by the reference signal. With a frequency division of N in the feedback loop, it is possible to create a high-frequency clock that is synchronized with a reference clock that has a $1/N$ frequency of the output clock. The loop filter is the most critical component in the PLL since it is the part that predominantly determines the performance and stability of the whole PLL circuit. It also is responsible for the PLL characteristics that are important to understand for test engineers measuring high-speed I/O signals. PLL loop filters are implemented as low pass filters. Most PLLs integrated in consumer and computation high-speed I/O devices use second-order lowpass filters that allow control of the natural frequency ω_n and loop dampening ζ with the design of the filter. The open loop transfer function of a typical second-order PLL loop filter follows (2.6). This results in a transfer function for the closed loop PLL system as described by (2.7). Figure 2.41 shows these transfer functions for the open loop filter and for the overall PLL. A description of how these transfer functions are derived from the single PLL components can be found in [25].

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.6)$$

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.7)$$

The PLL transfer function describes the phase relationship of the PLL output relative to the PLL input with respect to the frequency at which the phase change happens. Since phase changes of a clock signal can be transferred into the time domain as drifts or jitter, this transfer function also describes how jitter with a certain frequency is transferred through the PLL. At low frequencies all of the jitter that is present at the PLL input is transferred to the output of the PLL. As illustrated in Figure 2.41, there is a frequency-range where the jitter is amplified from the PLL input to the PLL output. The amount

of amplification in this peaking range mainly is determined by the dampening factor ζ selected for the loop filter. The magnitude of the jitter attenuation at the high frequencies is mainly determined by the order of the loop filter that is used in the PLL. As shown in Figure 2.41, the second order loop filter we considered in our discussion attenuates high-frequency jitter with a slope of -40 dB/decade.

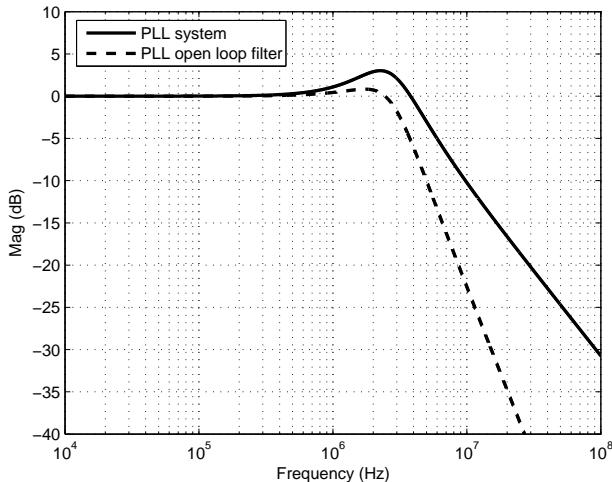


Figure 2.41 Transfer function of a second order PLL and open loop filter.

Another characteristic of PLLs that influences the test implementation for high-speed I/O interfaces is the fact that the phase relationship between the PLL input and its output is neither necessarily predictable nor repeatable. As a consequence of this behavior, the phase offset between the PLL output and its input might change if the input to a PLL is driven by a signal that has not enough signal transitions to allow the PLL keeping its lock state. The same effects can be observed when the PLL goes through a (potentially local) power cycle that might be triggered by a global reset or a reinitialization of the device that contains the PLL. This behavior causes timing phase uncertainties of a DUT in a test environment. As a consequence, the ATE timing usually has to be readjusted to the phase of the signals that are part of the clock domain defined by the PLL output if the PLL lost its lock state due to missing input data transitions or a PLL reset. This is also true if the PLL is restarted without changing the DUT.

2.6.2 Delay Locked Loop (DLL)

A delay locked loop is a feedback system that tracks the phase of a reference signal. Compared to a PLL, a DLL does not contain a VCO as phase control element, but a delay line that is controlled by the phase detector. Also DLLs typically contain a first order loop filter. The usage of a first order filter simplifies the goal to achieve a stable servo loop for a DLL compared to the PLL. A DLL also does not exhibit jitter accumulation as is the case for a PLL and has the big advantage that its input can be shut down without causing phase jumps on its output after restarting the input signal. On the other hand, a DLL typically does not filter jitter of its reference input and cannot be used as M over N frequency synthesizers like PLLs. Figure 2.42 shows a typical block diagram of a DLL.

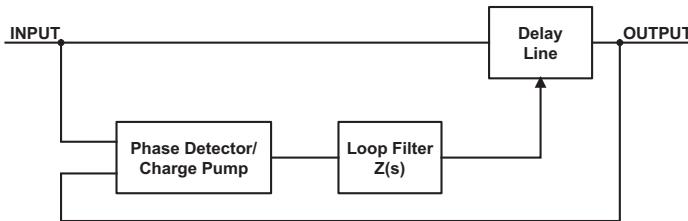


Figure 2.42 Basic block diagram of a delay locked loop.

DLL's are commonly used in high-speed I/O designs especially to implement features like skew compensation between multiple interdependent input signals of a device. Figure 2.43 shows a basic block diagram for such a skew compensation circuit and how DLLs in such circuits are used to remove the skew between signals (e.g., clock and data). With this feature, IC designers can address one major challenge in high-speed design, which is the alignment between the different data and clock signals. Without the usage of DLLs, in the past system designers had to make sure that all connections for these interdependent signals were matched with respect to their electrical length. With increased data rates, this task became more and more difficult. With the use of DLLs, the device can take care of the alignment internally, simplifying PCB routing for system designers significantly.

2.6.3 Clock and Data Recovery (CDR)

Clock and data recovery is used for embedded clock signals to extract the clock information from the received signal and to use this extracted clock signal to recover the data content from the received signal. There are a variety of different architectures and approaches for clock and data recovery (CDR)

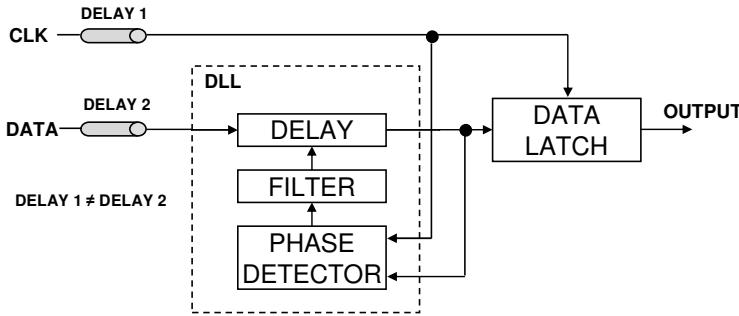


Figure 2.43 Basic block diagram of using a DLL to remove the skew between a clock and a data signal.

from high-speed digital signals and a large amount of published work in this area is available [7, 24, 26]. The most commonly used CDR circuit architecture is based on the usage of a PLL to extract a clock signal from the transitions that are present on the received signal. Figure 2.44 shows a simple block diagram of such a PLL based CDR circuit. A requirement to guarantee the correct operation of a PLL based CDR is that the received signal has to provide enough transitions so that the CDR phase cannot drift relative to the incoming signal. The required transition density usually is achieved by applying appropriate coding or scrambling (see Appendix D) to the transmitted data.

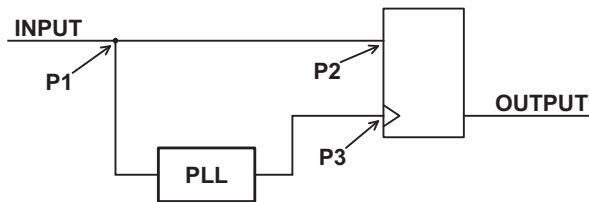


Figure 2.44 Basic block diagram of a PLL based clock and data recovery circuit.

In a PLL based CDR, the incoming signal is routed in parallel to the input of a PLL and the data input of the input latch circuitry for that pin. The PLL in the CDR generates a clock that is phase locked to the incoming signal. This extracted clock is used to sample the data content of the incoming signal in the center of the data bits. Since the sampling clock is derived from the incoming signal via a PLL, the transfer function as described in Section 2.6.1 also applies for the clock extraction. In particular this means that low-frequency jitter in the received signal by the CDR will be passed into the extracted clock signal while high-frequency jitter in the received

signal will be filtered according to the PLL transfer function. This low pass filter characteristic of the PLL in the CDR has quite some impact on the functionality of the overall CDR and the way jitter measurement results of high-speed I/O transmitters have to be interpreted.

Let's assume the signal at P1 of Figure 2.44 only contains low frequency jitter that passes the loop filter of the PLL completely. Let's also assume that the PLL does not introduce a phase difference between the signal at P1 and the extracted clock at P3. In such a case the jitter of the signal at P2 and the jitter of the clock at P3 are identical in amplitude and frequency. Thus, at all instances the timing delta between the signal transitions at P2 and the clock transitions at P3 are identical and maximum setup/hold times can be kept for latching the data content of the received signal. If the jitter frequency of the signal at P1 is identical to the frequency at the -3 dB point of the transfer function for the PLL, then the amplitude of the jitter of the extracted clock at P3 only will be 50% of the jitter in the signal at P2. This difference will cause a relative movement of the data eye at P2 compared to the clock edges at P3. This relative movement will reduce the available setup/hold times for the data latch and thus will shrink the available data eye at the data input of the latch. If the jitter frequency increases further to values where the jitter transferred through the PLL is more attenuated, the relative movement between the signal at P2 and the clock edges at P3 increases and the available setup and hold times for extracting the data content of the received signal are reduced down to a level where no reliable data extraction is possible anymore if the jitter amplitude on the received signal is too large.

Due to this behavior, jitter on a received signal can be separated into harmless low frequency jitter that has no influence on the data extraction and harmful high frequency jitter that reduces device internal setup/hold times of the data extraction and can cause bit errors. The exact classification of which jitter frequencies are harmless and which jitter frequencies are harmful is determined by the transfer function of the PLL used in the CDR. This separation into harmful and harmless jitter is especially important if the jitter that is measured on a transmitter needs to be evaluated against a single maximum jitter amplitude allowed by a specification. This evaluation has to take into account whether the receiver CDR will be sensitive with regard to the measured jitter or will be able to tolerate it. In order to do this evaluation, the measured jitter has to be weighted with the inverse bandwidth characteristic of the used CDR.

An example for such a weighting function is shown in Figure 2.45. If the jitter measured on a transmitter is weighted with this function, the harmless low frequency jitter components will contribute less to the overall jitter than the harmful high frequency jitter components [12]. For jitter measurements

using bench equipment, the jitter weighting is achieved by using a golden PLL with the desired bandwidth characteristic in the trigger path.

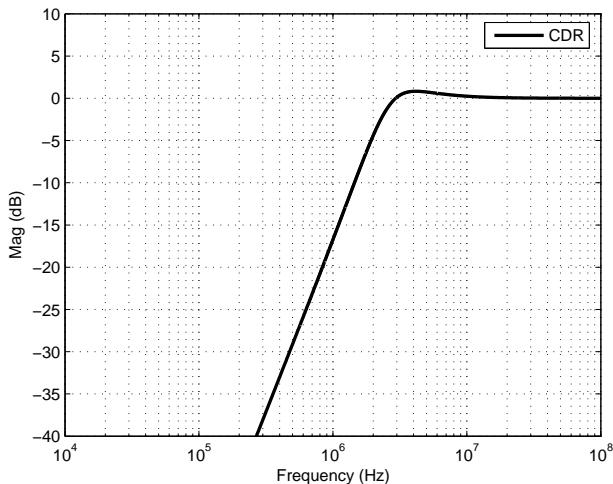


Figure 2.45 Example of an inverse CDR loop bandwidth plot.

One important point to note is that on a digital application the CDR loop bandwidth is not independent of the pattern. The pattern transition density has an effect on the CDR performance. In most designs halving the transition density can reduce the loop bandwidth by half. It is important to understand when looking at a given CDR loop bandwidth specification for which pattern density it was specified [27].

2.6.4 Pre-Emphasis/De-Emphasis and Equalization

The fact that high-speed digital signals are transmitted through a lossy medium like a printed circuit board (PCB) trace presents a challenge when moving to higher data rates. Since the transmission medium loss usually increases with frequency, it has a direct impact on the jitter added to the signal (ISI) and in this way on the BER of the transmission link (more on this topic in Section 8.2). Thus, it becomes critical to compensate for this loss when moving to higher data rates. One option is to improve the transmission medium loss by using lower loss materials and better design strategies but this option implies a higher cost and in some cases cannot solve the problem completely, especially for very high data rates. The other option is to compensate for the transmission medium loss by using equalization on the I/O cell driver and receiver. Basically equalization seeks to either

emphasize the high-frequency components or de-emphasize the low frequency components of the transmitted or received signals [13, 28]. Table 2.1 shows a list of different possible equalization approaches [29]. Note that for each of these equalization approaches there are multiple implementation options.

Table 2.1
List of Equalization Approaches

Implementation	Notes
Continuous time linear equalizer (CTLE)	Used in ATE pin electronics equalization implementations
Transversal FIR	
Rx sampled FIR	Most common approach for transmitter equalizer implementation and sometimes also referred as a feed-forward equalizer (FFE)
Tx sampled FIR	
Decision feedback equalizer (DFE)	Most common approach for receiver equalization

Equalization is an important and vast topic. References [3, 29] provide more detailed information on equalization since the objective of this section is only to provide a very high-level overview. Figure 2.46 shows an implementation of a transmitter equalizer using a FIR filter [30]. In the filter each element T corresponds to a delay of one UI. The input signal propagates through the delay elements of each stage. The input samples are multiplied by the tap coefficients (c_K). For each transmitted bit, the output of the taps are summed to provide the output signal. One challenge with any equalization approach like the one in Figure 2.46 is the choice of the optimal values for each tap. The challenge is that the optimal values will depend on the loss characteristic of the transmission medium.

On the transmitter side, equalization is sometimes referred to as transmitter pre-emphasis/de-emphasis. The transmitter pre-emphasis and de-emphasis techniques try to compensate for the frequency dependent loss of the transmission medium by amplifying the high-frequency components of the signal (pre-emphasis) or attenuating the low frequencies (de-emphasis). The shape of the frequency attenuation or amplification will depend on the exact implementation of the de-emphasis or pre-emphasis algorithm which is usually implemented using a sampled FIR approach. The most common method for pre-emphasis is to set the first bit of any series of equal bit values

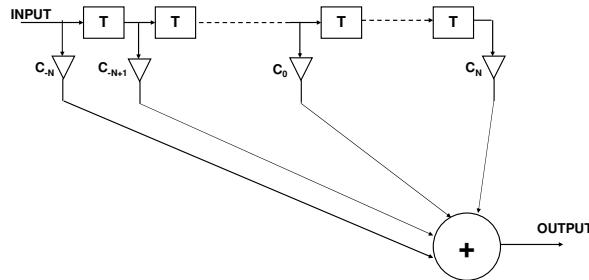


Figure 2.46 Example of a transmitter equalizer implementation using a finite impulse response (FIR) filter.

to a higher voltage level than the rest of the bits as shown in Figure 2.47 (top). In the case of de-emphasis, the typical approach is to set the rest of the bits in a sequence of equal bits to a lower voltage level with the exception of the first one as shown in Figure 2.47 (bottom). Although both methods might cause different design implementations on the driver, with regard to the waveform geometry aspect, they are exactly the same. Figure 2.48 shows a comparison of the data eye at a link partner receiver when the link partner transmitter uses pre-emphasis or not in a lossy signal path. Note that in the pre-emphasis example the signal at the receiver contains a smaller amount of ISI jitter providing a better timing margin for the receiver to correctly strobe the incoming bits.

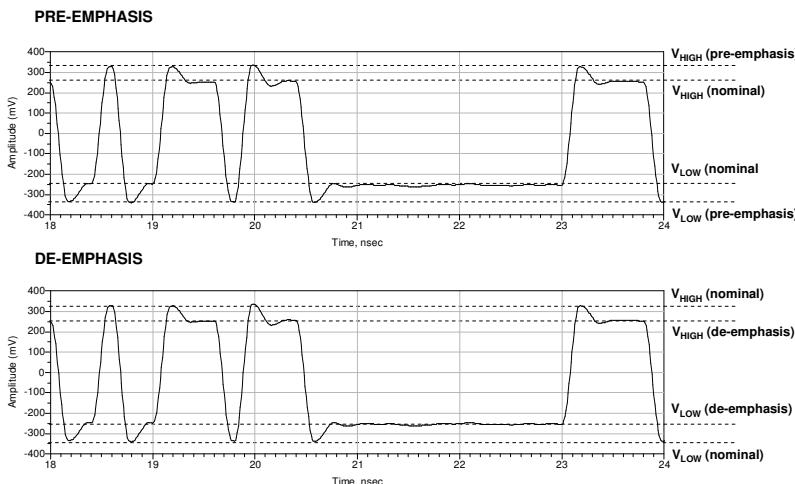


Figure 2.47 Examples of pre-emphasis (top) and de-emphasis (bottom) on a digital signal.

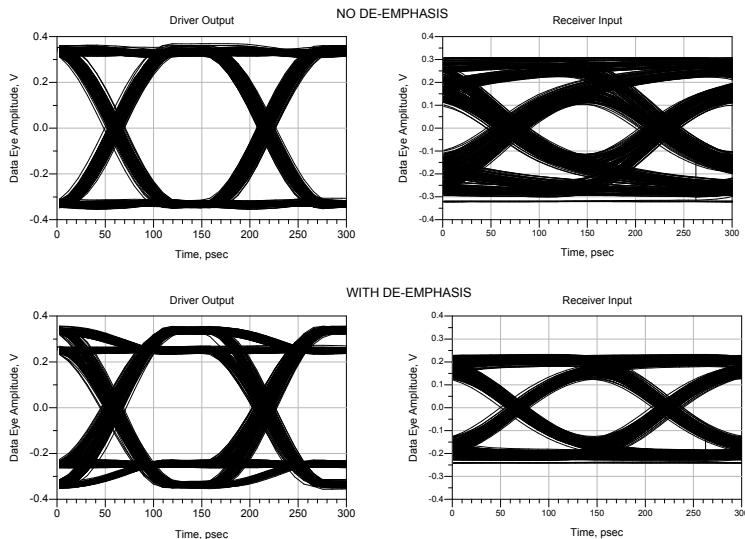


Figure 2.48 Example the improvement on the data eye at the link partner receiver when using pre-emphasis on the driver.

Figure 2.49 shows a high-level diagram of a decision feedback equalizer which is the most common type of receiver equalization implemented in high-speed digital receivers. The equalizer consists of a feed-forward and feedback equalizer implemented as FIR filters. This topology makes the DFE a nonlinear equalizer. Both the feed-forward and feedback equalizers on the DFE will consist of multiple taps that require appropriate values. Those values can be predefined or found automatically by the I/O cell through some training sequence and will depend on the transmission medium loss.

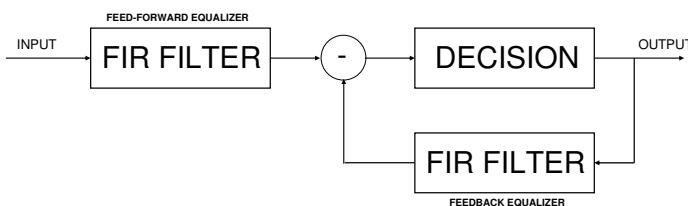


Figure 2.49 High-level block diagram of a decision feedback equalizer.

References

- [1] W. Maichen, *Digital Timing Measurements*. Springer, 2006.
- [2] D. Derickson and M. Mueller, *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization*. Prentice-Hall, 2007.
- [3] S. H. Hall and H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*. John Wiley and Sons, 2009.
- [4] L.-T. Wang, C. E. Stroud, and N. A. Touba, *System on Chip Test Architectures*. Morgan Kaufmann, 2007.
- [5] E. Bogatin, *Signal and Power Integrity Simplified*. Prentice-Hall, 2010.
- [6] H. Johnson and M. Graham, *High-Speed Digital Design*. Addison-Wesley, 1993.
- [7] B. Razavi, *Design of Integrated Circuits for Optical Communications*. McGraw-Hill, 2003.
- [8] J. Redd and C. Lyon, “Spectral Content of NRZ Test Patterns,” *EDN Magazine*, 2004.
- [9] Y. Cai, B. Laquai, and K. Luehman, “Jitter Testing for Gigabit Serial Communication Transceivers,” *IEEE Design and Test of Computers*, 2002.
- [10] J. Moreira, G. Haensel, and F. Koban, “Addressing the Challenges of Implementing an At-Speed Production Test-Cell for 10Gb/s Wafer Probing,” *DesignCon*, 2005.
- [11] M. P. Li, *Design and Test for Multiple Gbps Communication Devices and System*. IEC, 2005.
- [12] M. P. Li, *Jitter, Noise, and Signal Integrity at High-Speed*. Prentice-Hall, 2007.
- [13] Y. Takasaki, *Digital Transmission Design and Jitter Analysis*. Artech House, 1991.
- [14] INCITS, *Fiber Channel – Methodology for Jitter and Signal Quality Specification – MJSQ*, 2004. Technical Report for Information Technology.
- [15] B. Lerner and A. Lowenberger, “Understanding Jitter Requirements of PLL-Based Processors,” *Analog Devices Application Note EE-261*, 2005.
- [16] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, and A. Ivanov, “Jitter Model for the Design and Test of Gbps-Speed Serial Interconnects,” *IEEE Design and Test of Computers*, July 2004.
- [17] Maxim, “Jitter Specifications Made Easy: A Heuristic Discussion of Fibre Channel and Gigabit Ethernet Methods,” *Application Note HFAN-4.3.0*, 2001.
- [18] J. B. Johnson, “Electronic Noise: The First Two Decades,” *IEEE Spectrum*, 1971.
- [19] D. K. C. MacDonald, *Noise and Fluctuations*. Dover, 2006.

- [20] G. Foster, "Comparing DCD and F/2 Jitter," *Synthesis Research SR-TN078 Measurement Brief*, 2008.
- [21] B. Analui, J. F. Buckwalter, and A. Hajimiri, "Data-Dependent Jitter in Serial Communications," *IEEE Transactions on Microwave Theory and Techniques*, Nov. 2005.
- [22] J. Moreira, H. Barnes, M. Howieson, and M. Bromann, "Passive Equalization of DUT Loadboards for High-Speed Digital Applications," *Verigy Users Conference*, 2007.
- [23] B. Laquai and R. Plitschka, "Testing High Speed Serial IO Interfaces Based on Spectral Jitter Decomposition," *IEC DesignCon 2004*, 2004.
- [24] W. F. Egan, *Phase-Lock Basics*. IEEE Wiley Interscience, 2007.
- [25] D. Banerjee, *PLL Performance, Simulation, and Design*. Dog Ear Publishing, 2006.
- [26] B. Razavi, "Challenges in the Design of High-Speed Clock and Data Recovery Circuits," *IEEE Communications Magazine*, Aug. 2002.
- [27] G. Foster, "Clock Recovery Primer, Part I," *IEC DesignCon*, 2008.
- [28] J. Liu and X. Lin, "Equalization in High-Speed Communication Systems," *IEEE Circuits and Systems Magazine*, 2004.
- [29] B. Casper, P. Pupalaikis, and J. Zerbe, "Serial Data Equalization," *IEC DesignCon*, 2007.
- [30] L. W. Couch, *Digital and Analog Communication Systems*. Prentice-Hall, 2006.

3

High-Speed Interface Standards

The objective of this chapter is to provide an overview of some representative high-speed digital interface standards, concentrating on the challenges for testing the physical layer of these interfaces and the functionality in the standards that drive these challenges. Since there are similarities on the electrical (physical) layer between the different high-speed digital standards, some of them are described in detail in this chapter while others are not. Of course the detailed standard descriptions in this book that go beyond the pure description of the physical layers cannot and should not be exhaustive in a sense that they cover all facets of the respective standard. The details that are discussed for these standards are selected with regard to their representative consequences for ATE based testing that are also applicable in a similar form to other interface standards.

One important note regarding the data rate notation used in this chapter and through the entire book is the usage of bits per second versus transfers per second. Some standards define their data rate in terms of transfers per second. The reason is that encoding mechanisms used by those interfaces reduce the number of transmitted bits which contain real data information. In some standards only 80% of the transmitted bits for example contain information since the rest of the bits are used by the encoding protocol. In this context, transfers per second represents the rate of the total number of transmitted bits including those of the encoding protocol and bits per second refers to the rate of transferred bits that contain real information. To avoid confusion we use bits per second within this chapter to refer to the total number of transmitted bits.

3.1 PCI Express

3.1.1 Application Areas

The I/O bus system that connects the core components of a computation system has significant impact on the overall performance of such a system. When the predominantly used conventional PCI bus and its derivatives like the accelerated graphics port (AGP) became a performance bottleneck in computer systems and it turned out that the existing standard could not be enhanced in an economic way to address the immediate requirements with a solid path into the future, the PCI Special Interest Group (PCI-SIG association) started the development of the PCI Express (PCIe) technology.

In its initial implementations, PCIe mainly replaced AGP in computer systems. Over time, PCIe implementations became more pervasive and steadily pushed back conventional PCI in the PC area. In parallel, PCIe expanded into other form factors than just PC plug-in cards such as the ExpressCard form factor. With the definition of the PCIe external cabling standard [1] in 2007, PCIe backplane expansion systems that mainly target industrial and enterprise applications became available.

3.1.2 PCI Express Fundamentals

PCI Express was specified by the PCI-SIG association in 2002. The guiding principle for the development of the PCI Express standard was to design an I/O interface that allowed scalability along the bandwidth needs of the next decade with system manufacturing costs at or below the I/O interfaces used at the time of PCIe standardization.

After the first incarnation of PCIe in 2002 [2, 3], a second generation of the interface was standardized in 2006 [4, 5], and the third generation of PCI Express is in the process of standardization [6] at the writing of this book. Although the driver for a generation roll-over was always increased performance requirements that resulted in data rate changes, each generation also included the adaptation of existing features and the introduction of new functionalities to allow operation at the higher data rates. An overview of the various PCIe generations with a selection of test-relevant features that changed over time is shown in Table 3.1. It is important to note that the column for the third generation of PCIe is preliminary and might change with the finalized specification.

In order to allow a smooth transition from conventional PCI to PCIe interfaces, software compatibility between the two I/O interfaces had to be ensured while considering software handles for the requirements of the future like improved hot-plug support, advanced power management, and

Table 3.1
PCI Express Generations

	Generation 1	Generation 2	Generation 3¹
Release year	2002	2006	2010
Data rate	2.5 Gbps	5 Gbps	8 Gbps
Signal coding	8B/10B	8B/10B	128B/130B
Jitter filter	1-pole highpass	Step bandpass	TBD
Tx equalization	Tx de-emphasis single preset	Tx de-emphasis two presets	Tx de-emphasis, preshoot and boost 11 presets
Rx equalization	None	None	Continuous time linear equalizer and 1-tap DFE

¹ All values stated here are expected only, since third generation of PCI Express was not finalized at the writing of this book.

so on. Maintaining and emulating the existing operational and configuration capabilities of conventional PCI on the one hand and enhancing these capabilities with the requirements of the future on the other hand ensured software compatibility of PCIe with the existing conventional PCI standard. Since configuration parameters are passed from software into hardware on the transaction layer interface of the protocol stack, the levels from this layer down to the physical layer of PCI Express are shown in Figure 3.1.

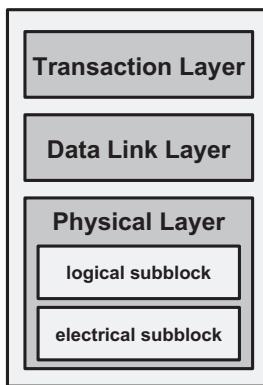


Figure 3.1 PCI Express protocol stack.

The interface from the upper layers to the transaction layer is defined in the PCIe architecture as a superset of conventional PCI. With this approach, software that was based on conventional PCI was also usable with PCIe

hardware initially while additional functionality only usable with PCIe could be defined as well.

The basis for the PCIe architecture is a dual-simplex embedded clock interface that uses differential signaling on point-to-point connections without the need for physical sideband signal connections. An embedded clock architecture was selected for PCIe because it offers excellent data rate scalability that so far allowed the definition of multiple PCIe generations. All of these generations are based on the original PCIe architecture that provides data-transmission capability on standard low-cost PCB material without the need to fundamentally change the underlying signal generation and data recovery mechanisms over a wide data rate range. A basic PCI Express link consists of two differential signal pairs, a transmit pair and a receive pair that establish the dual-simplex connection between two partner devices as shown in Figure 3.2. This combination of a single differential transmitter and a single differential receiver also is referred to as one lane. Data is exchanged via such a lane in a packet-based protocol. The packet structure of PCIe not only considers addressing information for the immediate link partner of a device, but also addressing information for bus segments and devices. With this, the implementation of complex PCIe bus topologies is possible that allow end-to-end communication between devices that do not have a common physical point-to-point connection but are physically separated from each other by one or more PCIe switch devices.

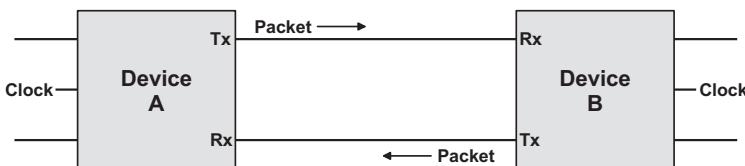


Figure 3.2 Fundamental PCI Express link.

In order to achieve scalability not only on the data rate axis, but also within one data rate class, PCIe makes use of a so-called multilane concept. This means that a data link between devices can consist of multiple lanes. If there is more than one lane that makes up the link, a logical data stream in the transmitting device is split into several substreams, which are serialized separately as shown conceptually in Figure 3.3.

Each of these substreams is then transmitted via a separate lane to the receiving device, which has to reconstruct the original logical data stream after deserializing the single substreams. Since each of the lanes carries its own clock information, multilane embedded clock interfaces can consider relative timing-displacements of data packets on the single lanes after they passed

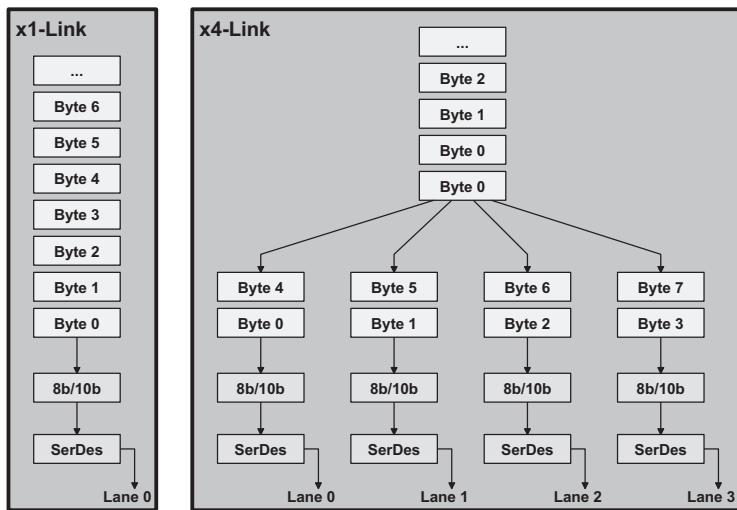


Figure 3.3 Multilane link concept.

the transmission media during the reconstruction of the original logical data stream. The relative displacements between the lanes are determined during link training in the protocol engine when a link is powered up. This allows system and PCB designers to have substantial differences in the physical trace length from differential pair to differential pair. Thus, system and PCB layout is simplified dramatically compared to centrally clocked bus architectures or at-cycle synchronous interfaces (see Section 2.5). Depending on the number of substreams or lanes, the PCI Express standard distinguishes x1, x2, x4, x8, x12, x16, and x32 links.

3.1.3 PCI Express Details

Before data alignment between the lanes of a link is possible, several other steps are executed during the power-up sequence of a PCIe link. The first of these steps is bit and symbol alignment per lane to enable the exchange of usable data between the link partner devices. After basic communication is possible on a per-lane basis, data rate and link width negotiation can take place. The initial training that covers all steps required to get the PCIe link into a fully operational mode is described in more detail in Section 3.1.4.3. Since the lane and link communication during the training is executed on a relatively low protocol level that does not yet include the protocol features that are used to assemble and disassemble the regular communication packets, PCIe contains a communication mechanism that uses low-level packet structures which are directly generated and disassembled on the physical layer. These

low-level packets are called ordered sets and are transmitted simultaneously on all lanes of a link. An overview of the ordered sets defined for PCIe is shown in Table 3.2.

Table 3.2
PCI Express Ordered Sets

Ordered Set	Length [symbols]	Description
TS1	16	Training ordered set used during link initialization
TS2	16	Training ordered set used during link initialization
EIOS	4	Electrical idle ordered set is sent before entering electrical idle state
EIEOS	16	Electrical idle exit ordered set is sent for exiting electrical idle state for links working beyond 2.5 Gbps
SKIP-OS	2-6	Skip ordered set is used to compensate link frequency offsets

Due to the embedded clock architecture that is used for PCIe, regular data transitions are required on the lanes of a link even if no payload data is transmitted. Thus, if no payload data is transmitted, PCIe requires a transmitter to generate idle data that is defined as the data byte 0x00 for PCIe. The 8B/10B coding together with the scrambling used for PCIe ensures that even though the idle data is the zero byte, transitions are generated during the transmission of that data. Of course, due to the signal transitions, the transmission of idle data consumes power in the transmitter as well as the receiver of the link. In order to save power during phases of longer link inactivity, PCIe defines the electrical idle link state. In this state both legs of the differential signal are pulled to the common mode voltage via a high impedance termination. The start of the electrical idle state is recognized by a receiver when it detects an EIOS ordered set. The receiver exits the electrical idle state when a certain differential voltage threshold is exceeded on its input. After electrical idle state, a (potentially shortened) link retraining is required.

Besides the required transition density, 8B/10B coding also delivers DC balanced signals that avoid baseline wander on the transmission media and allow AC coupling of data connections. AC coupling simplifies system design, since device pairs can be used regardless of the common mode voltage used on the legs of their differential signals. This allows combinations of devices that are manufactured in different processes, use different termination schemes,

or use different supply voltages which is especially important for components that are connected via a PCIe cable but reside in subsystems that are physically separated from each other with the PCIe cable being the only connection between the subsystems.

The scrambling (see Appendix D) that is used by PCIe follows the polynomial $G(x) = x^{16} + x^5 + x^4 + x^3 + 1$. It is applied to the data bytes before 8B/10B coding for the transmitters and after 8B/10B decoding for the receivers. In order to minimize electromagnetic interference (EMI) effects, PCIe requires support of spread spectrum clocking (SSC). SSC reduces the magnitude of frequency bins at the reference clock frequency and its derived frequencies in the power spectrum by applying a low frequency phase modulation to the reference clock. More details on SSC are discussed in Section 5.8.3.

3.1.4 PCI Express Protocol

3.1.4.1 Transaction Layer

This layer of the PCI Express protocol stack is responsible for the formatting of the payload data it gets from the higher layers (usually the operating system interface) into so-called transaction layer packets (TLP). A TLP consists of a header and a payload data section. Depending on the packet type, the header contains information about the packet format, address, and routing and a transaction descriptor that includes a transaction ID, traffic class selectors, and other attributes. The packet types supported by the PCI Express transaction layer are memory-, I/O-, and configuration-read/write transactions, as well as message transactions. Besides TLP assembly and disassembly, the transaction layer also is responsible for storing link configuration and link capability information. With regard to traffic prioritization, this layer implements virtual channels and traffic classes that allow isochronous data transfers via a PCI Express link and the usage of a single physical link as multiple logical links.

3.1.4.2 Data Link Layer

The main purpose of the data link layer is to ensure correct communication between two link partners. In the transmit data direction, the data link layer receives TLPs from the transaction layer. It complements these TLPs with a CRC code for error detection and a packet sequence number and hands this packet to the physical layer below which is responsible for physical data transmission. In the receive direction, the data link layer receives packets from the physical layer that were formatted in the same way from the data link layer of the partner device. On these packets, this layer performs an error check on

the CRC code and in case no error occurred passes the received packets in TLP format to the transaction layer in the order given by the sequence number of the received packets. In order to recover from potential errors, the data link layer stores transmitted TLPs for potential retries if the data link layer of the partner device detected an error on a packet and requests retransmission of the data from the faulty packet on. Data retransmission is requested and acknowledged by sending packets that are generated solely within the data link layer. Besides indicating retransmission requests and acknowledgments, these data link layer packets (DLLP) also exchange flow control information like buffer size updates from link partner to link partner. In its function as a link between the transaction layer and the physical layer, the data link layer also conveys information about link states and power state requests between these two layers.

3.1.4.3 Physical Layer

The physical layer of PCI Express architecture consists of a logical subblock and an electrical subblock. The logical subblock defines the logical behavior of a data connection like signaling type, data scrambling, data coding, and the behavior of two devices connected via a PCI Express link over time, whereas the electrical subblock defines the low-level electrical details of a connection such as electrical parameters.

One core component of the logical subblock in the physical layer of the PCI Express architecture is the link training and status state machine (LTSSM). This finite state machine controls the state of a PCIe device from power-up to power-down with all training and configuration steps required to bring the PCIe link into a fully operational mode as well as the transitions between the different link operation modes.

A block diagram of a typical PCIe implementation for the physical layer without the LTSSM is shown in Figure 3.4.

Link Training and Status State Machine

Embedded clock interfaces require that the receiving device synchronizes itself on the incoming data stream. This synchronization usually is done in multiple steps with the first step performing bit synchronization to allow the clock data recovery to sample the data bits with maximum setup/hold time margins. After bit synchronization is achieved, the receiving device identifies the word or symbol boundaries of the incoming data stream that is the basis for later identification of packet frames. With multilane interfaces, the electrical length differences between lane traces making up a link connection between two devices are compensated in the receiving device to allow correct

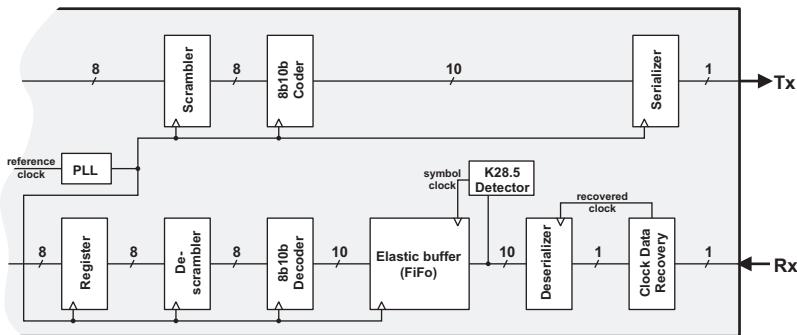


Figure 3.4 PCI Express PHY block diagram.

reconstruction of the data distributed over the lanes for transmission. For PCI Express, one responsibility of the LTSSM is to perform these steps that are fundamental to start up a data link correctly. Besides these synchronization tasks that are common for all embedded clock interfaces, the LTSSM also performs PCI Express specific tasks such as receiver detection or link and lane configuration negotiation. Furthermore, the LTSSM contains states that implement PCI Express specific DfT features and power management. A high-level diagram of the LTSSM with its states and state transitions is shown in Figure 3.5. A short description of these high-level states follows.

Detect In the detect state of the LTSSM, a PCI Express device is checking whether a link partner is connected to its transmitters. Detection is done for each transmitter separately and is the basis for later link width negotiation between two link partners. From electrical idle signal state, a common mode voltage step is applied to the media connected to the transmitters of a link. If a receiver is connected on the far end, the termination of that receiver prevents the media from following that step function quickly, but the media is charged slowly to the final voltage level. If no far end termination is connected, the media follows that step function quickly. A comparator integrated into the transmitter frontend senses the state of the media with some delay to the common mode step applied to the media. The threshold voltage for that comparator is set in a way that it can distinguish the case of a connected receiver on the far end from the case of a floating media.

Polling The polling state of the LTSSM is responsible for establishing bit and word synchronization as well as the data rate negotiation between two connected link partners. The synchronization tasks are achieved by exchanging, analyzing, and adapting to training sequence ordered sets (TS1

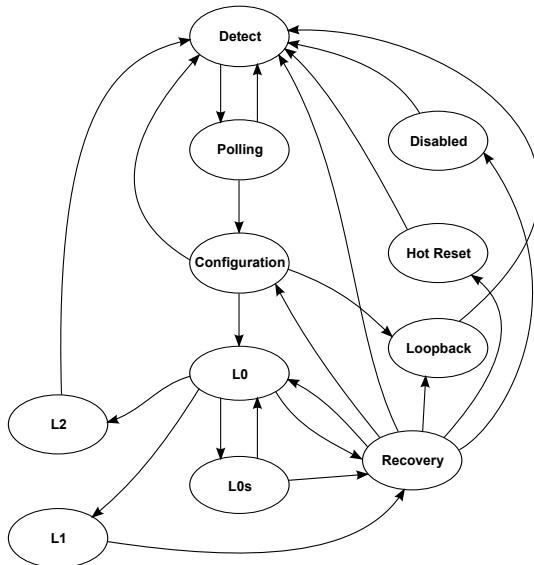


Figure 3.5 PCI Express link training and status state machine. (*From:* [5]. ©2002–2009 PCI-SIG. Reprinted with permission.)

and TS2) sent between two connected devices. In this state, a potential polarity inversion per differential connection is done if PCB layout restrictions force a system designer to connect the positive leg of a transmitter to the negative leg of a receiver.

Configuration This LTSSM state performs lane-to-lane deskew and evaluates the data content of the training sequences that are exchanged between link partners and branches into other LTSSM states (e.g., Disabled, Loopback) if the appropriate bits in the training sequences are set. Besides forced branch requests to other states, the training sequences also contain information about desired link configurations like scrambler control, the minimum number of training sequences required to establish bit and word lock, as well as link and lane numbering. To ensure a consistent link and lane numbering between link partners and to guarantee that an established link fulfills the link width capabilities of both link partners, link and lane numbers are negotiated between the two devices while in configuration state of the LTSSM.

L0, L0s, L1, and L2 After leaving the configuration state of the LTSSM to L0 state, a link between two partners is fully up and running and can be used for packet exchange. If no data exchange takes place, idle data characters are

sent in L0 to keep bit and word lock on the receiver connected to a transmitter. L0s, L1, and L2 are different power saving states that implement three levels of power saving potential with the lowest power saving in L0s and the highest power saving in L2. Depending on the power saving level, resuming into data exchange mode L0 results in different return paths through the LTSSM to L0. These different return paths require different amounts of time to re-establish a fully operational link. The options to transit to L0 from one of the power saving states vary from just sending a certain amount of training sequences for L0s to a whole cycle through the LTSSM starting from detect state for L2.

Recovery Based on the negotiated link and lane numbers, recovery state is reestablishing bit and word synchronization and serves as a branching state to L0, Configuration, Loopback, Hot Reset, and Configuration.

Loopback The LTSSM loopback state establishes loopback paths between the receivers and transmitters of a PCI Express physical layer implementation (PHY). Data patterns stimulated on the Rx pins of a PCI Express device appear with some latency exactly as received on the associated Tx pins of the device.

Hot Reset In this state, transmitters signal their partner devices that a reset action was initiated from a higher protocol level by sending training sequences with the reset bit asserted. Upon receipt of training sequences with the reset bit asserted, a device sends training sequences with the reset bit asserted and enters detect state after a timeout period.

Disabled In disabled state, a link switches to electrical idle signaling after the partner devices have been informed about this by sending out training sequences with the disable link bit asserted.

3.1.5 Electrical Specifications

The electrical specifications of PCIe, like most other embedded clock signal interface standards, are oriented on the data eye openings a transmitter has to ensure and a receiver has to be able to identify in order to guarantee an error-free communication between two partner devices. The difference between the minimum transmitter eye and the minimum receiver eye defines the level and timing budget a system design can use for its connection losses.

The two data eyes nail down the most important level and timing parameters for PCI Express such as minimum differential peak-to-peak

voltages, minimum data eye widths, and maximum timing jitter numbers. The importance of these data eyes is also indirectly contained in the detailed electrical parameters for PCI Express transmitters. For the transition time of a transmitter signal, only a minimum value is specified. There is no maximum specification. The only rule that limits the maximum transition time is the transmitter data eye compliance. Thus, for devices with low jitter, slower transition times are allowed than for devices with higher jitter numbers, since the transition time in combination with timing jitter on the transitions defines the data eye boundaries.

Besides the data eyes, another important parameter group for differential low swing signal interfaces are the DC parameters that define termination impedances of the transmitters and receivers. These impedances have substantial influence on the signal and data eye shape and the correct alignment of the positive and negative signal leg that form a differential signal. In addition to these DC impedance values, PCI Express also specifies parameters for the AC impedance matching in the form of return loss parameters for both transmitters and receivers.

Besides signal transition time, the electrical specification contains timing related parameters that define the allowed bit time or unit interval (UI) ranges as well as the lane-to-lane skew values that have to be met by the transmitters of a link and that have to be tolerated by the receivers of a link. A UI width specification in fact is defining the allowed 300 ppm frequency offset between the nominal 100 MHz reference clock and the actual reference clock frequency. This is due to the fact that the reference clock input of a device directly translates to the same relative change in UI width on a transmitter.

Another set of electrical parameters defines the level values various common mode voltage measurements have to fulfill. The most common of these parameters that also is found in most other differential signal interfaces is the AC peak common mode voltage parameter. PCI Express defines this value for generation 1 devices as rms value that is measured over 10^6 UIs. This differs from the maximum absolute peak-peak values as they are used in most other standard specifications and for PCIe generation 2 devices. Another special parameter in the common mode voltage parameter group of PCI Express is the absolute delta between the common mode voltages of the positive and the negative leg of a differential signal. At a first glance, there is no obvious reason for this parameter because DC balanced signal coding is used and the two legs of the differential signal usually are physically tied to the same current source in the transmitter. Thus, no difference in common mode voltage between the legs should be expected. The underlying physical deviation for this parameter however is duty-cycle distortion for the transmitting clock, which exhibits as a relative common mode shift between

the two legs of a differential signal even if DC balanced data streams are used. Additional electrical parameters defined in the parameter lists of PCI Express are voltage levels for PCI Express specific operation modes like electrical idle and detect or time constants that apply for operation mode changes of the PCI Express frontends.

Two parameters of the PCIe electrical specification list that require special consideration are the de-emphasis specification and the parameters around jitter, PLL bandwidths, and eye openings.

De-Emphasis

In order to compensate expected losses caused by the bandwidth limitation of the PCB connection between two devices, only the first bit after a transition (transition bit) in the data stream is driven at the nominal voltage level. The voltage level of subsequent bits that represent the same logical level as the first bit is attenuated by a specified ratio with respect to the nominal level. Thus, effects that are caused by the low pass characteristic of a typical PCB interconnection are minimized. PCIe defines such a de-emphasis setting for 2.5-Gbps operation and two settings for 5.0-Gbps operation. Since the purpose of de-emphasis is to minimize timing errors at the far end of bandwidth limited connections, the presence of de-emphasized signals at the signal probe points has influence on quite some of the timing parameters that are specified and need to be checked. More details on de-emphasis can be found in Section 2.6.4. In order to have a common basis for the reported timing parameter numbers, the PCIe standard defines procedures to de-embed the de-emphasis effects from the measured signals for the parameters that are influenced by de-emphasis. Details on these procedures can be found in [5].

Jitter

A central topic of discussion during the standardization of the second generation of PCIe was how the jitter specifications have to be defined to ensure proper device interoperability [7]. As a consequence of this discussion, transmitter phase jitter filtering functions were defined that need to be applied to the raw jitter that is measured on the transmitters. The filtering is required because PLLs providing the sampling clock for the CDR on the receiver side can track low jitter frequencies but not high jitter frequencies on the incoming data streams. Thus, high frequency jitter is more likely to cause wrong data latching than low frequency jitter and has to contribute to the specification boundaries to a higher degree than low frequency jitter (also see Section 9.5). The filtering functions were selected as a single-pole highpass filter for the 2.5-Gbps operation and a step bandpass filter for 5.0-Gbps operation. In

combination with the jitter filtering functions, PLL bandwidth and peaking parameters for the transmitter PLLs were introduced because the selected filtering functions only are valid in combination with certain transmitter PLL characteristics.

As an analogy to the bandpass transmitter phase jitter filtering, the jitter parameters that are specified for the 5-Gbps receiver operation are separated for two phase jitter frequency ranges. For each frequency range a separate random jitter component and deterministic jitter component is specified. For the 2.5-Gbps operation, the allowed Rx jitter inherently is defined via a minimum eye width only. Also for the receiver side, PLL bandwidth and peaking values are specified for the PLLs used in the CDR circuit of the receiver.

3.1.6 ATE Test Requirements

If the life cycle of a device is analyzed regarding the usage of ATE equipment in the various test steps that apply, one can see that there are mainly three sections that usually require the involvement of ATE systems with their ability to integrate into an environment that allows gathering a substantial amount of device data efficiently. The first of these steps is the transition from design to manufacturing where ATE equipment is used to support design verification and device characterization in combination with box or bench test equipment. The second life cycle step with ATE equipment usually involved is the production and technology ramp, and the third step is high volume manufacturing of mature devices. All three of these phases pose different requirements on the ATE instrumentation and test methodologies used.

For the design verification and characterization phase, most accurate tests on the complete parameter list are required. In this phase even parameters that can be guaranteed by design are verified directly to ensure that design implementation is correct. For design verification and characterization, ATE based test is complemented by box and bench equipment measurements or system level tests for test items that once verified can be guaranteed by design and which are more efficiently tested in a bench of system environment. For the parameters that are tested on the ATE, test instrumentation has to offer optimum performance regarding DC accuracy using integrated parametric measurement units (PMUs) to be able to accurately characterize DC parameters such as termination impedances and DC common mode voltages. Best AC level and timing accuracy is important to be able to characterize data eye parameters such as differential swing and jitter parameters. For characterizing the de-emphasis behavior of PCI Express, the

most critical instrumentation characteristic is analog bandwidth. Since de-emphasis is used to compensate bandwidth limitations of device connections, test instrumentation must not exhibit similar bandwidth limitations since the effect to be measured would not be visible any more in that case.

Another important instrument requirement is given by the multilane architecture of PCI Express. Multilane links on one hand require accurate relative timing measurements among ATE tester channels in the picoseconds resolution range for Tx lane-to-lane skew measurements, and wide relative timing shifts between these channels in the tens of nanoseconds range on the other hand for the Rx lane-to-lane skew tolerance characterization.

For the production and technology ramp phase of the device cycle, ATE based testflows usually do not test for parameters that are guaranteed by design and that have proven to be stable with enough margin to the allowed value boundaries during characterization. The most important task for PCI Express testing in this phase is to ensure data eye compliance and DC parameter accuracy. Data eye compliance in this regard also means specifically the requirements for jitter measurements and jitter tolerance. Also tracking of other parameters that have exhibited narrow test margins during characterization is important during technology ramp. Since test time is already under tight control in this test phase, only the values of the most critical parameters that were identified during characterization are tracked. The compliance of all other parameters is ensured by guard-banded pass-fail tests. Since the measurement accuracy has to be in the same range as for characterization, requirements for the test equipment do not change substantially compared to the characterization phase. In fact some of the requirements might even be tougher since test time is of more importance due to the higher volume that needs to be tested in this ramping phase. One example of this is the flexibility in termination mode switching ATE instrumentation has to provide. Whereas multiple test programs with different hardware configurations are acceptable for characterization to change ATE termination modes from, for example, differential termination to high impedance termination, ATE equipment has to be capable of switching between these modes within one program on the fly for tests in the technology and device ramp phase.

In the volume-manufacturing phase, the main goal is to find the optimum balance between test cost and test coverage. In order to achieve this balance, DfT approaches such as Tx to Rx loopback configurations (far-end loopback, see Section 6.3.1.3) are used widely. Since most of these DfT approaches are either rather new or lack in parametric variability to really stress device components such as the PCI Express receivers, ATE equipment has to offer capabilities to support the DfT based testing of PCI Express devices. One

example for such a DfT support is to provide instrumentation that allows timing and level parameter variation in a loopback path between a Tx and Rx lane of the PCI Express interface.

3.1.7 Test Support

The LTSSM of PCI Express implements two states that are useful for testing the parametrics and a substantial portion of the logic inside the PCI Express physical frontend (PHY) implementation [8].

The first one is the Polling.Compliance substate of the Polling state. It is entered if a transmitter detects a far-end termination but does not receive a valid differential signal-level on its associated receiver. This behavior indicates that the device is operated in a test environment and the transmitter starts to send a compliance pattern continuously. The content of the compliance pattern is defined to maximize ISI on the lanes and crosstalk between lanes of links with a width beyond x1. Maximum ISI is achieved by using a character with the maximum runlength 8B/10B codes allowed in combination with a character containing the shortest runlength. This leads to a pattern containing an alternating flow of a comma character (K28.5 with maximum runlength 5) and a clock pattern character (D10.2 or D21.5 with maximum runlength 1).

In order to obtain a well-defined compliance pattern, the standard specifies that the first K28.5 character of the K28.5-D21.5-K8.5-D10.2 compliance pattern building block has to have negative disparity. In order to achieve crosstalk maximization, it is required that not all of the lanes within a link generate this pattern synchronously, but that one victim lane stays stable for a certain number of UIs while all other lines operate as aggressors and perform maximum switching. This is achieved with the K28.5-D21.5-K8.5-D10.2 building block by inserting two additional delay characters in one of the lanes before and after this building block. If an overlay of a lane without delay character insertion is done with a lane that contains delay character insertion, one can see that the K28.5 characters with minimum switching activity on one lane falls together with the maximum switching activity characters D21.5 and D10.2 on the other lane. The delay characters are selected as K28.5 characters. A pair of two delay characters is required to keep the data insertion disparity-neutral. The delay character insertion starts at lanes 0, 8, 16, and 24 of a link, continues to shift through all lanes, and starts over on the initial lanes once each lane has inserted a building block with delay characters. With this insertion scheme, each of the lanes serves as victim once while all other lanes operate as aggressors.

The second LTSSM state supporting the test of PCI Express devices is the Loopback state that implements a far-end loopback path between the receivers of a PCI Express PHY with their associated transmitters. This mode of operation is activated by sending a training sequence ordered set (TS1 or TS2) with the loopback bit of the sequence enabled to a device receiver.

3.1.8 Test Challenges

Embedded clock interfaces in general and PCI Express technology in particular exhibit multiple challenges for ATE based testing using general-purpose test instrumentation. One key challenge with this kind of interface that ATE systems have to be capable of handling is nondeterministic data generation of the DUT. With PCI Express there are two notions of nondeterminism in the data streams a device generates [9].

The first notion of nondeterminism that is inherent for all embedded clock interfaces is the variable latency of the transmit data stream. This means that the exact position where a bit/word starts and where a bit/word ends is not predictable with respect to an external frequency reference (which the ATE runs on) and that the timing relationship between an embedded clock data stream and an external reference frequency might change if the reference clock feeding the PCI Express device's PLL is shut down and restarted. In order to deal with this challenge, ATE equipment has to be able to provide a continuously running reference clock and needs to have the capability to adapt its timing system to the timing dictated by the DUT.

One way to address this challenge with an ATE system is to use output dependent timing approaches which identify bit and word boundaries using timing searches and match loop implementations as discussed in Section 5.1. The timing of the ATE is adapted to the timing values found during the searches to align the compare strobe positions to the correct positions with regard to the DUT data stream. A search function on the compare strobe position for the Rx pins is done to identify the center of the data eyes and the loopback latency. After these two variables have been identified, the strobe positions of the ATE comparators are programmed to the center of the data eye with the appropriate latency per lane.

The critical parameters of an ATE guaranteeing success with this approach are the maximum timing range in which the timing programming range, the at-speed match loop capabilities, and the possibility to keep the reference clock of the DUT running during all the searches and timing re-programming. Once the correct strobe positions are identified and set, correct functional tests can be performed on general purpose ATE hardware.

The second notion of nondeterminism is nondeterministic data that occurs on PCI Express devices. Nondeterministic data occurs because embedded clock devices like PCI Express components have to be able to handle frequency offsets between link partners on the one hand and rely on continuously running data streams on the other hand. If two link partners communicate with a slight frequency offset, the slower partner is not able to process all the data at the pace it arrives. Since the data stream is continuously running, the receiving device has to skip some of the incoming data when its input elasticity buffers are full (or close to full). In order to prevent the device from skipping payload data, PCI Express foresees that transmitters have to insert so-called skip ordered sets in specified interval boundaries into the data stream they send. Receiving devices can then skip or add dummy data (SKP symbols) to these skip ordered sets to adapt to a potential frequency offset without loosing payload data.

The problem with skip ordered sets is that these are inserted into the data stream in a nondeterministic manner. It cannot be predicted with the help of simulations or other means when an ATE has to expect such a skip ordered set. Thus, tests requiring real-time comparison to prestored vector patterns are not possible in a normal operation mode of PCI Express if the ATE does not natively support skip ordered set handling. Moreover, undersampling techniques are not possible because the skip ordered sets occur at different positions within the repeated pattern section that is sampled. The only possibility to test a device with such a behavior is to do real-time sampling at the native data rate with postprocessing of the gathered data or to apply ATE hardware that can handle the interface protocol used by the DUT (see Section 9.6). Another methodology to test standard PCIe devices with ATE that does not have such capabilities is to configure the DUT in a far-end loopback configuration that allows the generation of data streams without SKP ordered sets.

A better way to test the functional behavior of PCI Express, however, would be to control skip ordered set generation by means of DfT. There is either the possibility of making the occurrence of skip ordered sets predictable using DfT or to switch off the skip ordered set generation of the DUT completely. Since the ATE provides a well-controlled environment, it is ensured that no frequency offset will occur, and thus, no skip ordered set generation is necessary in the communication with ATE equipment.

The LTSSM poses another challenge on testing PCI Express devices. In order to walk through the state machine in a mission mode manner, handshaking between the DUT and the ATE is required since in such a case, the ATE is considered as a partner device on the link. This, however, requires real-time response calculation of the ATE, which is not feasible with general

purpose ATE equipment. There are several ways to solve this problem.

The first of these solutions performs bit and word lock as described above on the LTSSM Polling.Compliance state. Once bit and word lock on the ATE is achieved, the LTSSM can be executed from Polling.Compliance to L0 by sending the data to the DUT that is expected within a certain time window. This data is independent of the DUT's response as long as the implementation of the LTSSM is correct.

Another approach to address this issue is to provide appropriate DfT that bypasses LTSSM states that are not required in an ATE environment. Since the DUT is well known by the test engineer, handshaking procedures like link width negotiation, link speed negotiation, link and lane numbering negotiation, and polarity inversion are not required in the ATE environment. Since all critical handshaking LTSSM states deal with the uncertainties a device might see in a system environment, it is feasible to bypass these states in the controlled ATE environment by means of DfT in the DUT.

A third option to address this issue would be to implement protocol aware ATE hardware or use a golden device approach. Since protocol testing and the test of electrical low-level parameters, which is the target for ATE based testing, access the device on very different levels, it might be difficult to get appropriate access to the low-level electrical parameters from a higher protocol level for measurement and debug purposes.

The last test challenges to be mentioned here are challenges on the lowest electrical level such as low swing capabilities in the range of less than 30 mV single-ended, analog bandwidth requirements in the range of 4 GHz for 1st generation PCI Express devices and raw data rate capabilities with sufficient signal quality. All of these low-level challenges have to be fulfilled on multiple parallel ATE channels to address the multilane architecture of PCI Express that can accommodate up to 32 differential lanes for a single link.

3.2 HyperTransport¹

3.2.1 Application Areas

The HyperTransport interface is based on an open standard developed by a consortium of different companies [10]. The HyperTransport interface has undergone three major revisions [11]. HyperTransport 1.x and 2.0 are referenced as Gen 1 since they are the same fundamental clocking architecture. HyperTransport 3.x is referred to as Gen 3 since the standard moved from an at-cycle source-synchronous interface to a clock forwarding interface. The main application areas of the HyperTransport interface is

¹Contributed by Michael Comai.

on the front side bus (FSB) of microprocessors for tasks like interchip communication on server systems with multiple microprocessors. One important requirement of any DC coupled HyperTransport interface is that it needs to be backward compatible with previous generations, which means a HyperTransport 3.0 interface must also be able to run in a HyperTransport 1.0 mode (it is not possible for Gen 3 AC to maintain compatibility and this is handled differently).

3.2.2 HyperTransport Protocol

The breakdown of data link layer versus physical layer is not explicitly stated in the HyperTransport specification. Implementations may vary in regard to how these functions are grouped. The delineation in the following sections is the author's interpretation of this separation.

3.2.2.1 Data Link Layer

Link Signals

One HyperTransport link consists of two independent clock forwarded unidirectional interfaces (one in each direction to form a full link). Each side of the link is comprised of clock, control, and data lines. The number of data lines across the link can be 2, 4, 8, 16, or 32. Links that are wider than eight are constructed by ganging multiple eight-bit links together. Eight-bit links that are ganged together are referred to as sublinks and each has a separate clock and control. Asymmetric widths are supported in the TX/RX directions. Gen 1 links only utilize one control signal per link, regardless of width, whereas Gen 3 links utilize one control per 8 bits of the link.

Packet Composition

All packets are multiples of 4 bytes. The control signal is used to differentiate control and data packets. Control packets that do not have an associated data packet can interrupt a data packet from an already in progress transaction and be inserted anywhere on a 4-byte boundary. Since these inserted packets are not allowed to have an associated data packet, this means that there can only be one data transfer in progress on the link at any point in time. The data stream is built by distributing the packets across the available link width.

Sideband Signals

HyperTransport includes a set of sideband signals to facilitate reset, initialization, power management, and interrupt handling. PWROK and RESET# are

the required reset/initialization signals and LDTSTOP# and LDTREQ# are optional power management signals. In systems utilizing power management, LDTSTOP# is used to hibernate all the links on a chain and LDTREQ# is used as an interrupt to request reconnection of the links on the chain. In x86 applications, typically a southbridge would be responsible for waking up the processors, and so the processors themselves would not require the LDTREQ# signal.

CRC Generation and Transmission Error Detection

A periodic window is used for HyperTransport Gen 1 devices to carry CRC information for error detection on the link. The periodic CRC is a 32-bit code and covers 512 bytes of data plus the associated control line(s). For links larger than 8 bits, the CRC code is calculated separately for each sublink. For links smaller than 8 bits, the extra control bits are disregarded in the CRC calculation.

To facilitate error tolerance, Gen 3 devices deploy a retry mode that will disconnect and reconnect the link when an error is detected. This brought about a new mechanism in HyperTransport for calculating CRC codes which integrates more tightly into the protocol called per-packet CRC. As the name implies, CRC codes are calculated and tacked on to the end of each packet. Unlike Gen 1/nonretry mode, each CRC code is calculated across the entire packet regardless of the link width. When retry mode is enabled and per-packet CRC is used, the time window for periodic CRC is still present but its function is used to handle asynchronous clock domain crossings and does not provide any error detection functionality.

Retry

The retry protocol deploys a packet history in the transmitter and an acknowledge mechanism to allow both devices to have knowledge of what packets were sent and received. Upon detection of an error by a receiver, a reconnect handshake is initiated. Figure 3.6 shows the transition from Operational to Training 0 state to reconnect the link. Initially the reconnect sequence bypasses Training 0 or with only a single Training Pattern 0 (Short Retry) and later attempts will wait in Training 0 for a programmable duration specified by the configuration register GlblLinkTrain[FullTOTime] (Long Retry). The number of retry attempts that are Short and the total number of allowed retries before a Sync Flood occurs is also programmable by configuration registers RetryControl[Short Attempts] and LinkTrain[Total Attempts], respectively. The long retry attempts are intended to give the

receiver more time to align to the incoming data before continuing with training.

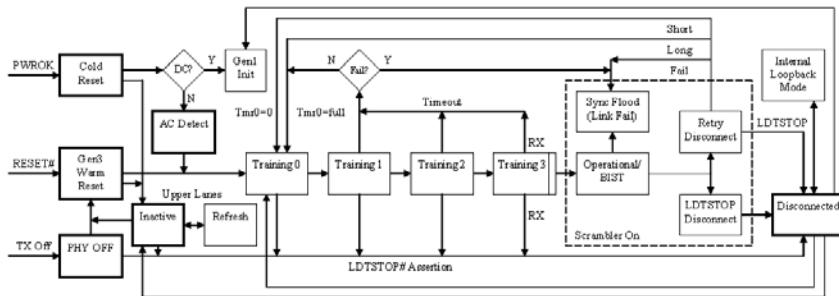


Figure 3.6 HyperTransport state diagram. (From: [12]. ©2001–2006 HyperTransport Technology Consortium. Reprinted with permission.)

3.2.2.2 Physical Layer

Frequency and Reference Clock Modes

Revision 3.1 of the HyperTransport specification supports clock frequencies from 200 MHz to 3.2 GHz as shown in Table 3.3. DC coupled links are required to default to 200 MHz and all other frequencies are optional depending on the application. AC coupled links initialize to 1.2 GHz.

Table 3.3
Clock Frequencies on the HyperTransport Protocol

Revision	Data Rate
1.x	200 MHz–800 MHz
2.0	1.x Rates and 1.0 GHz–1.4 GHz
3.0	2.0 Rates and 1.2 GHz–2.6 GHz
3.1	3.0 Rates and 2.8 GHz–3.2 GHz

There are several different reference clock configurations supported by the HyperTransport standard (Table 3.4). In a microprocessor application, one of the most common configurations is with the reference clock for devices on both sides of the link derived from the same source. HyperTransport refers to this as synchronous. When different references are used, this is referred to as asynchronous. The programmed data rate is also allowed to vary (limited to the maximum supported frequencies of the links). Matched data rates are referred to by default, and asymmetric frequencies are referred to as pseudo-synchronous and arbitrary asynchronous.

Table 3.4
Clocking Modes in the HyperTransport Protocol

Mode	TX/RX Rate	TX/RX Reference Clock
Synchronous	Same ratio	Common
Pseudo-synchronous	Different ratio	Common
Asynchronous	Same ratio	PPM frequency offset
Arbitrary asynchronous	Different ratio	PPM frequency offset

HyperTransport Gen 1 Clocking

HyperTransport Gen 1 was the first installment of the HyperTransport standard. The I/O architecture is an at-cycle source-synchronous interface. The minimum clock frequency for revision 1.0 is 200 MHz which could be increased to 800 MHz. Revision 2.0 added three new speed grades to the HyperTransport protocol: 1.0 GHz, 1.2 GHz, and 1.4 GHz, but maintained the at-cycle source-synchronous clocking.

HyperTransport Gen 3 Clocking

The HyperTransport Gen 3 clocking architecture utilizes a forwarded clock running at bitrate along with a clock data phase alignment circuit in the receiver. This provides similar benefits to an at-cycle source-synchronous architecture without the tight skew tolerances for the higher data rates. One implementation of a HyperTransport 3.0 receiver is discussed in [13] though other architectural implementations are possible.

To understand the HyperTransport Gen 3 clock architecture benefit, it is important to look at the way in which the specification bounds the transmitter and receiver timing characteristics. T_{TX-EYE} specifies the transmitter eye width referenced to the forwarded clock for that transmitter. $BW_{RX-TRACK}$ specifies the minimum tracking bandwidth between the forwarded clock and received data (the transmitter specification also references this in the construction of the transmitter data eye used for the T_{TX-EYE} measurement). Lastly, $T_{CLK-CAD-SKEW}$ bounds the total skew seen by the RX between the clock and the CAD signals.

There are other specifications relevant to the overall timing budget, but these are key to the clock architecture. Let's look at two examples of how jitter is bounded by this specification in end applications. Common mode (CM) jitter is jitter present on both the transmitted clock and data equally that is in phase — this can result from transmit PLL jitter, reference clock jitter, power supply variation, and so on. When interpreted by the specification it is tracked by the receiver from DC to a frequency where the phase shift induced by

$T_{CLK-CAD-SKEW}$ results in the clock and data jitter profile to be out of phase. The frequency that results in the worst common mode jitter performance is in the range of $1/(2 * (T_{CLK-CAD-SKEW} + UI))$. At this point the phase shift is 180° and the common mode jitter at the transmitter is converted to differential mode jitter at the receiver. Figure 3.7 shows conversion factor as a function of the common mode jitter frequency. The skew tolerance is much tighter for the at-cycle source-synchronous HyperTransport Gen 1 clocking so this effect does apply to this earlier generation.

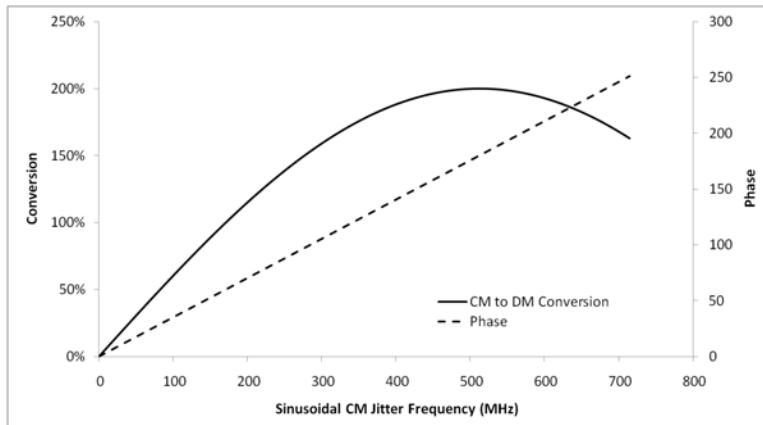


Figure 3.7 Common mode to differential mode jitter conversion for a HyperTransport channel designed for 6.4-Gbps operation with worst case $T_{CLK-CAD-SKEW}$, $T_{CH-CAD-CLK}$, and 3.5 UI of internal receiver skew.

Differential mode (DM) jitter is jitter between clock and data. In addition to the conversion from common mode jitter described above, DM jitter can be a result of ISI on the data line, crosstalk, and temperature variation. The phase alignment circuit in HyperTransport will track this jitter up to $BW_{RX-TRACK}$. In this context, it is assumed that the sources of differential mode jitter required to be tracked by the phase alignment circuit are low frequency so the bandwidth of this tracking loop is relatively low. Higher frequency DM jitter is absorbed by the receiver in the T_{RX-EYE} specification.

An important assumption implied by the transmitter specification is the benefit realized by CM jitter tracking. All of the CM jitter that is seen at the receiver is passed through to the sample clock strobing the data. The T_{TX-EYE} specification does not account for the CM to DM conversion resulting from the jitter profile of the transmitter after it passes through the channel.

Finally, these transmitter specifications were written with a typical receiver implementation utilizing a DLL within the clock recovery circuit. Other architectures are not excluded, however they may lose high frequency

common mode jitter tracking and require additional constraints not included in the specification.

HyperTransport Link Detection

Upon power-up, link initialization starts from the cold reset state. Pin states are evaluated to determine if a link has DC connected signals, and if so the link proceeds to Gen 1 link initialization. If a DC connected link is not detected, then an AC detect is performed.

AC detect is performed by discharging the transmitter differential output by going to the TXGNDTRM state then transitioning to TXACDETECT (a weak pullup to VTX-AC-DETECT). An AC detect comparator in the transmitter monitors the slew rate to determine if an AC coupled link is detected. If no connected link is detected then it is put into the inactive state.

For DC coupled operation, a 16-bit link can drive its CTL [1] transmitter to 0 instead of 1 to signal to the other that it supports unganging, and a 16-bit link can be initialized as two independent 8-bit links.

HyperTransport Gen 1 Link Initialization

Initialization for HyperTransport Gen 1 is fairly straightforward. CTL and CAD signals are transitioned through the states shown in Table 3.5. This sequence is responsible for initializing FIFOs and framing the first packet for operation.

HyperTransport Gen 3 Link Configuration and Training

For DC coupled links, the device will enter 200-MHz Gen 1 mode upon a cold reset event and must configure the link to transition to Gen 3 mode. Link frequency, width, scrambling, retry, and equalization parameters must all be configured to the appropriate settings while operating in the default mode. The device will transition to the new settings upon the next warm reset or LDTSTOP# event. AC coupled links are similar, except that they cannot operate in HyperTransport Gen 1 mode so startup occurs in a 1.2-GHz Gen 3 mode default configuration.

As illustrated in Figure 3.6, the normal progression of HyperTransport Gen 3 training starts by entering the Training 0 state to bring up the DLLs and phase recovery mechanism. From this state the link progresses through Training 1, Training 2, Training 3, and ends in the Operational state. Two timers control how long a link stays in the Training 0 state (T0Time and FullT0Time) to allow the receiver appropriate time to train. When exiting a warm reset, FullT0Time is always used. The link will use a shorter T0Time

Table 3.5
HyperTransport Gen 1 Link Initialization (Source: [12])

CTL	CAD	Duration: 8-, 16- and 32-Bit Links (Bit Times)	Duration: 2-/4-Bit Links (Bit Times)	Notes
0	1	N/A	N/A	Value held during reset
1	1	16 (minimum)	64/32 (minimum)	CTL asserts device-specific time after RESET# deasserts Pattern held at least 16/32/64 bit times after both devices sample assertion of CTL (50 µs of LDT-STOP# if CTL extended)
0	0	512+4N	2048+16N/ 1024+8N	1->0 transition on incoming CTL/CAD initializes load pointer in transmit clock time domain 1->0 transition on incoming CTL/CAD synchronized to core clock and used to initialize unload pointer in receive clock time domain
0	1	4	16/8	0->1 transition on CAD serves to frame incoming packets
1	??	N/A	N/A	0->1 transition on CTL defines start of first control packet and represents first bit time of first CRC window

Note: N can be any integer from 0 to 128

retraining as a result of an LDTSTOP# assertion/deassertion to lower the latency of returning to the Operational state. Training 1-3 are responsible for synchronizing and framing the two sides of the link before entering the Operational state.

Note: as mentioned previously, on a Short Retry event there is a special case where Training 0 is truncated to one transmission of the TP0 pattern or bypassed altogether.

Transmit Equalization

To support the faster data rates, HyperTransport Gen 3.0 specification brought the use of de-emphasis to the high-speed transmitted data stream [12]. De-emphasis creates a high pass filtering effect on the transmitter to equalize

the low pass effects of the channel (see Section 2.6.4). Figure 3.8(a) shows the waveform of a transmitter with de-emphasis enabled. HyperTransport has provisions for two postcursor taps of de-emphasis.

With band limited channels dispersion must also be addressed. This happens when energy of a transmitted bit gets spread across multiple UIs. To account for this there are also provisions for a precursor tap to help further shape the pulse that gets propagated through the channel.

Receive Equalization

For long reach channels, using transmit equalization can have diminishing returns because it effectively reduces the signal power that is pushed down the channel. Receive side equalization is useful in this scenario, providing similar benefit without attenuating the transmitted signal.

Figure 3.8(b) shows the basic concept of a receive side decision feedback equalization (DFE) that is included in HyperTransport. In this approach, the state of the current bit is fed forward to bias the sample threshold for the next bit.

A common implementation of DFE is to unroll the feedback loop, making two comparisons every cycle and the state fed forward simply controls which comparison is used to select the next bit [14]. While this approach is relatively straightforward to implement, it comes with added circuit and power overhead.

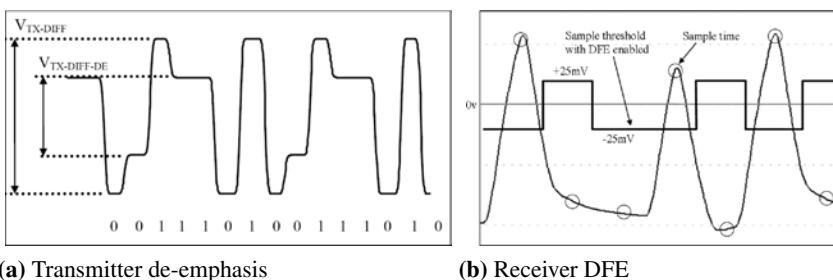


Figure 3.8 (a, b) HyperTransport equalization. (From: [12]. ©2001–2006 HyperTransport Technology Consortium. Reprinted with permission.)

Power States

All power states for HyperTransport are configured through in-band messaging and are delayed until an out-of-band LDTSTOP# or warm reset event puts the link into a disconnected state. In addition to the relatively wide

range of supported frequencies that allow links to scale to the needs of power sensitive platform designs, HyperTransport also includes provisions for managing link width and the power consumption of inactive components (impedance structures and in Gen 3, DLL and phase recovery circuits) to increase power savings. The protocol for managing power states is quite extensive, so we will focus mainly on the requirements that present the most consideration in the operation of a Gen 3 link.

A key factor in the effectiveness of the power management states, besides the obvious power savings, is the length of time it takes to resume from a disconnected state. In HyperTransport this is controlled primarily by the T0Time and FullT0Time counters discussed earlier. Upon resuming from a warm reset or a long LDTSTOP# event, the link is held in the Training 0 state for a time determined by the FullT0Time register. This duration is set to allow sufficient time for the receiver to obtain bitlock to the TP0 pattern before proceeding. The FullT0Time timer must be sufficient to allow lock to be attainable under worst case conditions. T0Time on the other hand sets the duration that a link stays in Training 0 when a link resumes from a short LDTSTOP# event (the threshold between a short/long LDTSTOP# event is set by the idle timer) and this can be optimized based on what other features of a HyperTransport link are enabled (LS2, InLnSt).

Balancing the power that a link consumes in a disconnected state and the latency requirements for resuming from this state can vary from application to application. Systems that require low power with little constraints on latency can afford to be very aggressive in their power management choices. These systems may go to the lowest power states and shut off large portions of the HyperTransport link. In systems where latency becomes a concern—a common example would be a mobile computer playing a video—some power savings are often sacrificed to achieve this. Links may be pulled out of the Disconnected state periodically to keep the DLL and phase recovery mechanisms trained without letting the idle timer expire and less circuitry may be shut down—for example, using LS1 instead of LS2 power state. This will keep the DLLs locked while in the power state which may allow for a longer idle timer or shorter T0Time. These are typical trade-offs made in battery powered applications, whereas servers and desktop computer systems usually have less power constraints and can afford using even less of the power management features so that they may achieve higher performance.

3.2.3 Electrical Specifications

This section discusses the majority of specifications for HyperTransport links. The focus is on the electrical characteristics that must be achieved in order for the physical layer to properly operate.

3.2.3.1 Transmitter

The transmitter is obviously the front end of the physical link. Specifications bound the output characteristics to ensure it can successfully launch a signal into the channel and be received on the other side. Additionally, the transmitter contains circuitry to perform detection and electrical idle to handle AC coupled links. A summary of the specifications that bound the transmitter are shown in Table 3.6. In addition to the transmitter's own signal characteristics that make it to the receiver, these specifications also attempt to account for crosstalk that happens between lanes as well as clock to data characteristics important to Gen 1 and Gen 3 links.

Impedance characteristics of the transmitter contribute to edge rates, eye height, reflections, and power efficiency. Because Gen 1 links have no phase alignment, they are sensitive to slew rate mismatch that can cause skew or jitter between the clock and data. The channel is bound by the skew it can introduce between clock and data, but error from mismatch of parameters like driver impedance (R_{ON}) and pad capacitance (C_{IN}) also eat away at the available margin a link has. This can affect the transmitter slew rates (T_{OR} and T_{OF}) and also impact parameters like the data valid time (T_{CADV}), differential matching (T_{ODIFF}), output differential (V_{OD}), and common mode (V_{OCM}) voltage.

Gen 3 links improve skew constraints with phase recovery, but at the higher data rates the timing budget shrinks. Because of this it is critical to minimize reflections and high frequency impedance discontinuities to avoid reflections. This allows the transmit and receive equalization to operate with mainly the linear channel losses. The high frequency impedance characteristics are bounded in Gen 3 by return loss parameters (RL_{TX*}). Transmitter return loss can contribute significantly to the output data eye characteristics ($V_{TX-DIFF}$ and T_{TX-EYE}). Gen 3 clocking characteristics discussed in Section 3.2.2.2 are also bound so as to minimize phase delay between clock and data ($T_{TX-CLK-CAD}$) as well as to bound the data dependent jitter that cannot be tracked by the phase recovery mechanism ($T_{TX-DJ-DD}$).

AC coupled links also have an additional circuit for AC link detection discussed previously and for electrical idle. These specifications bound the link detect voltage step ($V_{TX-AC-DETECT-STEP}$) and the impedance of the driver that creates this step ($Z_{TX-AC-DETECT}$). During electrical idle, the transmitter

must also control the impedance ($Z_{TX-IDLE}$) and voltage ($V_{TX-IDLE}$) connected to the channel.

Table 3.6
HyperTransport Transmitter Specifications

Parameter	Gen 1	Gen 3 DC	Gen 3 AC
Impedance	R_{ON}	$R_{L_{TX}}$	$Z_{TX-AC-DETECT}$, $Z_{TX-IDLE}$ Z_{TX-GND} , Z_{TX-HIZ}
Slew rate	T_{OR} , T_{OF} , C_{OUT}	T_{TX-TRF}	
Amplitude	V_{OD} , V_{OCM}	$V_{TX-DIFF}$ V_{TX-CM}	$V_{TX-AC-DETECT-STEP}$ $V_{TX-IDLE}$
Timing	T_{CADV} , T_{ODIFF}	T_{TX-EYE} , $T_{TX-DJ-DD}$ $T_{TX-CLK-CAD}$	

3.2.3.2 Receiver

The receiver is bound by specifications to ensure compatibility between the signal launched into the channel by the transmitter and the degradation that results after propagating through the channel.

A summary of the receiver specifications is shown in Table 3.7. As can be seen, many of the same properties are bound on the receiver side. However, there are some receiver characteristics that are important to discuss.

Differential signaling is generally used to take advantage of common mode noise rejection. However, most designs still have some degradation due to common mode noise. As a result, HyperTransport requires a center tap capacitor (C_{CM} and $C_{RX-CM-DC}$) on the receiver termination (R_{TT}). This provides a path to shunt high frequency common mode noise and improves the quality of the signal seen by the receiver.

The other receiver specifications shown in Table 3.7 are the phase recovery characteristics (discussed in Section 3.2.2.2) and impedance characteristics. It is important to highlight the impedance of a receiver in the off state (Z_{RX-OFF})—this is required to bound the AC detect time constant and ensure that a lane in this state does not get detected.

Values for the basic specifications of HyperTransport 3.1 are included in Section 3.2.5. For a complete listing along with complete footnotes for all of the tables reproduced in this section, refer to [12].

Table 3.7
HyperTransport Receiver Specifications

Parameter	Gen 1	Gen 3 DC	Gen 3 AC
Impedance	R_{TT}	$Z_{RL_{RX}}$, $Z_{RX-DC-DETECT}$	$Z_{RX-AC-TRM}$, Z_{RX-OFF}
Slew rate	T_R , T_F , C_{IN}	T_{RX-TRF}	
Amplitude	V_{ID} , V_{ICMAC} C_{CM}	$V_{RX-DIFF}$, V_{RX-CM} $C_{RX-CM-DC}$	
Timing	T_{CADVRS} T_{CADVRH} T_{IDIFF}	T_{RX-EYE} , T_{RX-DJ} $T_{CLK-CAD-SKEW}$ $T_{RX-CLK-TJ-HF}$ $BW_{RX-TRACK}$	

3.2.4 Test Support

Initial HyperTransport Gen 1 specifications did not have provisions for built-in self-test (BIST). DC test support is common (i.e., IEEE 1149.1) and custom solutions have been created. However, it was with the introduction of Gen 3 that a standard solution was included.

The internal loopback mode (ILM) is required for Gen 3 capable devices. This mode is shown in Figure 3.6, and it is important to note that the entry to ILM is directly from the disconnected state. This means that the training sequence for the receiver is the responsibility of the test equipment that is driving it. Section 15.2 of the HyperTransport specification [12] contains the full requirements and entry sequence for this mode.

BIST is also shown as a state in Figure 3.6. This state is shared with the operational state of the link and is optional to support this mode. However, it is useful when using far-end loopback test equipment as well as for system level testing.

Many custom solutions exist. However, one area not covered by the specification and worth noting is support for wafer level testing. In many environments, challenging signal integrity and cost constraints may not allow for high-speed test support. In [14] a custom solution is presented for exercising a significant portion of the interface through a near-end loopback path in the physical layer.

3.2.5 Test Requirements

There is a wide range of testing that is typically done on HyperTransport interfaces across compliance, ATE, and system level test environments.

This testing encompasses interoperability at the system level, electrical characterization, and production screening. With respect to ATE testing, the focus is on electrical characterization across the extremes of fabrication and environment variation. Production screening aims to bound outgoing quality to ensure functionality.

Gen 1 of the HyperTransport interface uses an at-cycle source-synchronous interface, and as such guaranteeing the data valid time of the transmitter and receiver is of critical importance. The timing budget for the these specifications becomes tighter as the data rate increases, and depending on the architecture, the PLL clocking the transmitters can have different characteristics that affect these timing parameters. This makes it important to test all supported rates for the device. Characterization tests should measure these parameters (listed in Tables 3.8 and 3.9). However, it is common to correlate the measured values to oscilloscope or other bench measurements so that production limits can be set appropriately to account for errors introduced by the test setup and still guarantee the devices meet the specification.

Table 3.8
 T_{CADV} Specification (Source: [12])

Parameter	Description	Data Rate	Min	Max	Units
T_{CADV}	Transmitter output	400 Mbps	695	1805	ps
	CAD/CTLOUT valid relative to CLKOUT	600 Mbps	467	1200	ps
		800 Mbps	345	905	ps
		1000 Mbps	280	720	ps
		1200 Mbps	234	600	ps
		1600 Mbps	166	459	ps
		2000 Mbps	183		ps
		2400 Mbps	123		ps
		2800 Mbps	110		ps

The phase recovery mechanism of Gen 3 puts different constraints on testing the transmitter and receiver timing relationships. Table 3.10 lists the transmitter timing specifications that must be tested for both AC and DC coupled links. Low frequency jitter between the clock and data can be adjusted with a high pass filter having a -3 dB corner frequency determined by $BW_{RX-TRACK}$. This relationship poses a challenge for many ATE instruments as discussed later in Section 3.2.6. It is also worth noting that compliance testing of T_{TX-EYE} on a de-emphasized link is measured after processing the data signal with an inverse of the de-emphasis function. In other words, T_{TX-EYE} is measured with no channel ISI. Unless an ATE can de-embed the test fixture and apply this inverse de-emphasis filter to the signal, it must be measured with de-emphasis settings that match the DUT de-emphasis to the

Table 3.9
 T_{CADVRS} & T_{CADVRH} Specification (Source: [12])

Parameter	Description	Data Rate	Min	Units
T_{CADVRS}	Receiver input CADIN valid time to CLKIN	400 Mbps	460	ps
		600 Mbps	312	ps
		800 Mbps	225	ps
		1000 Mbps	194	ps
		1200 Mbps	166	ps
		1600 Mbps	120	ps
		2000 Mbps	92	ps
		2400 Mbps	86	ps
		2800 Mbps	78	ps
T_{CADVRH}	Receiver input CADIN valid time from CLKIN	400 Mbps	460	ps
		600 Mbps	312	ps
		800 Mbps	225	ps
		1000 Mbps	194	ps
		1200 Mbps	166	ps
		1600 Mbps	120	ps
		2000 Mbps	105	ps
		2400 Mbps	86	ps
		2800 Mbps	78	ps

Table 3.10
 T_{TX-EYE} Specification (Source: [12])

Parameter	Description	Min	Units	Comment
T_{TX-EYE}	Cumulative transmitter eye width	0.75	UI	Measured or extrapolated to 10^{-12} BER time shifted by $T_{CH-CAD-CLK}$ and equalized transmit eye. (For 2.4-5.2 Gbps)
$T_{TX-EYE-6.4}$	Cumulative transmitter eye width	0.7	UI	Measured or extrapolated to 10^{-12} BER time shifted by $T_{CH-CAD-CLK}$ and equalized transmit eye. (For 5.6-6.4 Gbps)

Table 3.11
T_{RX-EYE} Specification (Source: [12])

Parameter	Description	Min	Units	Comments
T _{RX-EYE}	Cumulative receive total eye width	0.4	UI	Receiver jitter tolerance. Eye width is measured at the BER of 10^{-12} relative to CLK. (For 2.4-5.2 Gbps)
T _{RX-EYE-6.4}	Cumulative receive total eye width	0.45	UI	Receiver jitter tolerance. Eye width is measured at the BER of 10^{-12} relative to CLK. (For 5.6-6.4 Gbps)

Table 3.12
DLL and Phase Recovery Specification (Source: [12])

Parameter	Description	Min	Max	Units	Comments
T _{RX-CLK-TJ-HF}	High frequency total jitter on CLK	0.03	0.1	UI	Applied to receiver's CLK during jitter tolerance. Represents HF crosstalk in channel to the CLK. Measured with a high pass phase filter with corner frequency of bitrate/10.
BW _{RX-TRACK}	Receivers data phase tracking bandwidth	0.1		MHz	The -3 dB bandwidth for 20 ps peak-to-peak sinusoidal phase modulation applied between CLK and CAD/CTL during compliance testing.

ATE channel loss characteristics or calibrate the measurement to account for the ISI.

The T_{RX-EYE} and clock timing specifications for the HyperTransport Gen 3 receiver are shown in Tables 3.11 and 3.12. This is one of the most involved aspects of designing ATE tests for HyperTransport. Depending on the circuit architecture there may be different sensitivities, but to first order these are dominated by the profile of edge jitter on the clock and data signals. It is not always practical to provide profiles equivalent to a compliance eye. However, there are things a test engineer can tune to get close. Most modern ATE for testing high-speed IO usually includes some form of jitter injection capabilities which allow applying different periodic or other jitter to the device receiver, and tuning the load board to provide a desired amount of ISI to the signal are two considerations that must be investigated early when planning

for HyperTransport Gen 3 testing.

All DC coupled HyperTransport links must support Gen 1, and as such driver impedance must also be tested for this mode. In addition to verifying the impedance specifications (listed in Table 3.13) at any required process-voltage-temperature (PVT) corner, compensation techniques that are used need to be guaranteed across the range of operating corners. If a compensation circuit is used, this requires additional targeted testing in characterization to verify structural integrity. These measurements consist of DC parametric tests.

The levels are the last critical signal parameters of the link that must be tested (common mode and differential mode levels). Input common mode voltage affects the various modes of HyperTransport in similar ways—although the differential input ideally is immune to common mode effects, in practice the gain of the differential amplifier and comparators that receive the signal can vary across common mode voltage. This means that for both Gen 1 and Gen 3 DC, all common mode voltage (AC and DC) is critical both to the driver (V_{OCM} and V_{TX-CM}) and to the receiver (V_{ICM} and V_{RX-CM}). For devices that only support Gen 3 AC, the receiver termination sets the input DC common mode. It is only the AC characteristics that must be tested. However, if the device is tested in a DC coupled environment it must take into account the required DC common mode voltage of the device.

Table 3.13
Impedance Specifications (Source: [12])

Parameter	Description	Min	Typ	Max	Units
R_{TT}	Differential termination	90	100	110	Ohm
R_{ON}	Driver output impedance	45	50	55	Ohm
ΔR_{ON} (pullup)	High drive impedance magnitude change	0		5	%
ΔR_{ON} (pulldown)	Low drive impedance magnitude change	0		5	%

Table 3.14
 V_{ID} Specification (Source: [12])

Parameter	Description	Min	Typ	Max	Units
V_{ID}	Input differential voltage from 2.0 to 2.8 Gbps	200		900	mV
V_{ID}	Input differential voltage from 400 Mbps to 1.6 Gbps	300	600	900	mV
ΔV_{ID}	Change in Vid magnitude	-125		125	mV

Table 3.15
 $V_{RX-DIFF}$ Specifications (Source: [12])

Parameter	Description	Min	Max	Units	Comments
$V_{RX-DIFF-DC}$	Gen3dc cumulative peak-to-peak differential input voltage	170	1800	mV	Applied to Rx pins from a $50\ \Omega$ source. The peak differential voltage is approximated to be 50% of the peak-to-peak.
$V_{RX-DIFF-AC}$	Gen3ac cumulative peak-to-peak differential input voltage	120	1600	mV	Applied to the Rx pins from a $50\ \Omega$ source with DFE disabled. Note transmit de-emphasis must be programmed to ensure max is not exceeded.
$V_{RX-DIFF-RATIO-DC}$	Ratio between the peak differential amplitude between adjacent bits	4	—	mV	Measured over a 1.5 UI. Only applies to CAD and CTL, CLK has a ratio of approx 1.0. Receiver must meet all other input requirements when this occurs.
$V_{RX-DIFF-DFE25}$	Gen3ac cumulative peak-to-peak differential input voltage with DFE set to 25 mV	70	—	mV	Only applies to CAD/CTL, CLK must still meet $V_{RX-DIFF-AC}$. (For 2.4-5.2 Gbps)
$V_{RX-DIFF-DFE25-6.4}$	Gen3ac cumulative peak-to-peak differential input voltage with DFE set to 25 mV	80	—	mV	Only applies to CAD/CTL, CLK must still meet $V_{RX-DIFF-AC}$. (For 5.6-6.4 Gbps)
$V_{RX-DIFF-RATIO}(DFE25)$	Gen3ac ratio differential amplitude between adjacent bits with DFE set to 25 mV between the peak	2.5	7.5	—	Measured on the bit that causes the minimum eye height. Measured over a 1.5 UI interval. (For 2.4-5.2 Gbps)
$V_{RX-DIFF-RATIO}(DFE25-6.4)$	Gen3 ratio differential amplitude between adjacent bits with DFE set to 25 mV between the peak	2.5	6.5	—	Measured on the bit that causes the minimum eye height. Measured over a 1.5 UI interval. (For 5.6-6.4 Gbps)

To the receiver front end arguably the most critical parameter that needs to be tested is the differential voltage sensitivity. The differential voltage specifications for the receiver (shown in Tables 3.14 and 3.15) bound the characteristics required for Gen 1 and Gen 3, respectively. These are relatively straightforward and most ATE equipment has the capabilities to test and characterize the Gen 1 and Gen 3 specifications. In an ATE system, however, the actual data eye that is transmitted by an ATE and the data eye that the device is presented with are often quite different, requiring a thorough understanding of the test fixture characteristics and/or correlation to bench measurements.

Similarly for the transmitter, V_{OD} and $V_{TX-DIFF}$ are to a large extent what dictates the quality of the data eye that is launched into the channel and end up at the receiver. These specifications (shown in Tables 3.16 and 3.17) are the key voltages that need to be tested for the transmitter. Precursor settings intended for Gen 3 AC interfaces are specified for 8 dB ($V_{TX-DIFF-DE8-PRE1}$) and like de-emphasis, fixed settings are specified although other settings are allowed in the register encoding. All settings should be tested for devices that intend to support them.

Table 3.16
 V_{OD} Specification (Source: [12])

Parameter	Description	Min	Typ	Max	Units
V_{OD}	Differential output voltage	400	600	820	mV
V_{ODDE}	De-emphasized differential output voltage for 2.4 Gbps	0.77	0.8	0.83	% Vod
V_{ODDE}	De-emphasized differential output voltage for 2.8 Gbps	0.67	0.7	0.73	% Vod

3.2.6 Test Challenges

HyperTransport has been in existence since 2001 and has test challenges with each of its revisions. Revision 1.x and 2.0 did not contain specific loopback or BIST test mode support to facilitate ATE testing. This makes one of the largest challenges for HyperTransport Gen 1 finding the means to synchronize ATE patterns if no custom support has been designed into the DUT. Despite not being required, some designs have included BIST in their implementation of Gen 1. Gen 3 designs, however, are required to support BIST and ILM for supported Gen 3 frequencies with optional backward compatibility to the Gen 1 frequencies. Requirements to initialize the DUT and prepare it for BIST and ILM are typically implementation specific.

Table 3.17
V_{TX-DIFF} Specification (Source: [12])

Parameter	Description	Min	Max	Units	Comment
V _{TX-DIFF}	Differential peak-to-peak output swing	900	1500	mV	Minimum measured for a lone pulse high or low, maximum measured with zero de-emphasis. (For 2.4-5.2 Gbps)
V _{TX-DIFF-6.4}	Differential peak-to-peak output swing	700	1400	mV	Minimum measured for a lone pulse high or low, maximum measured with zero de-emphasis. (For 5.6-6.4 Gbps)
V _{TX-DIFF-DE3}	Differential postcursor de-emphasized output swing ratio -3 dB	-2.2	-3.8	dB	Measured sending continuous logic 1 or 0 for at least 4 UI.
V _{TX-DIFF-DE6}	Differential postcursor de-emphasized output swing ratio -6 dB	-5.2	-6.8	dB	
V _{TX-DIFF-DE8}	Differential postcursor de-emphasized output swing ratio -8 dB	-7	-9	dB	
V _{TX-DIFF-DE11}	Gen 3 differential postcursor de-emphasized output swing ratio -11 dB	-10	-12	dB	
V _{TX-DIFF-DE8-PRE1}	Gen 3 differential precursor de-emphasized output swing ratio for 1st bit	-7.3	-8.7	dB	Measured the bit before a transition after sending a continuous logic 1 or 0 for at least 3 UI.
V _{TX-DIFF-DE8-PRE2}	Gen 3 differential precursor de-emphasized output swing ratio for 2nd bit	-0.85	-1.35	dB	Measured the bit after the transition from sending a continuous 1 or 0 for at least 3 UI to sending a continuous 0 or 1 for at least 3 UI.

Moving beyond the logical challenge of data patterns, the next challenge of HyperTransport is that not all ATE platforms support source-synchronous strobing of DUT output. HyperTransport is a somewhat unique interface combination with differential signaling and an at-cycle source-synchronous clocking architecture. Without source-synchronous strobing in the ATE, valid time measurements can only be inferred from separate clock and data total jitter measurements. It becomes very difficult to test to the specification. The Gen 3 clocking architecture, adding continuous phase alignment to a clock forwarded interface, also stands out as unique which makes a directly compatible ATE uncommon for testing the HyperTransport Gen 3 transmitter timing characteristics of these interfaces.

For the same reasons that make output timing difficult for HyperTransport, care must be taken with input timings. Because it is a clock forwarded receiver, it is important to reduce sources of uncorrelated jitter between clock and data or be able to de-embed this from the ATE measurements. Depending on ATE instrumentation, artifacts in timing generation between clock and data may contribute to error in the RX eye being seen by the device. Some of these measurement errors can be addressed by good test fixture design practices (minimizing trace length, consolidating CAD/CTL lines onto the same instrument with their associated CLK).

3.3 XDR DRAM

3.3.1 Application Areas

The target application areas for extreme data rate (XDR) DRAM devices are multicore processing systems, gaming, and graphics as well as consumer applications that require high memory bandwidth with limited data connection widths like HDTV devices. The usage of XDR DRAM devices in the mass market is closely linked to the Cell processor [15] and derivatives of the Cell architecture [16] that use XDR DRAM as main memory. Thus, XDR DRAM devices can be found in server blades that use Cell processor versions with XDR memory interface and consumer products that use Cell-based high-definition (HD) graphic processors. The most prominent representative of systems that use XDR DRAM components is the Sony PlayStation 3 game console with its Cell processor CPU.

3.3.2 XDR Fundamentals

The XDR technology has been developed by Rambus, Inc., and was introduced into the market in 2003. The XDR architecture consists of three

core building blocks. These building blocks are XDR DRAM, XDR I/O (XIO), and the XDR memory controller (XMC). While XIO and XMC are a collection of hard and soft macros that can be licensed from Rambus, Inc., to be used in proprietary IC designs, XDR DRAM devices are complete DRAM components that use XIO as electrical interconnects and deploy a communication protocol defined by Rambus [17, 18]. XDR DRAM devices are manufactured under license by DRAM memory manufacturers.

The XDR technology represents a mixture between the traditional memory interface architecture with a parallel data bus and serial high-speed interfaces with a packet based serial connection to transmit command and address data. The packets that contain command and address data are named requests in the XIO terminology. In contrast to traditional single-ended multi-drop memory data bus implementations, XDR uses differential point-to-point connections for its data bus. These point-to-point connections are not operated in a semiduplex manner to achieve a bidirectional data flow as is the case for the differential interfaces described so far. XDR in fact exchanges data on the data bus in a full-duplex way. This combination of a bidirectional differential point-to-point connection that uses a full-duplex data flow gives the XIO interface of XDR a unique position among other high-speed I/O interfaces and also opens up some interesting aspects from an ATE test point of view. In order to cover a wide application range with changing requirements for data bus width and memory depth, XDR memory devices have a programmable data width. This means that the number of data signals that form the data bus can be reduced from the maximum native width a device is designed for to a lower number. The data bus width reduction goes hand in hand with a column address range expansion which is reflecting an increase of memory depth. The additional column address bits that are used for this column address expansion are called subcolumn (SC) bits. The amount of decoded SC address bits depends on the programmed data width reduction. An example for the relationship between programmed data width and decoded SC bits is shown in Table 3.18 for devices with native data bus widths of 32 and 16 bits.

XDR DRAM devices transfer 8 data bits with one master clock cycle in an octal data rate (ODR) manner on each of their DQ signals. In order to simplify the electrical design of systems that use XDR memory devices, a mechanism for static skew compensation between the single data bus signals is provided by XDR. This deskew methodology called FlexPhase allows the adjustment of the device DQ data stimulus and capture times individually per pin to compensate for different electrical lengths of the DQ signal paths. The FlexPhase adjustment happens during a calibration step in the initialization sequence for a XDR DRAM device.

Table 3.18

XDR Subcolumn Addressing and Used Data Width (Source: [17])

Native Device Width	Programmed Device Width	Decoded SC Bits	Fraction of Selected Column Amplifiers
x32	x32	None	All
	x16	SC[4]	1/2
	x8	SC[4:3]	1/4
	x4	SC[4:2]	1/8
	x2	SC[4:1]	1/16
	x1	SC[4:0]	1/32
x16	x16	None	All
	x8	SC[3]	1/2
	x4	SC[3:2]	1/4
	x2	SC[3:1]	1/8
	x1	SC[3:0]	1/16

3.3.3 XDR DRAM Details

XDR Timing Architecture

XDR devices present themselves to the memory controller via two separate interfaces. Each of these interfaces comes with its own clock signal that is routed from the memory controller to the XDR device. The first of these interfaces is a low-speed serial interface that is used to initialize and configure the XDR devices. One part of the device initialization that is supported via this serial interface is the FlexPhase calibration. The other interface is the high-speed interface that transfers commands (or in XDR terminology request packets) from the memory controller to the XDR device and transfers data in both directions between memory controller and XDR memory device. The pins forming these interfaces are listed in Table 3.19.

Table 3.19

XDR Pin List

Signal Pins	Type	Group
CFM, CFMN	I	Clock
RST	I	Serial interface
CMD	I	Serial interface
SCK, SDI	I	Serial interface
SDO	O	Serial interface
RQ0..RQ11	I	Request
DQ0..DQ15	IO	Data
DQN0..DQN15	IO	Data

Clock Pin Group The differential clock from master (CFM) signal serves as timing reference for the request and data pins of the XIO interface of XDR memories. The active transition of the clock that serves as reference is the falling edge of the differential clock signal.

Serial Interface Pin Group The serial interface pins are used to control basic functions of the XDR memory such as initialization and control register programming. Whereas the SCK, RST, and CMD pins are routed in parallel to all memory devices that form a XDR DRAM system, the SDI and SDO signals connect the single memory devices in a daisy-chain manner.

Request Pin Group The XDR memory device receives its operation commands on the pins of the request pin group. The request pins run at the data rate of the clock pins. Request commands are transmitted as request packets consisting of two consecutive bits for each pin of the request pin group. The first of these bits is latched into the memory device with the falling edge of the differential CFM clock. The second bit is latched with the rising edge of that clock.

Data Pin Group The data on the differential data pins of XDR are transmitted in an octal data rate manner with eight data bits per CFM clock cycle.

A schematic block diagram describing the XDR clocking architecture is shown in Figure 3.9. For the low-speed serial interface, the RST, CMD, and SDI input signals are latched into the device with the falling edge of the SCK clock signal. The SDO signal is generated with a specified minimum setup and hold time around the falling edge of SCK.

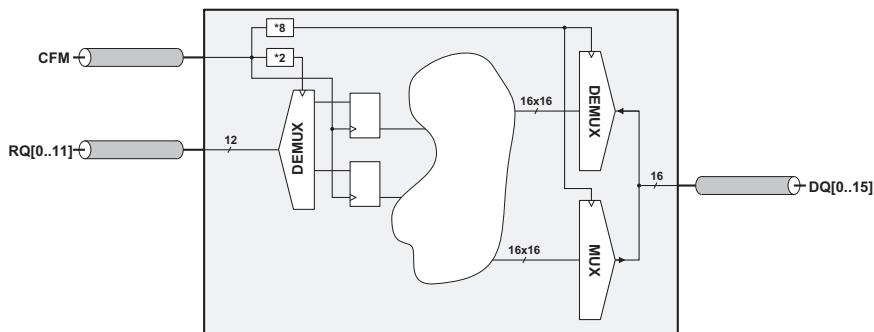


Figure 3.9 XDR timing architecture.

On the high-speed interface, the frequency of the differential CFM clock signal is multiplied in the XDR memory device by a factor of two and by a factor of eight. The multiplied internal clock signals are used to oversample the request-signals with two sampling points per CFM clock cycle. The DQ-signals are oversampled with 8 sampling points per CFM clock cycle for incoming data. For DQ data generated by the XDR memory, 8 data bits are sent out per CFM clock cycle. The oversampling of RQ and DQ signals happens in demultiplex circuits so that the nonmultiplied CFM clock signal can be used for further data processing after the demultiplexing. For the DQ signals that are generated by the XDR memory device, the data internally is clocked with the nonmultiplied CFM signal. Before the data is transmitted via the IO drivers, the data is multiplexed to 8 times the internal data rate. As on the low-speed interface, also for the high-speed interface, the RQ and DQ signals are sampled and generated with the falling CFM clock edge.

XDR Specific Functionality

The mechanism of scalable bus width per XDR device is applied for XDR-based DIMM memory modules to provide flexibility for the population of DIMM slots that are associated to a single x32 DQ channel. Due to the point-to-point topology of XDR, it is not possible to increase the storage amount served by such a DQ channel by just using a second DIMM that is daisy-chained to the already populated DIMM slots on the data lines. Instead, a topology called dynamic point-to-point (DPP) configuration was developed for XDR-based DIMMs. For a DPP base configuration that populates the first of two DIMM slots that are shared for one x32 DQ channel with an XDR DIMM, the second DIMM slot is occupied with a so-called continuity module. This continuity module just loops back half of the DQ lines to the memory controller. In case a second DIMM is installed, the continuity module is replaced by that DIMM. The memory controller is able to recognize the second DIMM and configures the data width of the XDR components on both DIMMs into half of their native device width. The DQ lines that previously were looped back by the continuity module from the first DIMM to the memory controller now are directly connected to the DIMM in the second slot.

Another specific functionality of XDR memory devices to be discussed here is dynamic request scheduling. For some requests transmitted to XDR memory devices the memory controller is allowed to specify a delay for the execution of that request. This delay control allows the memory controller to transmit the requests in a way that makes best use of the available bandwidth on the request bus while also keeping the amount of data transmitted within a

certain period on the data bus at an optimum without avoidable gaps between data packets.

A feature that also targets optimized usage of the data bus bandwidth is the segmentation of the memory device into so-called bank sets. The memory banks within different bank sets are served by internally separated data paths. This allows a quasi-parallel memory core access on the memory cells located in different bank sets. From an I/O perspective, this parallel access manifests itself in reduced gaps between data packets on the data bus if the data is associated to different bank sets. One specific XDR feature that uses this advantage of bank sets is “Early Read After Write” (ERAW). If the target bank set of a write request is different from the target bank set of a following read request, the write-to-read turnaround bubble on the data bus can be reduced compared to a single bank set operation. This is due to the fact that internally to the XDR device, the read operation on one bank set can already start on the local data bus of that bank set while the write operation on the other bank set still occupies the local data bus of that bank set. Together with the bank set concept, simultaneous activation and simultaneous precharge for the different bank sets might be supported.

The last XDR-specific feature to be discussed here is the access to the XDR memory device via the low-speed serial interface. This interface is mainly used to initialize the XDR devices and to set the configuration registers of a device. The topology of the low-speed serial interface for a XDR memory system is shown in Figure 3.10. While the RST, CMD, and SCK signals are shared between all devices, the SDI and SDO pins form a daisy-chain that ends on the SRD input pin of the memory controller. The SDI input of the first device in this chain is connected to the VTERM supply through a termination component (as is the case also for SCK, RST, and CMD). Since all logic in the XDR environment is low-active, the VTERM voltage at the SDI input is interpreted as a logical low by the XDR device. During device initialization, each device in the daisy-chain is assigned a unique serial identification value. This ID value then allows access to specific devices within the chain via the shared control signals of the low-speed serial interface.

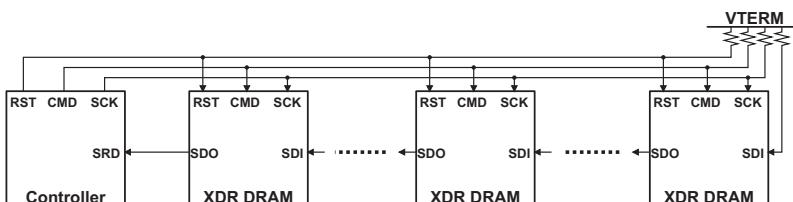


Figure 3.10 XDR serial interface topology.

3.3.4 XDR Protocol

Obviously, the low-speed serial interface and the high-speed IO interface of XDR deploy different protocols for the communication between memory controller and memory device. Although the low-speed interface should not pose any significant challenges from a testing perspective, we also will describe the data link layer of this interface because it plays a crucial role during the FlexPhase calibration that is described in the physical layer section.

3.3.4.1 Data Link Layer

For the XDR high-speed IO interface, a basic access is structured as shown in Figure 3.11. A request packet consisting of two consecutive bits on the request bus is transmitted with the first of the two bits center-aligned to the falling edge of the CFM clock. Depending on the command contained in the request packet, data on the data bus might occur with a defined latency to the start of the request. For some requests, no data transmission happens on the data bus. If the request triggers a data transmission, the direction of the data flow is determined by the request type (write or read request).

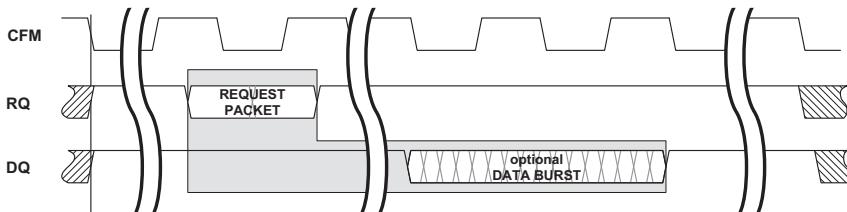


Figure 3.11 Basic access on the XDR high-speed interface.

A request packet contains operation code, address, and other control information for the memory device. XDR defines five types of request packets that are listed with the packet fields transmitted within each packet in Table 3.20.

The low-speed serial interface follows a serial protocol with a separately transmitted clock signal. During device initialization, each of the devices in the daisy-chain gets assigned a unique serial device ID (SID). Since the devices cannot be identified individually at power-up, the SID assignment cannot be done via command accesses on the shared CMD pin but has to follow a special protocol. After powering up the devices in the daisy-chain, the SCK pin starts clocking. Initially the RST pin is held low. The SDI input of the first device in the chain is connected to the termination voltage and thus sees a logical zero (negative logic) during the entire initialization sequence.

Table 3.20
XDR Request Packet Types

Request Packet	Packet Description	Packet Fields	Field Description
ROWA	Activation command	OP[3..0] DELA BA[2..0] R[11..0] SR[1..0]	Operation code that defines the command in more detail Delay field for dynamic request scheduling Bank address Row address Subrow address
COL	Read/write command	OP[3..0] WRX DELС BC[2..0] C[9..4] SC[3..0]	Operation code that defines the command in more detail Selects read or write command Delay field for dynamic request scheduling Bank address Column address Subcolumn address for dynamic width
COLM	Write mask command	OP[3..0] BC[2..0] C[9..4] SC[3..0] M[3..0]	Operation code that defines the command in more detail Bank address Column address Subcolumn address for dynamic width Mask data
ROWP	Precharge/refresh command	OP[3..0] POP[2..0] BP[2..0] ROP[2..0] RA[7..0]	Operation code that defines the command in more detail Delay field for dynamic request scheduling Bank address Specifies refresh command in more detail Refresh address
COLX	Remaining commands (e.g., calibration, power-down ...)	OP[3..0] XOP[3..0]	Operation code that defines the command in more detail Extended operation code

The SDO outputs of all devices generate a logical one when the RST pin goes high after its initial low-state. With the transition of the RST pin to low, the first device in the chain samples the logical zero on its SDI pin and sets its SDO output pin to the state sampled on the SDI pin with a delay of one SCK clock cycle. Thus, the logical zero initially sampled by the first device in the chain with the first low-state of the RST pin ripples along the device daisy-chain with a delay of one SCK cycle per device. Each device contains a state machine that counts the SCK clock cycles between the RST pin transitioning to low and the SDI pin of that device sampling a zero for the first time. The value of this counter delivers a unique value for each device in the chain after the memory controller sees the low-state arriving on its SRD pin and is stored as SID in each device.

After this initialization, the memory controller can apply the standard protocol for the low-speed serial interface. This standard protocol initiates write and read transactions from the memory controller to the memory devices via the shared CMD pin and receives the data for read transactions from the memory devices via the serial SDI/SDO daisy chain. The structure of such a basic access via the low-speed interface is shown in Figure 3.12.

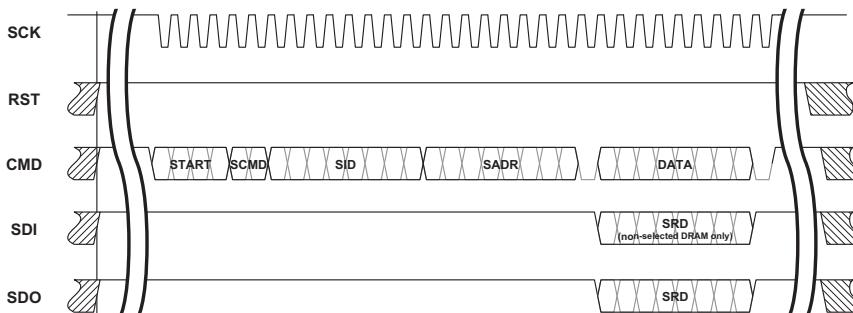


Figure 3.12 Basic access on the XDR low-speed interface.

After a START field that indicates the start of a transaction, the SCMD field defines the selected transaction type by a 2-bit opcode. The following serial byte defines the SID of the device in the chain the transaction is aimed at. Only the lower 6 bits of the byte contain the SID address. The upper 2 bits always are set to zero. The following serial byte SADR defines the address of the configuration register to be accessed with the transaction. After this address field, the serial data byte SWD is transmitted framed by two single zeros. For read transactions, the SWD byte is set to all zeros and the addressed device sends the requested SRD data byte on its SDO output. This data byte then ripples along the daisy chain toward the memory controller. The

commands supported by this serial protocol are listed in Table 3.21. From a testing perspective, the serial forced read (SFR) and serial broadcast write (SBW) are important commands of the low-speed interface. These commands allow access to the DUT via the low-speed serial interface without knowing the serial ID of the device. Since in ATE test applications the devices typically are not configured in a daisy-chain manner, but are treated as single devices per test-site, the SID of the device does not have any functionality impact and just can be ignored by the ATE using the SFR and SBW commands.

Table 3.21
XDR Serial Commands (Source: [19])

Opcode Command Description		
00	SDW	Serial device write - device addressed by SID is written
01	SBW	Serial broadcast write - all devices are written
10	SDR	Serial device read - device addressed by SID is read
11	SFR	Serial forced read - all devices are read

3.3.4.2 Physical Layer

On its physical layer, XDR uses Rambus Signaling Levels (RSL) on unidirectional pins and Differential Rambus Signaling Levels (DRSL) on bi-directional pins. The only exceptions are the SDO pin which uses CMOS signaling and the differential clock inputs that use differential signaling with larger minimum swing than DRSL.

RSL is the single ended signaling implementation that already was used for RDRAM memory devices [20]. RSL is aimed at shared configurations with one pin (e.g., of the memory controller for the single request pins) driving the signal that is used by multiple receiving devices. After the last device in the sharing chain, the signal is terminated via a termination impedance to a termination voltage. Such a termination topology also is called fly-by termination. An example for a fly-by termination is shown in Figure 3.10 for the RST, CMD, and SCK pins of the low-speed serial interface. The termination impedance at the far end of the chain is defined to be 28Ω for RSL signaling. The RSL input pins of the XDR devices are high impedance inputs. RSL signaling is based on the open drain methodology. RSL drivers only draw current if a low level (which is interpreted as logical high) is transmitted. If no data transmission takes place or a physical high level is transmitted, the signal line is pulled up to the termination voltage via the termination resistance at the end of the line. If XDR devices are operated in an ATE test environment with only a single RSL input connected to the ATE driver, a termination on

the test fixture usually is not required because the wave that is reflected due to the impedance mismatch at the DUT input is fully absorbed by the $50\ \Omega$ source impedance of the ATE driver and does not impact other devices.

DRSL signaling that is used on the data pins of XDR memory devices is an enhancement of the RSL signaling that covers differential signaling. The use of differential signals improves noise immunity of the transmitted data and allows the use of low swing signals to reduce current consumption at increased data rates over RSL. Since DRSL is used for point-to-point connections only, on-die termination (ODT) is used instead of fly-by termination. The structure of a DRSL link and its signal transmission is shown in Figure 3.13.

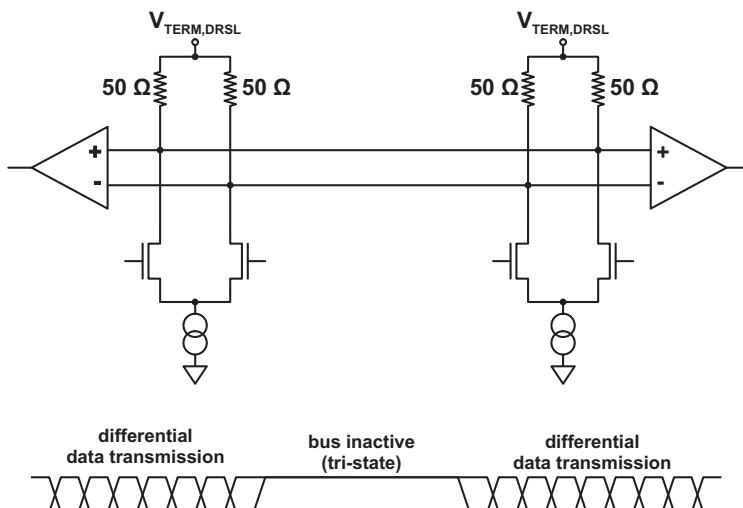


Figure 3.13 DRSL link structure and data transmission.

DRSL uses bidirectional data exchange on the differential signal connection. Due to the bidirectional connection, signal transmission on a low level is fundamentally different from other unidirectional differential interfaces. The first difference is that the differential receivers deployed in a XIO frontend have to be implemented using high impedance inputs instead of a floating or center-tap termination [21]. The second difference is that the termination on the receiving partner of a XIO link is provided by the termination resistances and termination voltage of the differential driver. As a consequence, the current that flows through the differential connection is not originating from the differential driver that stimulates the signal, but from the far-end termination. This is due to the fact that both link partners use the same termination voltage $V_{TERM,DRSL}$. Thus, there is no current flow on one of the differential legs and a current flow from the far-end termination

into the current sink of the stimulating driver on the other differential leg. During periods of inactivity on the signal connection (e.g., for bus turnaround bubbles), the termination voltages pull both legs of the differential connection to $V_{TERM,DRSL}$.

Device Calibration

In order to guarantee the specifications required by DRSL, XDR DRAM components might support specific calibration transactions. These transactions allow the calibration of the on-chip termination impedances and the DQ drive currents to achieve optimal signal performance. The details of these calibrations usually are not disclosed by DRAM vendors. Thus, from a XDR testing perspective, it is sufficient to apply the required and supported calibration requests to calibrate impedance and current settings in the DUT.

The XDR architecture also defines calibration steps for the memory controller that apply in the same way to the ATE system in a XDR testing environment. Besides the impedance and current calibration, these steps also contain timing calibration. While current and impedance calibration are taken care of by the ATE system calibration, the timing calibration needs to be performed on a per-DUT basis because here the calibration results interact with the timing behavior of the DUT. In a XIO memory controller the timing calibration is covered by the FlexPhase feature.

FlexPhase allows the memory controller to adapt its timing on the DQ data pins separately for the drive data and the receive data. This timing adaptation compensates static skew components between the single DQ signal connections that for example are caused by PCB trace length differences, differences in package traces and bond wires between the pins and driver/receiver mismatches. Timing training in a first step adjusts the receiver sampling phase in the memory controller (RX TCAL) and in a second step the driver delay for each DQ pin of the memory controller. The timing adjustment happens in a way that after the training the receiver sampling phase and the driver delays are set to their optimum so that the DQ data in the memory controller and the memory device are sampled in the center of the respective data eye. In an ATE environment, the ATE system has to adjust its driver and compare strobe timing in a similar way as is the case for the memory controller.

Since there is no reliable data exchange possible via the DQ pins before the training, the data required for the RX TCAL is written to the XDR device via the low-speed serial interface using the write data serial load (WDSL) register. Reliable data communication on the low-speed serial interface is possible because the timing conditions required on this interface are relaxed

enough to be guaranteed by design. Due to the fact that the amount of data required for the receiver phase calibration is relatively small, writing this data to the memory via the low-speed interface is feasible and does not have significant performance impact. After the data is loaded serially into the memory device, it is read at-speed via the data bus. Since the memory controller (or ATE) knows which data to expect, it can use this knowledge to adjust its timing as required to sample the single DQ signals in the center of their respective data eye.

After a successful RX TCAL calibration step, the driver timing calibration is done solely via the DQ bus because it is guaranteed that the read direction already has valid timing settings. Thus, mismatches between the data written from the memory controller to the memory device and the data read from the memory device stem from incorrect timing settings for the driver in the memory controller. During driver calibration, these timing settings then are varied to find the optimum setup.

3.3.5 Electrical Specifications

Level Specifications

XDR uses a supply voltage of 1.8 V. In contrast to true open collector or pseudo-open drain signals (see Section 3.4.4.2), the termination voltages $V_{TERM,RSL}$ for the RSL signals and $V_{TERM,DRSL}$ for the DRSL signals are not equal to the supply voltage, but are either provided via a separate pin for the RSL signals or generated at a lower level than VDD for the DRSL signals. Due to the nature of RSL and DRSL signal transmission, these termination voltages are identical to the high level of the respective signal voltages. While the minimum level swing for RSL, CMOS, and differential clock signals are specified well beyond 100 mV, the specified minimum swing of 50 mV for the DRSL signals is relatively small and requires special care for testing these device pins.

Timing Specifications

The timing specification parameters that are defined for XDR in general follow the known parameters of parallel bus interfaces. Of course, the most tight timing specifications for XDR are related to the high-speed DQ pins. Besides the skew between DQ pins, the most stringent parameters are the DQ setup and hold times for the DQ receive and the output delay variation over the consecutive bits of a single DQ pin. These parameters in fact define the data eye width of the DQ bus with respect to the CFM clock.

3.3.6 ATE Test Requirements

The specific requirements of ATE for the test of XDR memory devices mainly stem from the new features and interface implementations XDR deploys. The first feature to be mentioned here is the timing calibration of the DRSL interface. To enable the economical test of XDR memory devices, the ATE system has to provide efficient means to support the timing calibration required by XDR. The ATE features used to perform timing calibration must provide a fast identification of the individual per-pin data eye timing and a per-pin readjustment of the ATE timing to the requirements of the DUT. Moreover, these timing features not only have to be fast if they are executed only on a few pins, but also in massive multisite implementations where timing calibration needs to be done for hundreds of device pins.

Another requirement for the ATE is set by the bidirectional implementation of the differential DRSL signals. The inactive DRSL signals that pull both legs of the differential signal to the termination voltage are somehow untypical for conventional differential interfaces. The differential pin electronics of ATE systems might not support this specific signal condition, especially if true differential drivers are used. Thus, special care has to be taken that the ATE channels used for testing XDR have such a capability for their differential pins.

3.3.7 Test Support

XDR in its basic definition does not contain any dedicated features to support device testing besides the broadcast serial write and serial forced read instructions for the low-speed serial interface that simplify device initialization significantly. Since this basic definition specifies mandatory features only, this, however, does not mean that specific XDR implementations do not have such test support features. In fact, XDR is prepared for additional manufacturer specific features by a mandatory set of configuration registers. The content and functionality controlled by these registers is up to the specific XDR design. An example for the freedom XIO/XDR gives to design engineers with regard to test features can be found in [21]. In the XIO design implementation described there, features like serial loopback paths, internal signal probing, and, of course, scan chains are used.

3.3.8 Test Challenges

Some of the test challenges with XDR were already mentioned in Section 3.3.6. We do not want to describe further the ones that are mainly on the ATE feature side. Here we want to mention two test challenges that are driving the

parametric requirements for the ATE system to their limits. The first of these is the low voltage swing at which the DRSL pins of XDR operate. These swings that can go down to 50 mV are a challenge for drive and receive operations with the more challenging conditions for the ATE receivers that have to be able to reliably detect a difference of 25 mV to a fixed threshold voltage. The fact that XDR uses differential signals certainly helps for functional testing. However, if the differential signal implementation is analyzed on a low level, as in Section 3.3.4.2, it becomes obvious that it still has some relationships to single-ended implementations that might influence parametric measurements.

The other challenge is the accuracy required to do parametric timing measurements. Here especially, the setup/hold times defined for the DQ bus drive the requirements into limits that rarely can be fulfilled by ATE systems. Thus, parametric testing and characterization of XDR often require focused ATE calibration support (see Section 9.3).

3.4 GDDR SDRAM

3.4.1 Application Areas

As the name “graphics double data rate synchronous dynamic random access memory” (GDDR SDRAM) suggests, GDDR memory devices are used as high-speed memory for graphics processing units (GPU) to store texture and frame buffer information. Due to their tight linkage to GPUs, GDDR memory devices in general are used in a well-controlled system environment close to the GPU as, for example, on PC graphics cards or game console motherboards. Usually both of the link partners, GPU as well as GDDR memory devices, are directly soldered on the same piece of PCB avoiding connections that compromise signal integrity. This allows for substantially higher data rates for the GDDR interfaces than for the corresponding double data rate (DDR) main memory generations which are connected via long traces and socket connectors to their memory controllers in the chipset or the CPU.

3.4.2 GDDR Fundamentals

Like the DDR main memory standards, GDDR memory device standards are set by the Joint Electron Device Engineering Council (JEDEC). The most recent generation of the GDDR5 standard that was released by JEDEC in December 2009 is GDDR5 [22]. GDDR memories traditionally are ahead of main memory DDR devices with their I/O data rate, and they also often define feature subsets that are candidates for integration into DDR main memory standards. The interesting point about GDDR is that although the

data rate goes up into the multigigabit data range (e.g., 6 to 7 Gbps for GDDR5), it still uses single-ended semiduplex data transmission. This is a fundamental difference to the interfaces discussed before which all use differential signaling and predominantly dual-simplex data transmission. In Table 3.22 a short comparison of the key characteristics between the GDDR generations since GDDR3 is shown. From this table it becomes obvious that keeping the single-ended semiduplex data transmission while increasing the data rate within the GDDR family is achieved by introducing device training steps and modifications of the device timing architecture to increase tolerance for timing parameter mismatches and their variations [23]. Data rate limitations due to the increased noise level caused by the relatively large signal swings required for single-ended signals are addressed by innovative features like data bus inversion (DBI) or address bus inversion (ABI). For the highest speed generation of GDDR, a cyclic redundancy check (CRC) based error code detection was introduced to address the increased likelihood for errors.

Table 3.22
GDDR Comparison

	GDDR3	GDDR4	GDDR5
Release year	2003	2006	2009
Data rate	1 .. 2.6 Gbps	2.2 .. 2.8 Gbps	3.2 .. 7.0 Gbps
Timing architecture	At-cycle source sync.	At-cycle source sync.	Forwarded clock
Device training	No	Data	Address, clock, and data
Bus inversion	No	Data	Data and address
Error detection	No	No	Yes

In the following sections we will focus on GDDR5 as the latest representative of the GDDR family that runs at the highest data rate and also poses new test requirements on ATE systems due to new functionality and clocking architecture changes that are unique for memory devices.

3.4.3 GDDR5 Details

GDDR5 Timing Architecture

The interface timing architecture of GDDR5 consists of three initially independent clock domains. The clock sources for all of these clock domains are not directly provided by a central system clock but are located in the memory controller. Thus, GDDR5 like the previous GDDR generations makes use of source-synchronous interfacing. However, in contrast to earlier GDDR generations, the standard GDDR5 operation mode no longer uses an at-cycle

source-synchronous timing architecture for high-speed data transmission. Instead, a forwarded clock architecture is used. For lower speed operation and better compatibility to earlier GDDR generations, GDDR5 offers the option to operate the devices in a classical at-cycle source-synchronous mode which will not be discussed in this section.

GDDR5 devices follow the traditional DRAM signal grouping into a command/control pin group, an address pin group, and a data pin group. An overview of the GDDR5 signal pins, their type, the clock domain they belong to, and the groups they are associated with is shown in Table 3.23.

Table 3.23
GDDR5 Pin List

Signal Pins	Type	Domain	Group
CK, CK#, WCK01, WCK01#, WCK23, WCK23#	I	-	Clock
RESET#, MF, SEN	I	-	Static
CKE#, CS#, RAS#, CAS#, WE#	I	CK	Command
A8/A7, A9/A1, A10/A0, A11/A6, A12/RFU, BA0/A2			
BA1/A5, BA2/A4, BA3/A3, ABI#	I	CK	Address
DQ0..DQ7, DBI#0, EDC0	IO	WCK01	Data
DQ8..DQ15, DBI#1, EDC1	IO	WCK01	Data
DQ16..DQ23, DBI#2, EDC2	IO	WCK23	Data
DQ24..DQ31, DBI#3, EDC3	IO	WCK23	Data

Clock Pin Group In addition to the differential reference clock signal CK that was used in previous GDDR generations, GDDR5 introduces two new differential clocks: WCK01 and WCK23. These additional clock signals serve as references for the data pin group during write and read operations. The WCK signals run at a frequency that matches the specified device data rate in double data rate mode. The CK clock runs at half of this frequency. For the WCK signals, the JEDEC standard foresees an optional PLL in GDDR5 memory devices. This PLL predominantly will be required for the higher speed grades that are supported by GDDR5.

Command Pin Group For GDDR5, the data rate on the command pins is one quarter of the specified device data rate. This means that the command pin group represents the slowest group of pins of the device (besides the DC pin group that contains power supply and reference voltage pins). The command pins are the only group of pins for which no training step is specified.

Address Pin Group The GDDR5 address bus follows a time multiplexed data transmission concept. Over the nine address signals, 13 address bits and 4 bank bits are transmitted at half the specified device data rate.

Data Pin Group The GDDR5 data pin group that runs at the specified device data rate is divided into four DQ-octets of eight bidirectional signal pins plus the two sideband signal pins DBI and EDC per DQ-octet. While the DBI signals per DQ-octet were already used for previous GDDR generations, the new EDC signals per DQ-octet were introduced with GDDR5. These EDC pins serve as signals which transmit a CRC checksum per DQ-burst from the memory device to the memory controller for read and write operations during normal operation.

The GDDR5 pin groups are organized along the three clock domains that are defined by the CK, WCK01, and WCK23 differential clock signals. The CK signal represents the main reference clock and defines the domain command and address pins are referenced to. While the command data transmission follows a single data rate scheme with command bits sampled by the rising CK edges, addresses are transmitted in double data rate manner with their bits sampled at rising and falling CK clock edges. The WCK01 and WCK23 signals serve as clock references for the DQ data pins, the DBI pins and the EDC pins. Each of the two WCK signals is the reference for two of the DQ-octets and its associated DBI and EDC pins. The three clock domains of a GDDR5 device are synchronized by a dedicated training step during device training that will be described later. In order to facilitate this synchronization, the clocking architecture of GDDR5 allots for a circuitry that provides a feedback loop for CK to WCK phase comparison. A schematic block diagram of the GDDR5 timing architecture is shown in Figure 3.14. The figure does not represent an implementation example but a schematic view of signal-to-clock dependencies only.

GDDR5 Specific Functionality

JEDEC specifies 512 Mb, 1 Gb, and 2 Gb densities for GDDR5 devices. All devices are configurable during power-up to a x16 organization activating 16 of the 32 DQ signals or to a x32 organization that activates all 32 DQ signals. The activation of the x16 organization obviously implies an address space doubling compared to the x32 organization in order to be able to address all cells of the memory core. The 512-Mb devices support eight banks while 1- and 2-Gb devices support 16 banks. Like the previous GDDR generations and DDR3, GDDR5 supports a mirror functionality which is controlled by

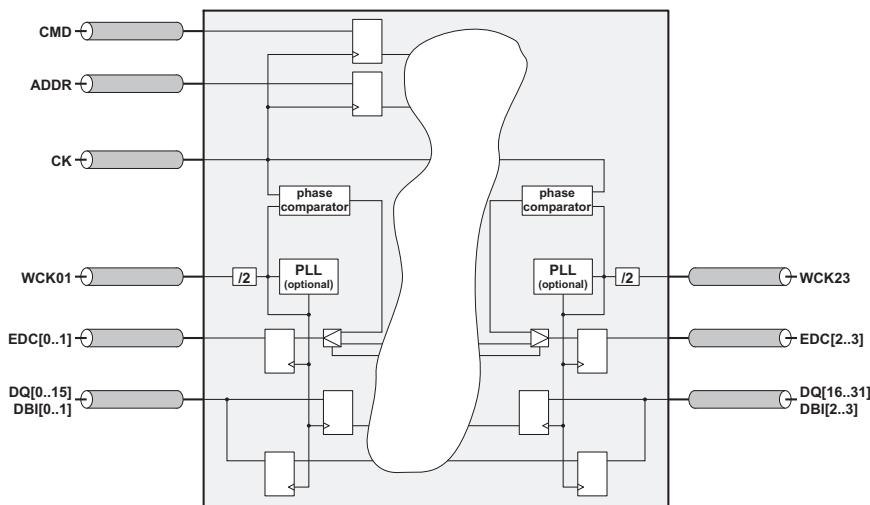


Figure 3.14 GDDR5 timing architecture. (*From:* [22]. ©2009 JEDEC. Reproduced with permission.)

a dedicated device pin. If the mirror function is enabled, the signal-to-ball assignment of the device is reshuffled in a way that the ball-out is mirrored compared to a disabled mirror function. The benefit of the mirror function is that GDDR5 memory devices can be mounted on the frontside and the backside of a printed circuit board in a way that the footprints for devices match not only geometrically but also from a signal point of view on both sides of the PCB. Thus, certain connections between the GPU and both memory devices like the ones for the command and address pins can be shared without generating additional routing overhead to achieve trace matching of the connections to the two different devices as shown in Figure 3.15.

Due to the congruent mounting of the devices to the PCB from both board sides, the mirror function is called the clamshell mode. From a GPU point of view, such a configuration of two memory devices with shared command and address pins looks the same as one single device with doubled capacity. This doubled capacity is represented either by a doubled address space if the devices are configured in x16 mode or by a doubled data word width if the devices are configured in x32 mode. In order to allow the command/address sharing usually used in clamshell mode, GDDR5 offers a dynamic adaption of the on-die-termination (ODT) values by programming the appropriate mode register (MR) bits of the device. By doubling the ODT values of the memory device for shared pins compared to nonshared operation, the GPU pins will see the same termination impedance for a dual-device shared configuration as for a single-device nonshared configuration.

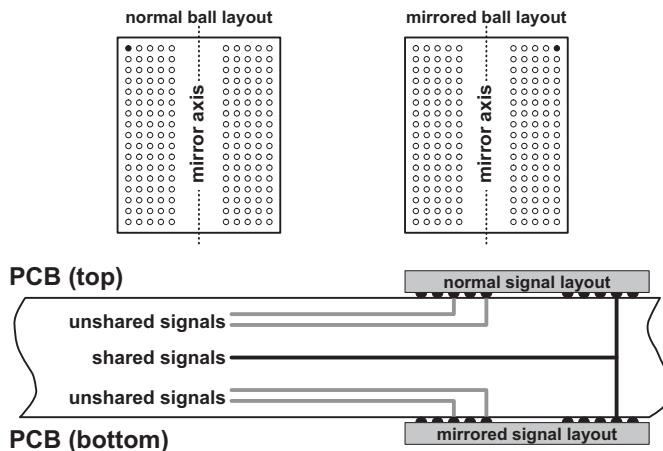


Figure 3.15 Mirrored device mounting to PCB. (From: [22]. ©2009 JEDEC. Reproduced with permission.)

Separate MR impedance control for the WCK clock pins, the data pins, and the command/address pin group allows for the implementation of various sharing topologies.

As will be described in Section 3.4.4.2, GDDR5 uses its supply voltage VDDQ as pin termination voltage if ODT is turned on. As a consequence, each time a low level is applied to a terminated GDDR5 device pin, a current is flowing from the VDDQ supply via the termination impedance into the connecting trace. Thus, the amount of pins driven by a low level can contribute significantly to the dynamic power consumption and supply noise of a device. This especially is true if large signal swings are used as in the case of single-ended interfaces.

With GDDR3 and GDDR4, the DBI technique was applied which aimed to minimize the power consumption and supply noise caused by this effect on the data bus. DBI for GDDR3 and GDDR4 supported two implementation flavors, DBIdc and DBIac. Since DBIac was dropped for GDDR5, we will restrict the discussion to DBIdc and automatically refer to DBIdc if the term DBI is used.

With GDDR5, the DBI (DBIdc only) concept was extended to the address bus as address bus inversion (ABI). If bus inversion is enabled, the amount of address or data bits in low state on the respective bus is analyzed. This analysis happens over the complete bus width for the addresses and over single byte sections for the data bus. If there are more signals in low state than in high state within an analyzed bus section, the logical levels of the pins in that section are inverted. This inversion is signalized on the DBI pin associated

to a data byte for that byte or the ABI pin for the address bus.

Such a bus inversion implementation ensures that worst case maximum half of the bits of a bus are drawing current via the termination impedances due to the applied low level while the other half is drawing no or only a minimal amount of current. The bus inversion for driving and receiving operations of course needs to work slightly differently. For receiving operations, depending on the status of the DBI/ABI pins, the received data needs to be inverted. For drive operations, the ABI/DBI signals are set according to the data/address bit analysis. Thus, both partners involved in a GDDR5 communication link, the memory device as well as the memory controller, have to support bus inversion if this feature is going to be used in the communication.

Another functional feature of GDDR5 to be mentioned here is the use of CRC codes (see Appendix D) for error detection during data transmission. The CRC polynomial used by GDDR5 is the CRC-8-ATM polynomial $x^8 + x^2 + x + 1$. The CRC code is calculated based on the 64 bits of one 8-bit burst per DQ byte plus the 8 burst bits of the associated DBI data. Thus, 72 bits are condensed via the CRC polynomial into one 8-bit CRC code as shown in Figure 3.16.

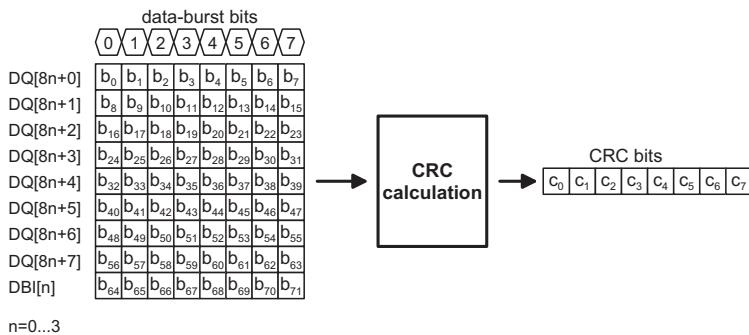


Figure 3.16 GDDR5 CRC code generation. (From: [22]. ©2009 JEDEC. Reproduced with permission.)

The CRC code data flow by GDDR5 is always from the memory device via the EDC pins to the memory controller. As long as no CRC codes are transmitted on the EDC pins, a so-called EDC Hold Pattern consisting of 4 bits is transmitted repeatedly. This pattern is defined by the user with an appropriate mode-register setting. Thus, for normal memory operation, the EDC pins are unidirectional memory output pins.

For write operations, the memory device calculates the CRC code according to the data received on the DQ and DBI pins and returns this calculated CRC code to the memory controller. The memory controller

compares the CRC code it receives after the write operation to the CRC code it has calculated on the write data it sent to the memory device earlier and thus can identify data transmission errors if both CRC codes do not match. For read operations, the memory device sends the locally calculated CRC data with some latency to the read data and the memory controller compares the received CRC code to the CRC code it calculated based on the received data. Figure 3.17 shows a schematic overview of the data flow for read and write operations with enabled error detection.

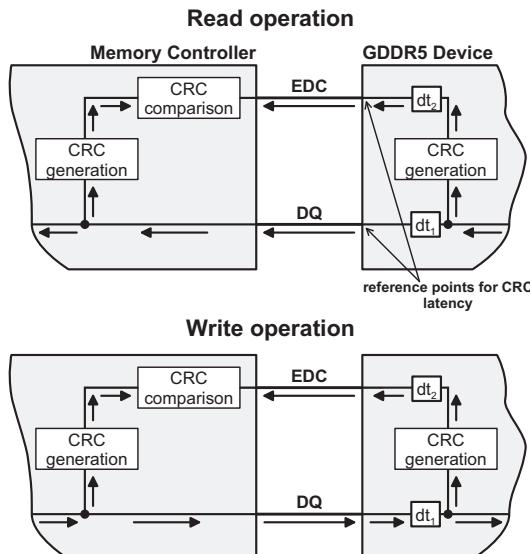


Figure 3.17 CRC data flow for read and write operations.

In order to support the training of the DQ, EDC, and DBI pins that is described later in more detail, GDDR5 devices contain a FIFO that is written either via the address pins or via the DQ pins depending on the command used to send data to the FIFO. The data stored in the FIFO can be transmitted via the DQ, EDC, and DBI pins. With the ability to fill the FIFO via the address pins that run at only half of the DQ data rate, it is possible to provide known data on the pins that run at the highest data rate to adjust to their timing.

3.4.4 GDDR5 Protocol

Since the basic data access command sequences of GDDR5 devices are the same as for other synchronous DRAM components, we will not cover these in detail in this book. If the reader is interested in more detail of how DRAM devices are accessed and what boundary conditions need to be taken into

account for these accesses, this can be found in [24]. In the following we will cover the GDDR5 specifics on the data link layer and the physical layer which are the protocol layers test engineers primarily have to deal with.

3.4.4.1 Data Link Layer

A typical GDDR5 access with address and data transfer is shown in Figure 3.18. In this figure one can see how a data transfer is initiated by an appropriate command that is applied on the command pins. If an address or any other data that needs to be supplied via the address pins is required with the command, the first slice of time multiplexed address data is supplied and latched into the device together with the command on the rising edge of the CK clock signal. The second slice of the time multiplexed address data is latched to the device with the following falling edge of the CK clock. Potential data is supplied or received with certain latency to the rising CK edge that latched the command. This latency is set by a user selected mode register (MR) setting.

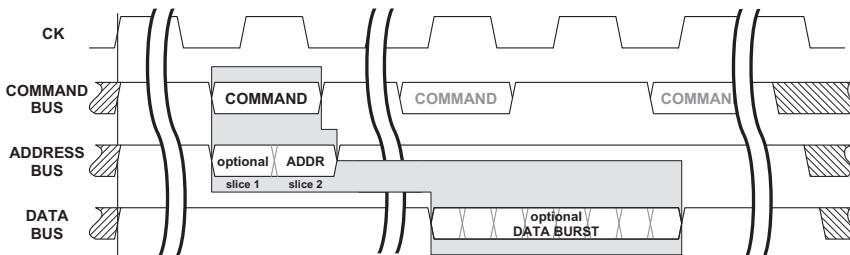


Figure 3.18 Basic GDDR5 access.

Another important detail to mention here is that for accesses which require DQ data in write direction, the DQ bits are center aligned to the associated WCK clock signal. For accesses that expect DQ data in read direction, the timing relation between DQ bits and the appropriate WCK clock signal is edge aligned. The same is true for the bits transferred on the DBI pins which are transmitted together with the associated data on the DQ pins. Potential CRC data is transmitted with a user selected latency to the start of a data burst that is set in the mode register.

As Figure 3.18 shows, GDDR5 operates with 8-bit bursts on its DQ signals. In contrast to previous GDDR generations, GDDR5 does not foresee an optional subburst addressing anymore. Thus, addresses that are supplied to GDDR5 devices address the memory cells on an 8-bit address grid with the LSB of the external address data supplied being the LSB+3 of the physical address space within the device.

GDDR5 CRC Operation

As already described in Section 3.4.3, the CRC code data flow for GDDR5 is always directed from the memory device to the memory controller. From the description above, it is obvious that the base data required to calculate the CRC code in the memory device is available with different latencies relative to the start of the operation for read and write operations. This latency difference is propagated to the EDC pins of the device and thus also is visible on the EDC pins for read and write operations as shown in Figure 3.19. The CRC latencies on the device interface are measured relative to the CK edge that aligns to the first DQ bit of a data burst.

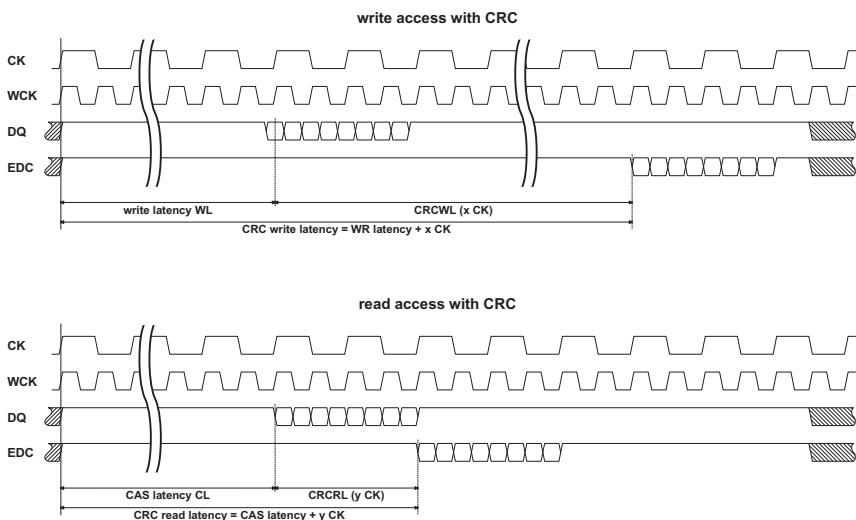


Figure 3.19 GDDR5 CRC timing for read and write accesses. (*From: [22]. ©2009 JEDEC. Reproduced with permission.*)

3.4.4.2 Physical Layer

On the physical layer, GDDR5 uses electrical signaling according to the JEDEC 1.5V Pseudo Open Drain I/O (POD15) standard [25]. All device signals except the differential clocks CK, WCK01, and WCK23 are single-ended. The POD15 standard defines a flexible termination scheme that is able to maintain an equivalent $60\ \Omega$ termination load to the VDDQ supply on shared input pins with one, two, or four inputs operated in parallel as shown in Figure 3.20. The different receiver impedances required to achieve the effective $60\ \Omega$ impedance for the various configurations are programmed by the device according to user controllable register settings. GDDR5 allows

separate impedance control for the Address/Command pin group, the Data pin group, and the WCK pins. Drive pins are defined with a $40\ \Omega$ impedance in the POD15 standard.

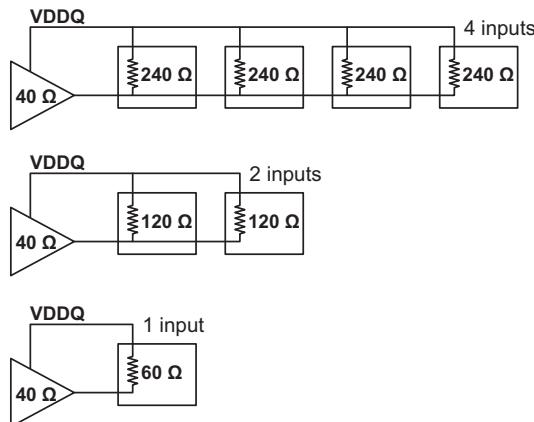


Figure 3.20 POD15 flexible termination scheme. (From: [25]. ©2006 JEDEC. Reproduced with permission.)

In order to allow the device to calibrate its on-die termination (ODT) impedance, the POD15 standard requires a device external reference resistor of $240\ \Omega$ to GND that is connected to the ZQ pin. GDDR5 deviates from the POD15 standard regarding this detail and requires a $120\ \Omega$ resistor to be connected between ZQ and GND. The threshold voltages used by POD15 I/O pins are derived from a reference voltage VREF that usually is provided from an external voltage source. With VREFD and VREFC, GDDR5 requires two of these external reference voltages. The threshold voltage for the data pin groups (DQ, DBI, and EDC pins) is derived from VREFD and the threshold voltage for the Address/Command pin group is derived from VREFC. GDDR5 also offers the option to provide a device internally generated VREFD voltage.

As mentioned already above, the differential WCK pins can be terminated using the on-die termination of the device. For the differential CK clock, this option is not available. Thus, external termination resistors with a value of $60\ \Omega$ each are required between the two CK signal legs and VDDQ. If the ODT option for the WCK pins is not used, also an external cross termination is required for the differential WCK pairs.

Although the GDDR5 training does not necessarily belong to the physical layer from a configuration and execution point of view, we cover it in this section because it has significant impact on the signal timing of the

physical interface and requires timing control in the memory controller and the ATE on the lowest level that is accessible.

GDDR5 Training

In order to align the timing of the memory controller to the timing conditions a GDDR5 device expects, a training sequence as shown in Figure 3.21 is executed as a first step when powering up a GDDR5 device.

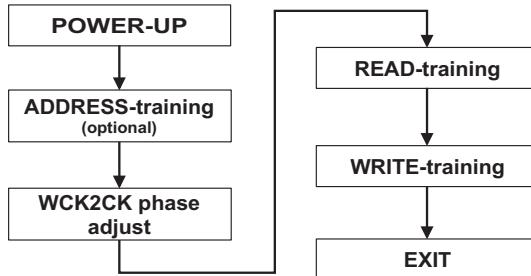


Figure 3.21 GDDR5 training sequence.

The only high-speed signal for which the timing is kept constant during all of the training steps is the differential reference clock CK since this clock serves as a timing reference for all the other signals in a GDDR5 device. After completion of all training steps, the timing relationship between the signals of the GDDR5 device is shown in Figure 3.22. It should be noted that this timing diagram shows the device internal timing relationship at the relevant sampling points inside the device.

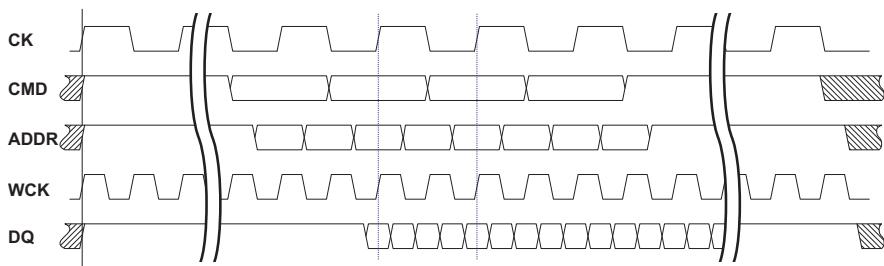


Figure 3.22 GDDR5 timing after training. (From: [22]. ©2009 JEDEC. Reproduced with permission.)

Address Training Address training is an optional step during GDDR5 training. This training step aligns the center of the address pin data eyes, which

are generated by the memory controller/ATE, to the fixed CK dependent latch timing that is valid within the GDDR5 device. For the address training, a feedback loop indicating the status of the training to the memory controller/ATE is created through selected DQ pins which transmit the data sampled on the address pins of the GDDR5 device back to the memory controller/ATE. The data rate on the DQ pins during the address training can be controlled by the memory controller/ATE through the rate at which new addresses are applied. This means the data transmitted by the DQ pins can be sampled safely by the memory controller/ATE even though no read training has taken place when the address training occurs.

Besides the training of the address pins, address training also contains a substep which adjusts the ABI timing to the timing location as required by the memory device. ABI training is performed after the training on the other address pins is completed. Instead of directly comparing to the transmitted address bits on the DQ pins, the memory controller/ATE tests for the inverted address data when performing ABI training.

Clock Training During WCK2CK phase adjustment, the rising edges of the WCK signals are aligned to the rising edges of the CK signal. GDDR5 memory devices contain phase detection circuitry with the functionality represented by the circuitry shown in Figure 3.23. The frequency for each of the two WCK clocks is divided by two in this circuitry and then sampled with the rising edges of the CK clock. The logical result of this sampling operation is transferred via the EDC pins to the memory controller/ATE.

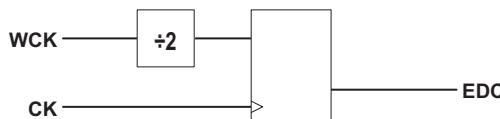


Figure 3.23 GDDR5 clock training phase comparator functionality. (From. [22]. ©2009 JEDEC. Reproduced with permission.)

Since the nominal frequency of the WCK clocks is twice the CK frequency, the divided WCK clocks and the CK clock deliver signals of the same frequency to both inputs of the sampling flip-flop. This means that the logical level captured by the flip-flop can be used as an indicator for the timing relationship between the rising edges of the respective WCK clock and the CK clock. As long as the output of the phase measurement flip-flop is a stable high or low, the rising edges of WCK and CK are far enough apart from each other to not cause setup/hold time violations of the sampling flip-flop. In the case where the WCK and CK edges are aligned to each other, the setup/hold time

violations on the inputs of the flip-flop will cause instable data on its output, and thus on the associated EDC pin of the device.

The two possible states on the EDC pins are referred to as “early” or “late” states as shown in Figure 3.24. An early state is given if the divided WCK is already in the high state when the CK rising edge occurs. The device will indicate such an early state with a high level on the respective EDC output. The late state, on the other hand, is characterized by the rising edge of the divided WCK clock being positioned after the rising edge of the CK clock. Thus, a low state is captured into the phase identification flip-flop and transmitted via the respective EDC pin.

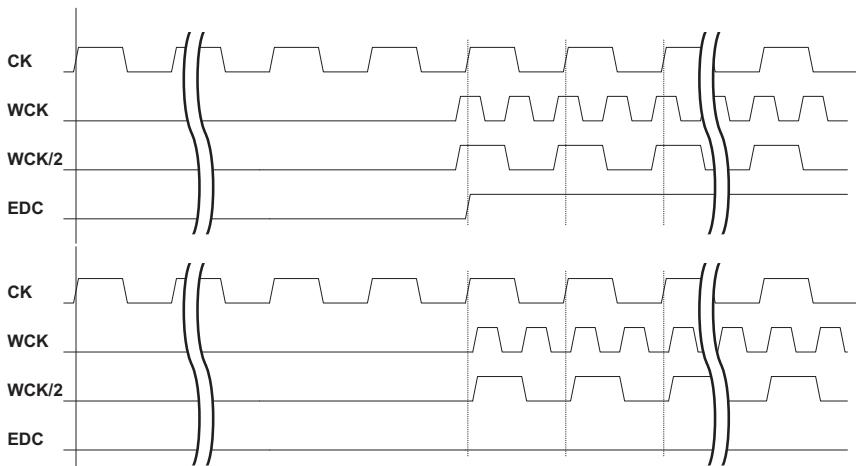


Figure 3.24 Early (top waveform) and late (bottom waveform) GDDR5 WCK to CK alignment states before clock training (reprinted with permission from [26]).

This behavior means the alignment point of the rising WCK and CK edges can be identified by simply moving the WCK edges versus the CK edges and scanning for a state change on the EDC pin(s).

Read Training During read training, the timing of the DQ sampling circuit in the memory controller is adjusted to match the center of the data eyes received from the memory device as shown in Figure 3.25. In order to perform the read training for GDDR5, the data used for the training is written to FIFOs in the device via the already trained address pins. A special load FIFO (LDFF) command is used to load this training data into the FIFOs. A read training command (RDTR) is then executed and one data burst is sent from the FIFOs to the DQ pins of the memory device. Since the memory controller knows

which data to expect, it can align its sampling point per DQ pin by repeatedly executing read training commands and analyzing the data burst responses from the memory device.

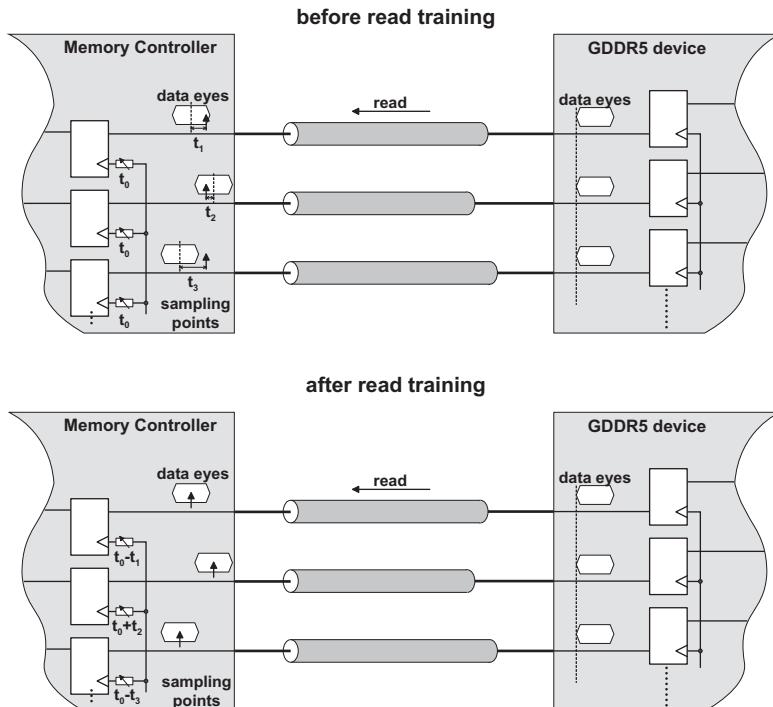


Figure 3.25 GDDR5 read training functionality (reprinted with permission from [26]).

Write Training In theory, write training could be done similarly to the read training by adjusting the DQ sampling point timing inside the memory device instead of the memory controller. However, in order to keep the logical complexity inside the memory device to a minimum, no timing adjustments are done by the GDDR5 device. This means the DQ transmitter timing in the memory controller must be adjusted in such a way that the memory device will receive the DQ data eyes at their optimum fixed sampling points per DQ, as shown in Figure 3.26.

As with the read training, no data transfers to and from the memory core take place during the write training. Special write training commands (WRTR) direct the memory to write the data it receives on its DQ inputs into the FIFOs that are also used for the read training. From there, the memory controller reads back the data the memory device sampled using the read

training command. With the knowledge of the data that was sent to the memory device, and the results that the read training command delivers, the memory controller can adjust its transmitter timing for the single DQ pins until it reads back exactly the data it originally sent into the FIFOs of the memory device before.

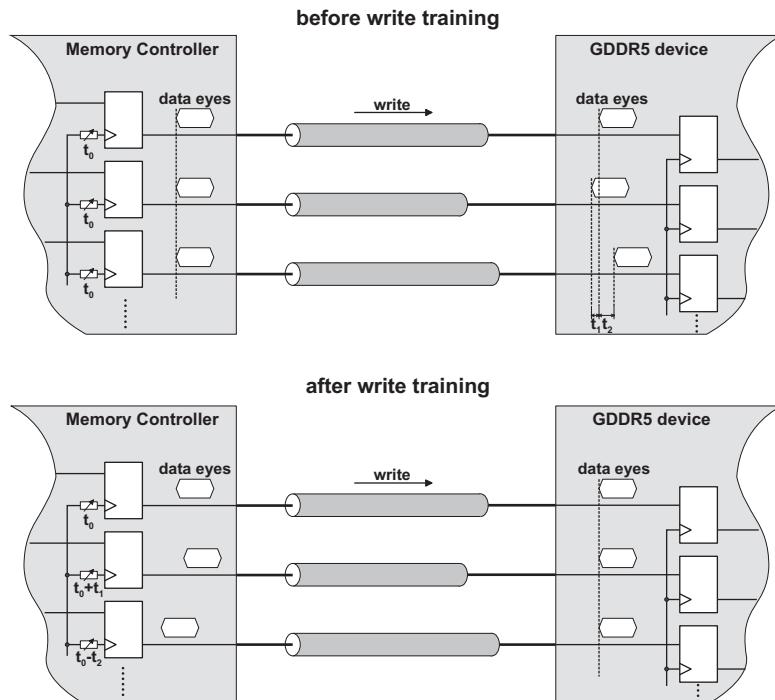


Figure 3.26 GDDR5 write training functionality.

3.4.5 Electrical Specifications

Level Specifications

As already described, the level specifications for GDDR5 follow the POD15 standard. Since POD15 defines the electrical specifications for single-ended signals, the level swings for GDDR5 are relatively large compared to differential interfaces that run at similar data rates. The larger swings are required to provide enough noise margin around the threshold voltages used in the receivers of a GDDR5 interface. Due to the fact that POD15 mimics an open drain electrical interface, three drive voltage levels are required to operate a POD15 interface correctly. Besides the low and high input voltages

that are usually set symmetrically around the respective reference voltage (VREFC or VREFD), a termination voltage with the value of VDDQ is required that needs to be activated when the respective I/O pin is operated in receive mode.

Timing Specifications

As for other high-speed digital interfaces, the eye opening of the data signals is one of the most important timing related parameters that needs to be guaranteed for GDDR5 devices. Compared to previous GDDR generations and DDR memories up to DDR3, setup and hold times between the data signals (DQ) and their timing reference (WCK) have lower priority because the read and write training adjusts the timing in a way that optimum setup and hold times are achieved. Although GDDR5 can contain a PLL for its WCK input, no dedicated jitter specifications are available for GDDR5 as for the other standards discussed so far that contain PLLs as key building blocks. Since the application range and variety of components that communicate with GDDR5 devices is limited and a qualification of certain GPU/GDDR5 combinations can be done on a case-by-case basis, this is a feasible approach to ensure interoperability between components without a formal specification.

3.4.6 ATE Test Requirements

Besides the support for the high-speed device data rate, the primary requirement for testing GDDR5 is that the ATE is able to perform the device training steps because a successful training is a prerequisite for any other test. From a functional point of view, the ATE needs to support the generation of the CRC code patterns that are based on the DQ data used for the test. In order to be able to test the ABI and DBI functionality, the ATE also needs to support the generation of APG-based test patterns that stimulate and compare toggling ABI and DBI pin values.

For parametric measurements the timing accuracy of the ATE system is an important factor. In contrast to other memory interfaces the relative timing between ATE pins is of less importance because setup/hold times are only specified for the lower speed command and address signals. Also the allowed skew for the DQ signals within the two double bytes does not require the ultimate skew accuracy of a few picoseconds between ATE pins which for example is the case for accurate setup and hold time measurements between multigigabit signals.

The timing accuracy within a single pin like the timing delay line linearity is of much higher importance in order to do accurate data eye measurements. As mentioned above, although the GDDR5 specification does

not contain any parametric jitter specification, it seems to be important to have the capability to do jitter measurements on the data interface of GDDR5 because jitter on the DQ and WCK pins will result in a reduction of the setup/hold time margins of the device internal data sampling flip-flop. In order to perform the required jitter tolerance and jitter transfer tests, jitter injection capabilities are required for the ATE channels serving the data pin group and the WCK pins and jitter measurement capabilities are required for the channels serving the data pin group.

3.4.7 Test Support

GDDR5 does not provide any dedicated features that are specified and implemented predominantly to support the component test or characterization besides a non-IEEE 1149.1 compatible boundary scan implementation which is mainly targeted to support interconnection tests for mounted GDDR5 components. GDDR5 devices, however, can be configured into a test mode via a mode-register setting. The device functionality that is enabled in this test mode is not further specified and thus is vendor specific and varies from manufacturer to manufacturer.

Besides this test mode, some of the functionality required by the standard can be reused for test and characterization purposes. One component of the memory that is a perfect fit for such a reuse is the FIFO that is used for read and write training. The availability of this FIFO provides an easy way to access the memory with a user defined pattern that is suited to do high-speed I/O characterization without the need to take care of memory refresh cycles during, for example, time consuming bit error rate measurements on the I/O cells for characterization purposes.

3.4.8 Test Challenges

The test challenges for GDDR5 cannot be simplified to just consider the high data rate although it is an important ATE hardware related factor. This is especially true if it comes to cost efficient production solutions with maximized site-count. However, application relevant challenges like the complex GDDR training sequence require appropriate tooling and ATE capabilities to adapt the timing of the test system to the device on a per-pin basis [26, 27].

Another application related challenge is the variable read and write latency for the GDDR5 CRC generation. This variable latency might not be a significant issue if test patterns are derived from simulation. However, for patterns that are generated algorithmically, as is the case for memory patterns, this can be a challenge if the pattern generation hardware or the pattern

compiler does not support the specific GDDR5 case. The last challenge to be discussed here is the impedance mismatch between the ATE pin electronics that usually are designed with a $50\ \Omega$ impedance and the GDDR5 device impedance which is $60\ \Omega$ for device inputs and $40\ \Omega$ for device outputs.

3.4.8.1 High-Speed Data Rate

Most of the test challenges presented by the high data rates that GDDR5 deploys do not differ from the other high-speed applications discussed so far. They are mainly created by bandwidth limitations of the signal paths between the ATE pin electronics and the DUT and the effects of these limitations on signal integrity. The fact that GDDR5—as XDR—transfers data in a semiduplex manner, of course, raises the same challenges that already were discussed in Section 3.3.8.

The difference with XDR, however, is that GDDR5 uses single-ended signaling. The large swings of this single-ended interface can have a significant impact on the requirements for the complete signal path that consists of the test fixture, the ATE internal signal routing, and the pin electronic front end. In particular, critical items in this regard are, for example, crosstalk, power distribution, drive and threshold voltage offset accuracies, and the bandwidth of all signal path components.

3.4.8.2 Training Test Challenge

If the different training steps are analyzed with respect to their general requirements for an ATE based test implementation, it becomes obvious very quickly that two timing alignment methodologies are required to address all GDDR5 training steps. These methodologies are eye center alignment and transition alignment, which are described in more detail in the following. The challenge with both of these training steps is that the ATE needs to provide an efficient way to adapt its timing individually per pin to the requirements of the device. The CDR functionality many ATE systems provide with their high-speed pin electronics only solves the challenge raised by the read training in this regard. The write training is not covered by this functionality because a CDR only works for device output pins whereas the write training needs to adjust the timing on device input pins.

Of course, a potential CDR also is not usable for the WCK2CK phase adjust training. Here the rising edges of two signals need to be aligned relative to each other based on the feedback of the phase detection circuitry in the GDDR5 memory device. In order to achieve an accurate transition alignment between the WCK and CK signals, simply scanning for a transition of the state of the EDC feedback pin while moving the WCK timing might be sufficient

for production oriented functional testing, but not for device characterization. The reason for this is that in a real circuit, there is not one distinct time at which the early/late state transition on the EDC pin will occur. Instead there is a band or region in which that transition will occur with a certain probability for each distinct timing point within that band as shown in Figure 3.27.

This behavior is caused primarily by the jitter that is present on the CK clock and the WCK clocks. Because the jitter on these signals is typically uncorrelated, the early/late result on the EDC output might vary even if the timing settings for CK and WCK stay constant. The jitter on the clock signals and the relative timing variations between CK and WCK that are caused by the jitter follow statistical processes. The early/late measurement is influenced by this statistical behavior and needs to be analyzed by statistical means to get the most accurate training results.

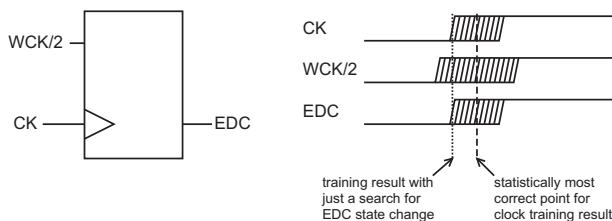


Figure 3.27 GDDR5 transition alignment uncertainty (reprinted with permission from [26]).

Another contributor to the EDC early/late switching uncertainty is the setup and hold time behavior of the sampling flip-flop in the phase measurement circuitry. Since the goal for WCK2CK phase adjustment is to align the transitions of the CK and WCK (and thus the divided WCK) clocks, the transition alignment point represents the worst case setup/hold time situation for the sampling flip-flop. A statistical measurement approach for the EDC early/late indication will also cover the unpredictable behavior of the sampling flip-flop while it is operated in the setup/hold time violation band.

3.4.8.3 GDDR5 CRC Test Challenge

Testing the CRC functionality for GDDR5 represents a challenge because memory test patterns usually are generated algorithmically. While the pure generation of the CRC codes can be achieved through linear feedback shift registers on classical hardware based APGs or XOR operations on compiler based per-pin APGs, the common challenge is the handling of the varying latency between read and write operations.

The CRC compare data needs to be generated in the respective APG at the time the DQ data is valid. However, the generated CRC data has to be compared at a later point in time that corresponds with the different CRC latencies for write and read operations. For hardware based APGs, this means that not only the very specific CRC polynomial used by GDDR5 needs to be implemented in hardware, but also the CRC checksum compare needs to be delayed, for example, by a variable length FIFO in the CRC signal path.

For compiler based per-pin APGs the same requirements apply. The flexible adaptation to the CRC polynomial used by the application usually can be easily done via XOR operations in the pattern description itself. To address the variable CRC delay between CRC data generation and CRC compare, the APG compiler needs to support storage of the CRC data for a user defined delay.

3.4.8.4 Termination Impedance Mismatch

As described in Section 3.4.4.2, POD15 drivers use a $40\ \Omega$ impedance that drives into $60\ \Omega$ terminated POD15 receivers. In contrast to this, ATE pin electronics usually provide a $50\ \Omega$ impedance for both drive and receive functionality.

The problem with this impedance mismatch is twofold. From an AC point of view, the impedance mismatch, of course, represents a discontinuity that causes reflections on the signal path. The mismatch, however, is small enough that reflections do not contain too much power and are absorbed to a large extent by the loss of the signal path. Also, an impedance matching on the test fixture usually would create even bigger AC issues due to mounted components as described in Section 8.8. The second aspect of the impedance mismatch is the DC effect that causes a shift of the device pin voltage levels compared to the ones programmed on the ATE drivers and receivers. In order to drive the desired voltage levels and measure the correct voltages at the device pins, an adaption of the ATE drive and receive levels to the impedance environment is required.

3.5 Other High-Speed Digital Interface Standards

Apart from the high-speed digital standards presented in the previous sections, there are many other standards that are also important. One of these is the serial ATA standard (SATA) [28].

The SATA standard was developed to supersede the ATA and SCSI standards used mainly for connecting storage devices in computer platforms. The current standard (generation 3) runs at 6 Gbps with generation 2 at 3 Gbps

and generation 1 at 1.5 Gbps. Two features of the SATA standard that might present some new challenges are the SATA implementation for out-of-band signaling (OOB) and spread spectrum clocking (SSC) requirements. In SATA, OOB is used for link initialization, recovery from low power states, and for specific actions during test modes.

The signaling used for OOB applies low-speed differential data with gaps where both legs of the differential line will go to the same voltage as shown in Figure 3.28 [28]. In the gaps between data transmission response data from the link partner is expected. With this kind of signaling, a handshake protocol is implemented that allows communication between the link partners even if the link is not fully powered up and initialized. SSC is discussed in Section 5.8.3.

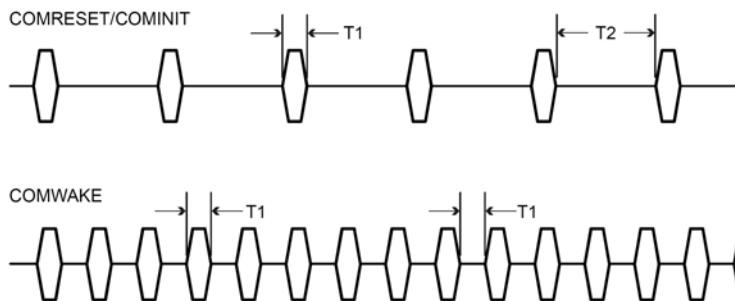


Figure 3.28 Out-of-band signaling in the SATA standard. (From: [28]. ©2002–2009 Serial ATA Organization. Reprinted with permission.)

Another important standard is the Universal Serial Bus (USB) [29]. USB was designed to allow many peripherals to be connected using a single standardized interface socket and to improve the plug-and-play capabilities by allowing devices to be connected and disconnected without rebooting the computer (hot swapping) and other important features. The design of USB is standardized by the USB Implementers Forum (USB-IF), an industry standards body incorporating several companies. The USB standard has several generations with different data rates. Generation 3 supports a maximum data rate of 5.0 Gbps.

The High-Definition Multimedia Interface (HDMI) standard is a compact audio/video connector interface for transmitting uncompressed digital streams. Like most standards it has evolved over time in different generations [30]. HDMI/TMDS interface is a clock forwarding based interface since a

clock needs to be transmitted on a separate link with the data but this clock provides only a reference to the link partner and is not used to directly latch the data at the receiver.

One of the major challenges of the HDMI/TMDS interface is its AV_{cc} voltage level of 3.3 V. This level might pose a challenge for testing HDMI especially for ATE pin electronics that is designed for high-speed digital applications which typically operate at lower voltage levels. While the specified swing value of HDMI should not be an issue for such a high-speed pin electronics, the common mode voltage resulting from the 3.3 V requirement might be beyond the supported level ranges.

In situations where the measurement instrumentation cannot achieve this voltage level, two options are available. One option is to use an extra low-speed digital channel which has a larger voltage range or a power supply to help the high-speed digital channel to achieve the required common mode voltage by using the appropriate connection circuit. Another option is to use a second power supply to create a virtual ground reference on the device and this way shift the required common mode value inside the level range of the ATE pin electronics.

References

- [1] PCI-SIG, *PCI Express External Cabling Specification Revision 1.0*, Jan. 2007.
- [2] PCI-SIG, *PCI Express Base Specification Revision 1.0*, July 2002.
- [3] PCI-SIG, *PCI Express Base Specification Revision 1.0a*, Apr. 2003.
- [4] PCI-SIG, *PCI Express Base Specification Revision 2.0*, Dec. 2006.
- [5] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [6] J. Morriss, “PCI Express 3.0 Electricals,” *PCI-SIG PCIe Technology Seminar*, Feb. 2010.
- [7] PCI-SIG, *PCI Express Jitter and BER Revision 1.0*, Feb. 2005.
- [8] H. Werkmann, “PCI Express: ATE Requirements and DFT Features for Functional Test Optimization,” *IEEE European Test Workshop*, May 2003.
- [9] H. Werkmann, “Enabling the PCI Express Ramp – ATE Based Testing of PCI Express Architecture,” *IEC EuroDesignCon*, Oct. 2004.
- [10] HyperTransport Consortium, HyperTransport Consortium Web site, 2008.
<http://www.hypertransport.org>.
- [11] B. Holden, J. Trodden, and D. Anderson, *HyperTransport 3.1 Interconnect Technology*. Mindshare, 2008.

- [12] HyperTransport Consortium, "HyperTransport Specification 3.10 Revision B," 2009. http://www.hypertransport.org/docs/twgdocs/HTC20051222-0046-0033_changes.pdf.
- [13] E. Fang, G. Asada, R. Kumar, S. Hale, and M. Leary, "A 5.2Gbps HyperTransport™ Integrated AC Coupled Receiver with DFR DC Restore," *IEEE Symposium on VLSI Circuits*, 2007.
- [14] G. Esch and T. Chen, "Loopback Architecture for Wafer-Level At-Speed Testing of Embedded HyperTransport Processor Links," *IEEE Custom Integrated Circuits Conference*, 2009.
- [15] M. W. Riley, J. D. Warnock, and D. F. Wendel, "Cell Broadband Engine Processor: Design and Implementation," *IBM Journal of Research and Development*, vol. 51, no. 5, 2007.
- [16] H. Yasukawa, H. Goto, and K. Ueno, "LSI Architecture of SpursEngine™," *Toshiba Review*, vol. 63, no. 11, 2008.
- [17] R. Perego, "XDR Architecture," *Rambus Advanced Information*, 2005.
- [18] W. Wang and F. A. Ware, "XDR IO Cell," *Rambus Advanced Information*, 2006.
- [19] Qimonda AG, *IDGV1G-05A1F1C-[40X/50X/55X] Internet Data Sheet*, Dec. 2008. <http://www.qimonda.com/graphics-ram/gddr5/index.html>.
- [20] R. Crisp, "Direct Rambus Technology: The New Main Memory Standard," *IEEE Micro*, pp. 18–28, Nov. 1997.
- [21] V. Khawshe, K. Vyas, R. Rangnekar, P. Goyal, V. Krishna, K. Prabhu, P. Kumar Venkatesan, L. Raghavan, R. Palwai, M. Thrivikraman, K. Desai, and A. Abhyankar, "A 2.4Gbps-4.8Gbps XDR-DRAM I/O (XIO) Link," *22nd International Conference on VLSI Design*, pp. 373–378, Jan. 2009.
- [22] JEDEC Solid State Technology Association, *GDDR5 SGRAM*, Dec. 2009. JSED212.
- [23] H. Werkmann and A. Olenyi, "High-Speed Memory Interfaces – Providing Bandwidth for the Future," *Verigy Users Group Conference (VOICE)*, Dec. 2007.
- [24] D. T. Wang, *Modern Dram Memory Systems: Performance Analysis and Scheduling Algorithm*. Ph.D. thesis, University of Maryland at College Park, 2005.
- [25] JEDEC Solid State Technology Association, *POD15 - 1.5V Open Drain I/O*, Dec. 2006. JSED8-20.
- [26] H. Werkmann, D.-M. Kim, and S. Fujita, "GDDR5 Training – Challenges and Solutions for ATE-Based Test," *Asian Test Symposium*, Nov. 2008.
- [27] H. Werkmann, H.-H. Jin, J.-W. Seo, and H.-W. Kang, "GDDR5 Test Challenges and Their Solutions on the Verigy 93000 HSM Series," *Verigy Users Group Conference (VOICE)*, Sept. 2008.

- [28] Serial ATA International Organization, *Serial ATA Revision 3.0*, June 2009. Gold Revision.
- [29] *Universal Serial Bus 3.0 Specification*, Sept. 2008.
- [30] Hitachi Ltd., Matsushita Electric Industrial Co. Ltd., Philips Consumer Electronics International B.V., Silicon Image Inc., Sony Corporation, Thomson Inc., Toshiba Corporation, *High-Definition Multimedia Interface Specification Version 1.3a*, Nov. 2006.

4

ATE Instrumentation for Digital Applications

The objective of this chapter is to provide a basic introduction to automated test equipment (ATE) for high-speed digital applications. For readers who have no ATE experience, [1, 2] can provide a starting point.

Like the semiconductor industry, ATE systems have evolved in the last decades with larger pin counts and higher data rates. Figure 4.1 shows two ATE systems from different manufacturers. From the hardware point of view, state-of-the-art ATE systems are typically composed of two parts: first, a testhead that can be filled by different pin electronics cards (digital, mixed-signal, power supplies, RF instruments) depending on the application one intends to address; second, a support rack that connects to the testhead and contains support infrastructure like workstation communication, power converters, and cooling. The exact split of functionality between the testhead and support rack components strongly depends on the ATE manufacturer and even on the application area an ATE system is targeted for. For example some ATE systems that are targeted for specific applications like memory or mixed-signal do not have the option of testing other applications, even by adding or exchanging the pin electronics cards. Typically the pin electronics cards are provided by the manufacturer of the mainframe although there have been efforts to allow different vendors to share the same testhead [3].

Typically the ATE system is not used on its own, especially in a production environment. It can be connected either to a handler for packaged parts testing or to a wafer prober for testing the parts at a bare die level as shown in Figure 4.2 [4].

Each pin electronics card typically contains several measurement channels. Figure 4.3 shows a few examples of digital ATE pin electronics



Figure 4.1 Example of automated test equipment (ATE) systems from different manufacturers (left: courtesy of Verigy; right: courtesy of Teradyne).



Figure 4.2 Example of an ATE system in a production environment: right: ATE system next to a handler for packaged parts testing (courtesy of Verigy); left: ATE system docked to a prober for wafer testing. (From: [5]. ©2005 Jose Moreira. Reprinted with permission.)

cards. Some pin electronics architectures divide the pin electronics card into channel assemblies containing a subset of the total number of channels. This architectural approach has significant advantages during tester maintenance where only the channel assembly containing the malfunctioning measurement channel needs to be substituted and not the entire pin electronics card. Figure 4.4 shows a picture of a channel assembly containing four single-ended measurement channels that is a part of an ATE digital pin electronics card containing eight channel assemblies or a total of 32 single-ended measurement channels.

Another important hardware component is the connection of the ATE system to the test fixture where the DUT will reside. Figure 4.5 shows one

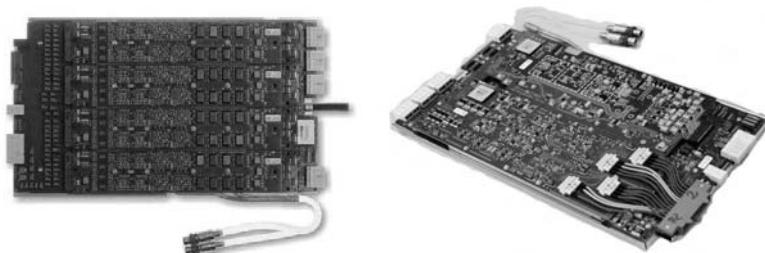


Figure 4.3 Photograph of a medium-speed (533 Mbps) ATE pin electronics card (left) and a high-speed (12.8 Gbps) ATE pin electronics card (right). (Courtesy of Verigy.)

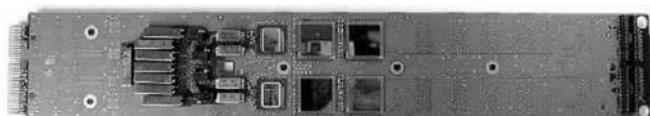


Figure 4.4 Photograph of a pin electronics channel assembly containing four 3.6-Gbps digital pins. Eight of these channel assemblies are then used on an ATE pin electronics card. (Courtesy of Verigy.)

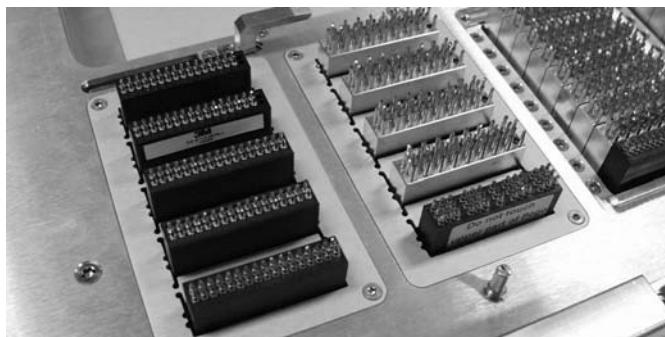


Figure 4.5 ATE to test fixture connection through a pogo pin assembly. (Courtesy of Verigy.)

example of a pogo pin type ATE connection interface. Although the pogo pin approach is prevalent in most ATE applications, other options such as coaxial connectors do exist [5].

There are two common approaches in using ATE channels to measure and test high-speed digital I/O interfaces as shown in Figure 4.6. The first is based on using the driver/receiver in the ATE digital pin electronics card not only for performing functional tests but also for AC parametric measurements (transition time, jitter and so on). The second approach is based on using a sampler ATE card for characterizing the DUT transmitter and relying on the standard digital pin electronics for functional tests and the stimulus signal to the DUT receiver. Note that for this option the performance requirements on an ATE digital pin electronics are more relaxed than those on an ATE digital pin electronics card for complete I/O characterization.

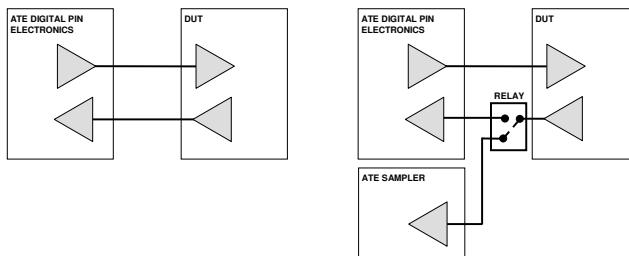


Figure 4.6 The two typical configurations for testing or characterizing a high-speed digital I/O cell with an ATE system. Using a pure digital pin electronics approach (left) and using a sampler/digitizer ATE instrument together with a digital pin electronics card (right).

Also note that it is technically possible to use other types of measurement instrumentation like high performance time interval analyzers or a real-time sampling instrument. However, this type of instrumentation is typically not available as an integrated ATE card but as stand-alone instrumentation. It is feasible to integrate these instruments into an ATE system although it does present some mechanical and signal integrity challenges (see Section 7.9).

It is important to note that specialized instruments, whether they are fully integrated or semi-integrated stand alone, typically do not achieve the same amount of parallelism as is the case for digital pin electronics cards. Thus, usage of specialized instruments is only feasible for devices with only a few high-speed I/O pins without affecting the test time significantly.

The next sections discuss the digital pin electronics ATE card and the sampler ATE card in detail. Digital power supply ATE cards are also briefly discussed.

4.1 Digital Pin Electronics ATE Card

The digital pin electronics ATE card is the main element of an ATE system for digital applications. It provides the stimulus in the form of a digital waveform with programmed levels and timing at a given data rate. It also compares an incoming digital waveform from the DUT to an expected pattern with a programmed voltage threshold and compare strobe timing.

The pattern data for stimulus and compare typically is stored in the vector memory of the ATE system. For certain applications or device building blocks (e.g., memory devices or embedded memory) some ATE systems generate the pattern data on-the-fly using dedicated algorithmic pattern generation (APG) hardware. Algorithmic pattern generation also is used by some digital pin electronics cards that not necessarily target memory test but offer DfT support (e.g., with PRBS data generation and comparison—see also Section 6.3.1.2). Figure 4.7 shows a high-level block diagram of the typical bidirectional driver/receiver pin electronics in an ATE card for high-speed digital applications. As further shown in the figure, digital pin electronics architectures usually include the possibility of performing DC measurements and sometimes even offer additional capabilities like an integrated time interval analyzer (TIA) for the measurement of timing events and jitter.

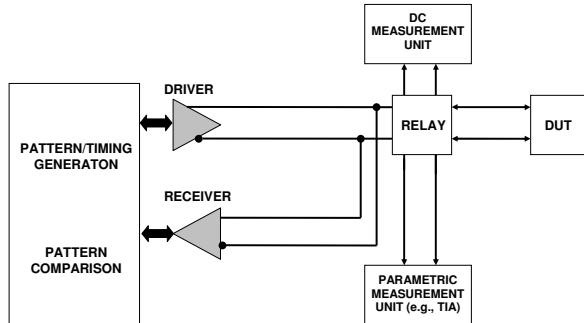


Figure 4.7 High-level diagram of a differential driver/receiver ATE pin electronics for high-speed digital applications.

The pin electronics driver/receiver topology can be divided into different categories as shown in Figure 4.8 with some ATE pin electronics implementations containing several of those topologies to allow the testing of a larger number of applications.

Figure 4.9 shows a block diagram of the three pin electronics topologies for differential high-speed digital applications.

In a unidirectional interface configuration, the driver and receiver are connected to separate pins. This is the typical topology that is required for

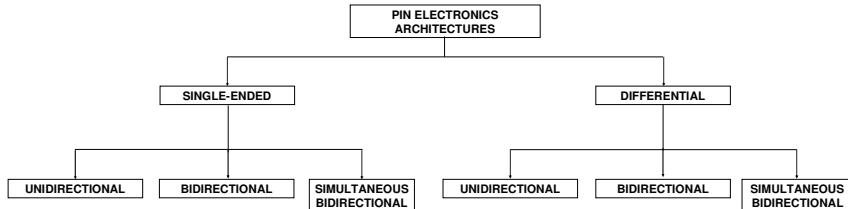


Figure 4.8 High-level categorization of the pin electronics architectures.

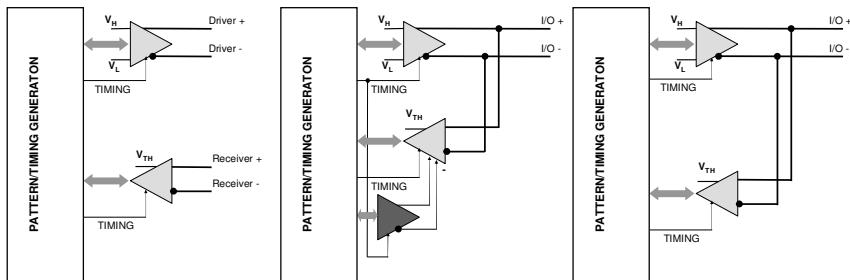


Figure 4.9 Different topologies for an ATE digital pin electronics driver/receiver assembly (left: differential unidirectional; center: differential bidirectional; right: simultaneous bidirectional).

serial interfaces like PCI Express [6]. This kind of topology allows the design of very high-performance pin electronics since one avoids the disadvantages of connecting the driver and receiver together, which are discussed later. The drawback is the lack of flexibility in assigning the ATE pins to the DUT and also the need to use techniques like dual transmission line (also referred to as fly-by) for testing bidirectional interfaces (see Section 9.4).

In a bidirectional configuration both the driver and receiver are connected together. This is the standard topology of most ATE digital pin electronics. This configuration allows any pin to be used as driver or receiver, increasing the flexibility in the ATE pin assignments with the DUT. From a performance point of view, the electrical connection of the driver and receiver does have its design challenges (e.g., the driver capacitance will be present when the receiver is measuring). Although this configuration can handle bidirectional interfaces, the round-trip delay between the pin electronics and the DUT might make bidirectional testing with a single pin impossible, again requiring the use of fly-by techniques.

In a simultaneous bidirectional configuration the challenge of the round-trip delay of a standard bidirectional configuration is solved by subtracting the ATE driver signal from the DUT signal being measured at the ATE receiver.

This topic is further discussed in Section 4.1.6.

Apart from the standard capabilities of a digital pin electronics card and the possible driver/receiver configurations, there are several more advanced components that a digital pin electronics card might contain to provide specialized support for the characterization and testing of high-speed digital applications. Figure 4.10 shows an ATE pin electronics block diagram that includes several of these additional nonstandard components which are described in the next sections.

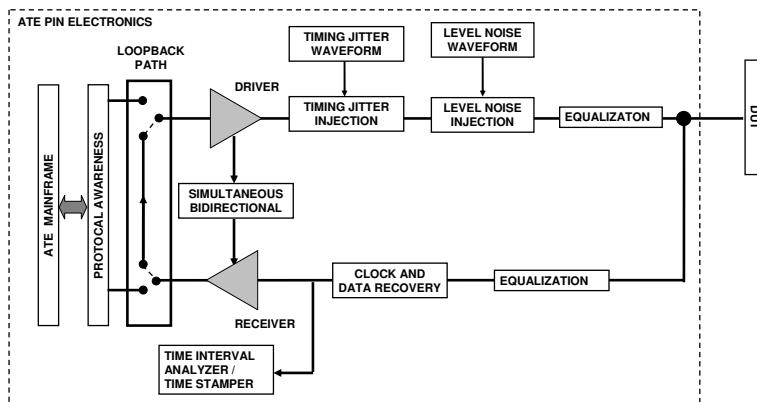


Figure 4.10 Block diagram of an ATE digital pin electronics including several nonstandard capabilities.

4.1.1 CDR and Phase Tracking

Providing clock and data recovery (CDR) or similar capabilities on the ATE pin electronics receiver has some significant advantages for high-speed digital applications. One major advantage is the ability to automatically set the receiver strobing point in the center of the data eye without having to resort to timing searches. Also, several high-speed standards require a CDR with a specific response to be included for measurements like jitter generation (more on this topic in Section 9.5).

However, it is important to note that implementing a CDR with a programmable loop filter is not a trivial task and will increase the pin electronics cost while a CDR with a fixed loop filter might not be appropriate for some high-speed standards depending on the loop filter characteristics. Figure 4.11 shows a block diagram of an ATE pin electronics receiver with an integrated phase detector block. The CDR functionality is achieved by using the phase detector in combination with a digital control loop that is connected to the ATE timing system.

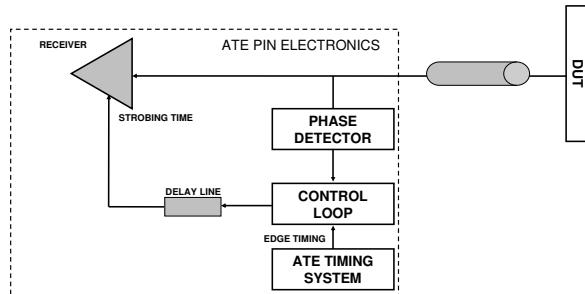


Figure 4.11 Example of a high-level block diagram of an ATE pin electronics receiver with an integrated phase detector.

Note that in the block diagram the control loop not only takes into account the phase information from the phase detector to set the strobing point in the middle of the data eye but also the timing shift of the edge that is required by the ATE timing system for measurements like a data eye diagram (Section 5.4.2).

4.1.2 Equalization

Equalization (also referred to as pre-emphasis/de-emphasis on the driver side) is one key item to address the challenge of waveform degradation due to the test fixture loss for high-speed digital measurements. The challenge is that even when designing the test fixture using the best available materials and design techniques the loss might be too high for the measurement. The reason is that the length of the signal traces on a typical ATE test fixture coupled with the number of pins makes it very difficult to create a solution that provides minimal degradation for a multigigabit digital signal. To address this problem, modern ATE pin electronics include an equalization block that compensates for the test fixture loss both on the drive and receive sides. Section 9.8.3 discusses equalization in more detail.

It is important to note that although equalization will compensate the test fixture loss, this does not mean that proper test fixture design is no longer important. Test fixture design is still critical, as will be discussed in Chapter 8.

4.1.3 Time Interval Analyzer or Time Stamper

Some digital pin electronics implementations include a time interval analyzer or time stamper which records the time when a predetermined event occurs. This event is defined by the signal under test crossing a given level threshold with a given edge polarity (rising, falling, or either). The term “time stamper”

or “time interval analyzer” refers to the same type of measurement approach, but with the term “time interval analyzer” used usually for an external instrument [7]. By integrating this type of instrument on the pin electronics, measurements like the frequency of a clock signal can be performed in a very short time compared with other types of instruments. Section 7.3 discusses the time interval analyzer in the context of bench instrumentation.

4.1.4 Timing Jitter Injection

The ability to support timing jitter injection on the pin electronics driver is critical for high-speed digital testing (see Section 5.7). The objective is to measure the ability of the DUT receiver to tolerate jitter that is present on the incoming data waveform. This requires that the ATE pin electronics driver is able to inject well-defined timing jitter on the stimulus data waveform to the DUT receiver. Several approaches are possible for jitter injection implementation on the ATE pin electronics.

Timing Jitter Injection Through a Voltage Controlled Delay Line

The most straightforward way to inject timing jitter into a waveform is to use a voltage controlled delay line as shown in Figure 4.12. Depending on the voltage level at the control input of the delay line, the time delay through it will vary. Consequently, this will delay the edges of the waveform. The voltage control signal can be generated by a waveform generator and can be completely arbitrary. The advantage of this approach is that it is continuous in time in the sense that the delay line is injecting jitter (delay) continuously on the waveform although for measurement purposes only the timing of the transition edges will be important.

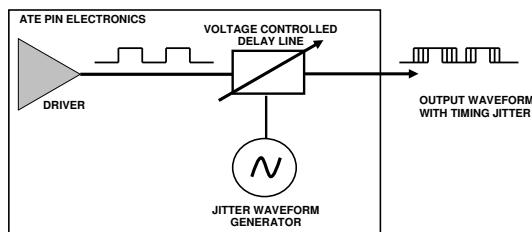


Figure 4.12 Jitter injection through a voltage controlled delay line.

One variation on the approach shown in Figure 4.12 is to inject the jitter into the reference clock of the ATE system which then is used to latch out the data from a FIFO as shown in Figure 4.13.

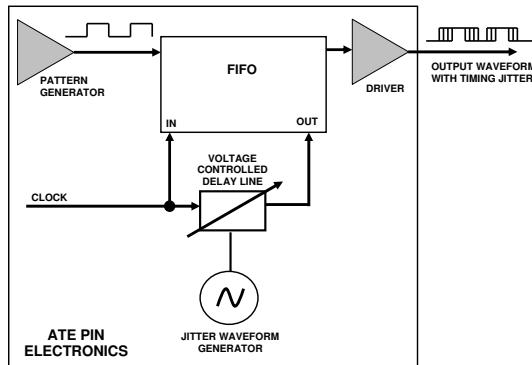


Figure 4.13 Jitter injection through a voltage controlled delay line in the reference clock and a FIFO.

Timing Jitter Injection Through Edge Time Reprogramming

Another possible approach for jitter injection is to use the timing programming capabilities of the driver pin electronics. Instead of correctly programming the timing for the driving edges of the data pattern, the timing is misaligned so that jitter is added to the pattern timing as shown in Figure 4.14 [8]. The disadvantage of this approach is that it depends on the timing programming capabilities of the pin electronics. Some ATE timing systems might be very restricted regarding reprogramming the driving edge timing. This can significantly restrict the ability to inject jitter using this approach. The other disadvantage is that this type of jitter injection is not easy to map to the typical jitter injection requirements from high-speed digital standards.

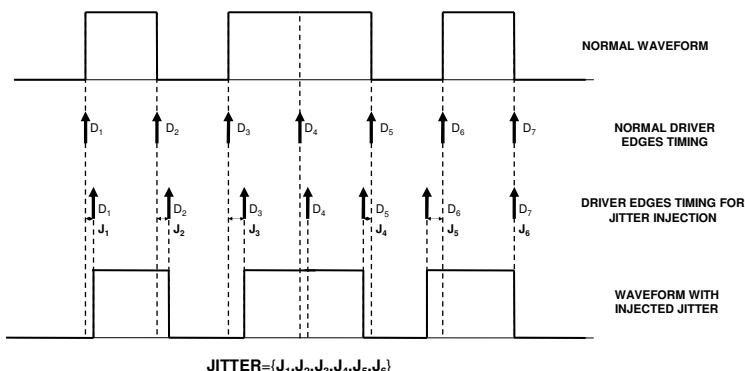


Figure 4.14 Jitter injection through edge timing programming.

Timing Jitter Injection Through Programmable Equalization/Pre-Emphasis

Since some ATE pin electronics cards for high-speed digital applications include programmable equalization (also referred to as pre-emphasis/de-emphasis) on the driver to compensate the test fixture loss, there is the possibility to use this programmable pre-emphasis to inject jitter in the form of data dependent jitter (DDJ) to the DUT. This topic is discussed in more detail in Section 9.8.3.4.

4.1.5 Amplitude Noise and Common Mode Voltage Injection

Another typical requirement on the pin electronics driver circuitry for high-speed digital applications is the ability to inject amplitude noise on the ATE driver signal to test the DUT receiver tolerance for this effect. While amplitude noise on a single-ended signal results in amplitude noise only on the DUT receiver, amplitude noise on the legs forming a differential signal can manifest itself in the form of two effects at the DUT receiver. These are differential amplitude noise and common mode noise.

For differential interfaces typically both of these parameters are verified. If the amplitude noise on both legs of the differential interface has the same phase and magnitude at all instances (e.g., due to crosstalk on the differentially coupled connection), this translates into common mode noise only. If the noise characteristics on the two legs differ, phase differences and magnitude differences translate into differential amplitude noise.

Two possible approaches that are used on pin electronics architectures to stimulate amplitude and/or common mode noise are discussed below.

Continuous Time Voltage Source

The most straightforward way is to modulate the signal from the driver with a voltage waveform (e.g., a sinusoidal wave) as shown in Figure 4.15. If the two legs of a differential signal are modulated by separate modulation sources, nonuniform modulation for the differential legs is possible. This allows dedicated common mode and differential amplitude noise injection. This approach could be implemented using a resistive network for example.

Extra Switchable Level Range

Another less efficient method to inject amplitude noise also available in some ATE pin electronics is the ability to switch to another level setting during a functional test as shown in Figure 4.16. For differential interfaces, different levels on the two legs might be used to create differential amplitude noise

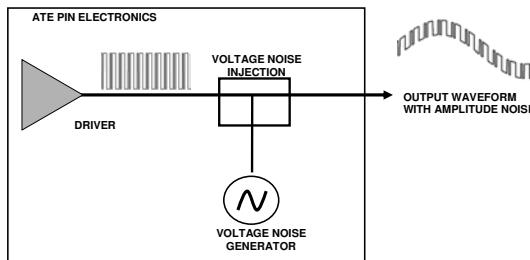


Figure 4.15 Amplitude noise injection through a voltage source.

injection. Timing edge offsets between corresponding edges of the differential legs can force common mode variations. The alternative level control is typically implemented by additional data pattern states. The problem with this approach is that it typically only allows switching between two level ranges and this switch can take a significant amount of time. This makes this approach very limited for testing the DUT receiver for amplitude and common mode noise tolerance.

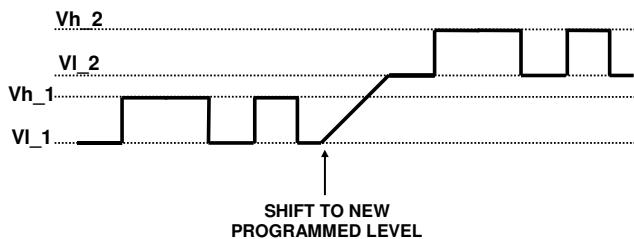


Figure 4.16 Amplitude noise injection through an additional level.

4.1.6 Bidirectional and Simultaneous Bidirectional Support

Support for bidirectional applications can depend significantly on the capabilities of the pin electronics. If the pin electronics architecture is unidirectional then the only available solution to test a bidirectional interface is to use techniques like dual transmission lines [9, 10].

If the pin electronics contain a standard bidirectional architecture like on Figure 4.9 (center) then the pin electronics can natively test a bidirectional application. The only challenge is that due to the long signal traces that are typical on an ATE test fixture, functional testing problems might arise when testing the application with a very short turnaround time between drive and receive. This is exemplified in Figure 4.17 where two examples of a read/write operation between the ATE pin electronics and the DUT are shown.

In the first example, illustrated in Figure 4.17(a), the test fixture delay is much smaller than the time between the read and write operations of the DUT. But in the second example shown in Figure 4.17(b), the test fixture delay is larger than the first example which creates a conflict between the driver and receive operation on the ATE pin electronics. In this case the pin electronics driver must start driving a signal while the receiver is still measuring the write signal from the DUT which will cause the driver signal to be superimposed to the write receive signal. In such a condition, which is described in more detail in Section 9.4, the pin electronics receiver will not be able to measure the write signal correctly.

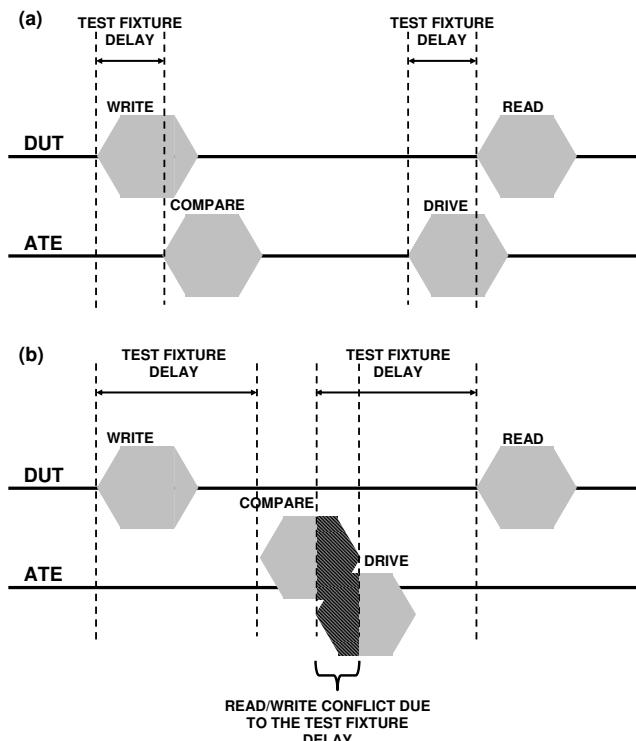


Figure 4.17 Demonstration of the impact of the test fixture delay on testing a bidirectional interface. In example (a) there is no conflict, but in example (b) where the test fixture delay is larger than (a), there is a read/write conflict on the ATE pin electronics.

One way to address this challenge is to use a simultaneous bidirectional (SBD) pin electronics architecture that is shown at a very high level in Figure 4.9 (right). To completely avoid the round-trip time between the pin electronics and the DUT, the ATE pin electronics can drive and receive signals

simultaneously because the receiver has a “copy” of the driver signal which allows it to subtract the signal that is being driven from the signal at the input of the receiver and in this way figure out the signal that was driven from the DUT.

4.1.7 Protocol Engine

Some digital interfaces/standards have complex protocols that have a non-deterministic behavior in some situations; that is, the exact bit pattern to be expected cannot be determined *a priori* (e.g., through simulation). One way to address this challenge is to incorporate the capability to understand the protocol of the application being tested on the ATE pin electronics. Note that this “protocol aware” approach is a significant change with regard to the classical functional test approach with a deterministic pattern. Section 9.6 discusses this topic in more detail.

4.1.8 ATE Loopback Path

An ATE loopback path provides a way for the data bits received by the ATE receiver to be sent to the ATE driver for transmission to the DUT. In this configuration the DUT driver generates a data pattern through some embedded DfT engine and the pattern is then looped back to the DUT receiver through the pin electronics (a more detailed discussion on loopback based testing is done in Chapter 6). In this mode the pin electronics simply loop the signal from the DUT driver to the DUT receiver without having to generate any pattern. In some implementations it is even possible to recondition the electrical parameters of the signal (e.g., rise time, levels, jitter) to stimulate the receiver with controlled levels and/or timing parameters.

4.1.9 Parametric Measurements

Although digital channels historically are mainly seen as being responsible for performing functional tests with simple pass/fail results or for DC/scan type measurements, they can also perform detailed parametric measurements for level and timing parameters such as jitter histograms, device level measurements, or even full data eye diagrams. The capability of digital channels to perform this kind of parametric measurements for certain applications is mainly determined by their level and timing accuracy as well as the available methodologies for data post-processing. Besides the specified timing and level accuracy, the bandwidth of the pin electronics also has significant influence on the result of parametric measurements.

On a typical digital ATE pin electronics, the user has separate control of driver and receiver voltage and timing settings. Depending on the specific architecture of the ATE pin electronics, there might be some limitations (e.g., a true differential receiver might not have a variable threshold voltage).

For a standard functional test, the objective is to verify that the DUT reacts in the expected manner to a stimulus applied to its input pins. The stimulus and expected data patterns are typically derived from device simulations. For the execution of a functional test, the device input levels and timing parameters are set according to the requirements of the device specification. In order to check the DUT output pattern, the receiver voltage threshold is set to a value that allows distinguishing between the high and low output levels as defined by the DUT specifications. The compare strobe timing is set to occur in the center of the received DUT bits to guarantee the maximum possible timing margin.

In order to perform parametric measurements for level or timing parameters or both, the respective ATE control (drive levels, drive timing, compare level threshold, or compare strobe timing) is not set fixed to the optimum that guarantees maximum margins with respect to the DUT signal but is varied for iterative functional test executions. A change in the functional test result from one iteration to the next indicates that the test decision point that is controlled by the varied ATE resource has been reached. Using the knowledge of the values set for this ATE resource for both iterations that caused the test result transition allows the reconstruction of the respective signal value. An example for such a parametric level measurement is shown in Figure 4.18 for an output voltage measurement.

It is important to note that for output pins, such measurements often do not necessarily require a functionally passing test because it does not matter whether the test result transition for two iterations is from pass to fail or from fail to pass. One important condition for these kind of parametric measurements, however, is that the behavior of the DUT is repeatable and deterministic for a sequence of functional test executions. Due to the required test iterations to determine the parameters for a single measurement point, special care should be taken to the length of the test pattern executed per iteration since this will directly impact the required measurement time.

The extent to which this methodology can be used for parametric measurements strongly depends on other features that are available for the ATE receiver used. A basic functional test typically delivers a single pass or fail result for all bits that are compared within a test pattern. Thus, a parametric measurement that is performed as described before using such a typical single pass-fail evaluation over all bits will deliver a result that represents the overlay of all bits of a pattern. In order to obtain the parametric values for one or more

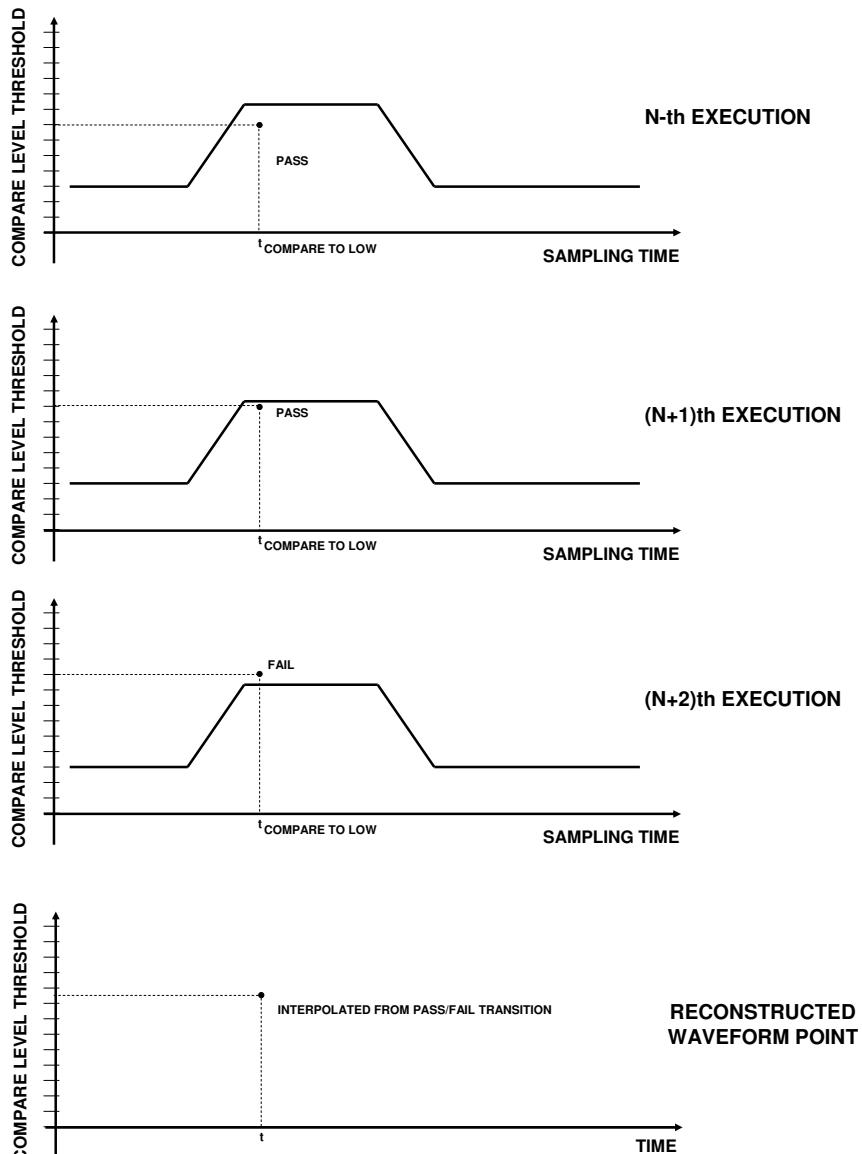


Figure 4.18 Acquiring the waveform using a strobe time/level threshold sweep approach.

dedicated bits of the executed pattern, multiple executions of the parametric measurements would be required with only a single active compare at the desired bit positions if the ATE receiver only supports the classical single pass-fail reporting per pattern execution.

A significant improvement for parametric measurements is represented by ATE receivers that allow collection of and access to the pass-fail information per active compare strobe instead of the global pass-fail result only. Such a feature allows analysis of the parametric values on a per-bit basis as shown in Figure 4.19 for a per-bit level measurement without the need to repeat the measurement sequence per bit with an active compare at the bit of interest only.

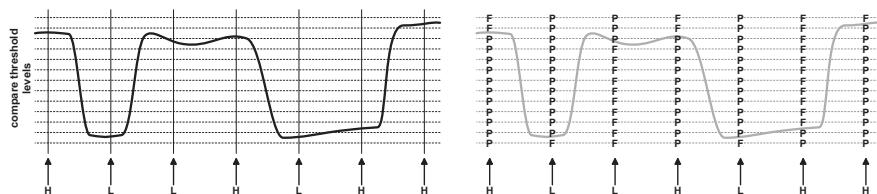


Figure 4.19 Using digital pin for parametric level measurements.

A further feature of ATE receivers that extends the described parametric measurement capabilities is a per-pin error counter which delivers the number of errors for a pin over one complete pattern execution. If the number of errors per pin is evaluated for each test iteration with varying compare level or timing settings, the distribution of the received DUT signal over the iteration range of the level or timing is obtained. This distribution data can be used for statistical analysis of level or timing data.

Although the previous examples mainly concentrated on parametric level measurements, it has to be emphasized that the same approaches are also usable for parametric timing measurements if the ATE timing is varied instead of the level parameters.

An interesting point of this parametric measurement methodology is that if it is applied for level measurements it can be seen as being equal to using a 1-bit analog digital (A/D) converter. Thus, the same sampling methodologies as for sampler and digitizer instruments that will be described in Section 4.3.3 are applicable with digital receivers to sample analog signal waveforms. The accuracy and resolution of this approach depends on the bandwidth of the ATE receiver, the timing/level accuracy, and the timing resolution on the ATE edge control.

4.2 Sampler/Digitizer ATE Card¹

In ATE systems for mixed signal applications, there are typically two kinds of analog measurement instruments available. One is called a digitizer and the other is called a sampler. Both of them use analog to digital converters (ADC) to convert the analog signal waveform at their inputs into a digital representation of that waveform. In order to understand the difference between both instruments and to select the right instrument for an application, it is important for a test engineer to understand how analog signals are acquired by mixed signal instruments and how the characteristics of these signals are influenced by the data acquisition.

The fundamental rule to obtain the correct waveform representation after sampling the analog waveform with fixed sampling intervals is Shannon's theorem that can be traced back to the sampling limit according to Nyquist [11]. Shannon's theorem states that if a signal over a given period of time does not contain any frequency component greater than f_x , all of the needed signal information can be captured with a sampling rate of $2f_x$. The sampling rate of $2f_x$ is referred to as the Nyquist rate. The corresponding time interval $(1/2)f_x$ is called the Nyquist interval. Strangely enough, the corresponding Nyquist frequency is f_x . If a signal is not sampled according to the rules of Shannon's theorem, an effect called aliasing falsifies the sampled signal.

4.2.1 Aliasing

In Figure 4.20 a periodic signal is sampled with timing sample intervals that are larger than half of the period of the signal itself. Thus, the used sampling interval is larger than the Nyquist interval and Shannon's theorem is violated by this fact. Such a kind of sampling also is called undersampling. If the data samples that should represent the original signal are analyzed without the knowledge of the original signal, the waveform reconstructed from the data samples has a lower frequency than the original signal. In the frequency domain, aliasing folds the frequencies that are beyond the Nyquist frequency back into a frequency band below the Nyquist frequency with a distance to the Nyquist frequency that is the same as the distance of the original frequency to the closest integer multiple of the Nyquist frequency.

While aliasing is used sometimes on purpose to measure signals beyond the sampling capabilities of an instrument in an undersampling configuration, it is usually avoided if possible. If undersampling is used, special care has to be taken that the alias effects are taken into account during data analysis.

¹In collaboration with Hideo Okawara.

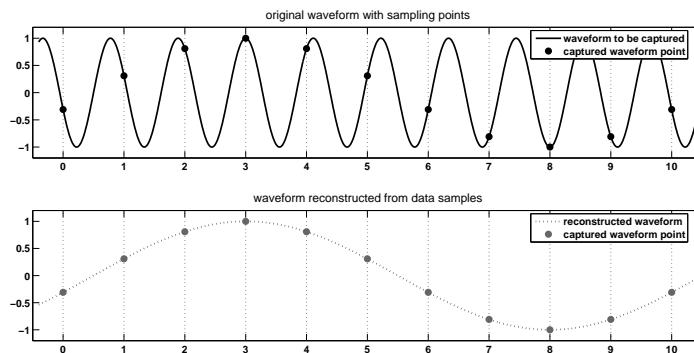


Figure 4.20 Signal aliasing for sampled data.

4.2.2 Digitizer

In order to avoid aliasing effects due to frequency components (e.g., noise components) beyond the Nyquist frequency, digitizers limit the bandwidth of their input signals. Figure 4.21 shows a typical block diagram of a digitizer.

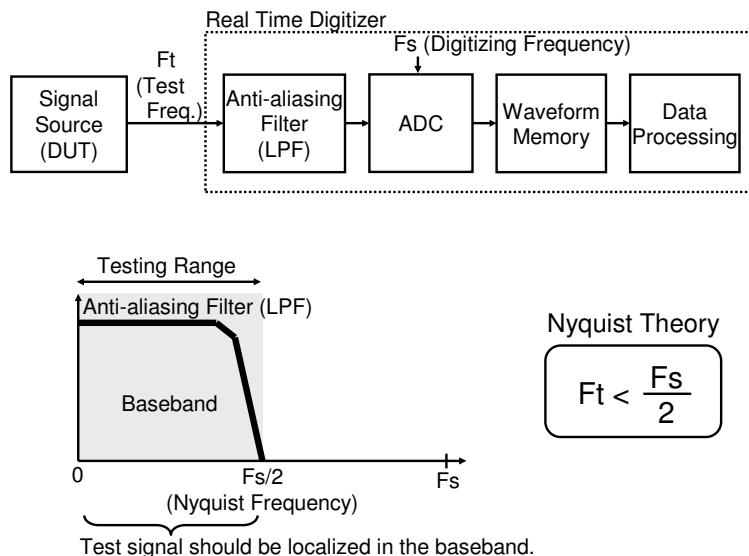


Figure 4.21 Digitizer block diagram.

The input signal first is fed through a lowpass filter (LPF) called the antialiasing filter. The band-limited signal after the antialiasing filter is digitized by the ADC. The sampling frequency of the digitizer that is

controlled by the test engineer should be greater than twice the cutoff frequency of the antialiasing filter to conform with Shannon's theorem. The digitized data array is stored in the waveform memory of the ATE system. The data captured into the waveform memory is then processed using digital signal processing (DSP) methodologies to derive required test parameters. The analog input bandwidth of a digitizer is specified by the antialiasing filter and the real-time sampling frequency of the ADC.

In order to avoid aliasing within the full sampling rate range a digitizer supports, the antialiasing filters that are used typically have a programmable bandwidth. This allows a user to adapt the antialiasing bandwidth to the sampling rate that was chosen for a certain measurement to avoid aliasing effects. The maximum antialiasing filter bandwidth usually is matched to the maximum sampling rates that are possible with the high resolution ADCs used in the digitizers in a way that Shannon's theorem cannot be violated. This, however, usually limits the bandwidth of digitizers in a way that they are not well suited for testing high-speed I/Os. As a general rule, a test engineer can assume that digitizers take care of the signals that conform to the Nyquist requirements.

4.2.3 Sampler

While the maximum frequency of a signal that the digitizer can measure is limited by its ADC sampling rate and the resulting maximum antialiasing filter bandwidth, a sampler does not have this constraint regarding analog input bandwidth. Figure 4.22 depicts a sampler block diagram and its frequency domain coverage. The difference between the digitizer is that a sampler has no antialiasing filter. Instead, a sampling head with a track-and-hold stage is deployed at its front end. The analog input bandwidth of the sampler is specified by the performance of the sampling head, which is usually several GHz. This enables sampler instruments to test high-speed I/Os that operate in the multigigabit range.

Of course, the ADCs used in samplers usually cannot implement real-time sampling rates that conform to Shannon's theorem. This is especially true if parametric measurements on the captured waveforms are in the focus where sampling intervals of few picoseconds might be required. Thus, high-speed I/O data capturing with samplers uses the knowledge about the expected data rate of the signal to be sampled and allows aliasing effects to happen for the sampled raw data. In a post-processing step, the aliasing effects are corrected and the original DUT signal is reconstructed.

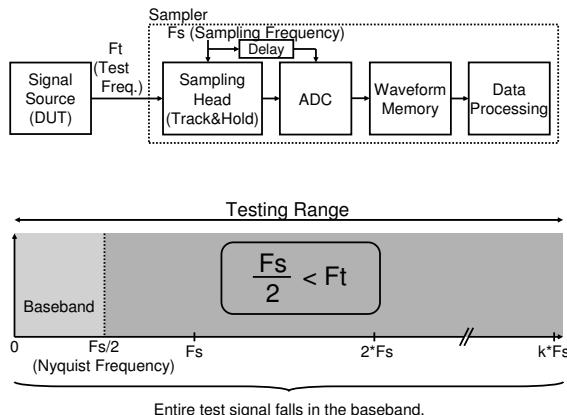


Figure 4.22 Sampler block diagram.

4.3 Parametric Measurements with Sampled Data²

The heart of a digitizer/sampler is a multibit resolution ADC that allows a direct transfer of the analog waveform voltage level into a digital representation within a single test pattern execution. As described in Section 4.1.9, digital pin electronics that provide a pass/fail result for certain level conditions can also be used to measure the same digital voltage level representation of a DUT signal with repeated test pattern executions and a different compare threshold setting for each test pattern iteration. Thus, although this section describes the fundamentals on how a digitizer or sampler can be used to acquire and measure a digital waveform, the same principles also can be applied on analog data captured by a digital channel into a digital waveform representation. Note that we will not discuss how sampler/digitizers are utilized in mixed-signal applications since it is not in the scope of this book. Nevertheless the reader should be aware that there is a significant additional body of theory related to sampler/digitizer usage [11, 12] that is not considered in this section.

4.3.1 Undersampling of High-Speed I/O Signals

As mentioned before, the maximum sampling rates of state-of-the-art sampler instruments are not sufficient to acquire multiple samples per UI of a high-speed I/O signal. In other words, sampler instruments cannot acquire high-speed I/O signals in real-time with the timing resolution required for

²In collaboration with Hideo Okawara.

parametric measurements. In order to be able to perform such measurements despite this limitation, samplers use undersampling.

Undersampling can be applied to periodic patterns only. The required samples of the signal that are taken during the different iterations of the periodic pattern are then folded back in a post-processing step into the unit test period (UTP) that represents the length of the fundamental pattern that is repeated. This post-processing step reconstructs one representation of the fundamental pattern. Since the sequence of the raw samples is not representing the sequence of the samples with regard to their phase location inside the UTP, the post-processing step that reconstructs the UTP has to reorder the samples along their appropriate phase within the UTP. The phase difference between subsequent samples in the reconstructed UTP defines the effective sampling rate that is much higher than the physical sampling rate and virtually can be increased to any value required within the accuracy limits of the sampler itself. The sample reshuffling operation is key for undersampling applications and usually is supported by ATE systems by a dedicated application programming interface (API).

For example, if a sampler is running at the sampling rate of 148.51 MHz and digitizes a 1.0-GHz sinusoidal waveform, the captured data and the phase offset of the samples with respect to the UTP looks as in Figure 4.23. One UTP in this example matches exactly one cycle of the original sine wave. There are 15 sampling points in the UTP and exactly 101 cycles of the sine waveform are captured. Since 15 is not an aliquot³ of 101, each of the sampling points hits a different phase for each cycle of the UTP. By reshuffling each sampling point according to its phase offset with respect to the UTP, a single cycle of precise sinewave can be reconstructed, as shown in Figure 4.24. The effective sampling rate achieved in this example would be 15 GHz.

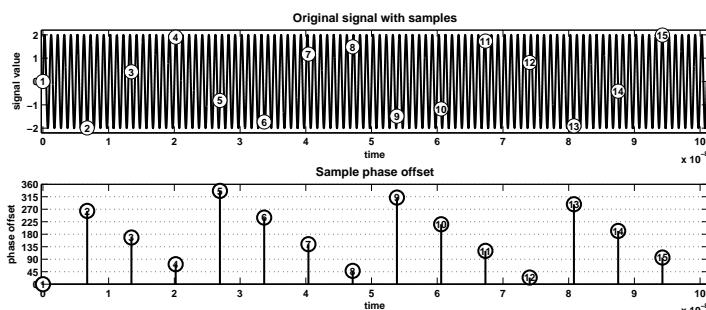


Figure 4.23 Original waveform with data samples and sample phase.

³An aliquot (or aliquot part) of an integer is any of its integer proper divisors. For instance, 2 is an aliquot of 12.

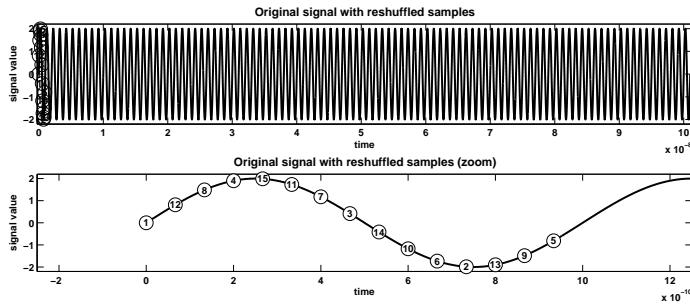


Figure 4.24 Reshuffled data and reconstructed UTP.

In the previous example the original signal was sampled to reconstruct a single UTP. Undersampling, of course, can also be used to sample the data for more than a single reconstructed UTP. In order to avoid phase jumps at the UTP boundaries of the reconstructed signal, the parameters used for sampling the data in this case have to fulfill the requirements of the coherency equation described in the following section.

4.3.2 Coherency Equation

Coherency is the key for successful undersampling measurements in digitizer/sampler applications. In order to have a coherent condition between the signal to be sampled and the sampling rate, the requirements defined by (4.1) have to be fulfilled [13].

$$\frac{F_t}{F_s} = \frac{M}{N} \quad (4.1)$$

In this equation, F_t is the frequency represented by the UTP of the signal to be measured and F_s is the sampling frequency of the sampler. M is the number of captured UTP cycles and N is the number of acquired samples per UTP. Thus, the effective sampling rate is obtained by the division of N with the length of one UTP. M and N must be integer numbers, and should be mutually prime to get maximum signal information with a minimum of samples.

In order to provide more freedom for the selection of the sampling frequency F_s , the coherency condition can be extended according to (4.2). As with the standard coherency equation M_x is the aliased number of cycles into the baseband. For the extended coherency equation, M_x should be between $-N/2$ and $N/2$. Usually N is selected as a power of two to allow a direct FFT of the reconstructed signal. Thus, M_x usually is an odd number to maintain the guideline that N and M_x should be mutually prime. The new term K in the

extended coherency equation determines how the frequencies of the signal to be measured are folded back into the baseband in dependency of the sampling frequency F_s for the selected coherency conditions. In the following we will call K the frequency zone number:

$$\frac{F_t}{F_s} = K + \frac{M_x}{N} \quad (4.2)$$

From the frequency domain's point of view, the coherency (4.2) can be represented as the folded spectrum in Figure 4.25. If $M_x > 0$ the signal is located in the front page. If $M_x < 0$ the signal is located in the back page. Consequently $|M_x|$ indicates where the aliased signal would fall in the baseband. When the parameters for undersampling are selected, this information is used to set up the spectrum of the reconstructed signal in accordance with the application needs.

As one can see in Figure 4.25, a certain frequency range of the signal to be measured falls into a certain frequency zone for a given sampling frequency F_s . If the sampling frequency is changed, the frequency zones also change relative to the signal to be measured. In other words, the frequency range to be measured can be covered by multiple combinations of frequency zone numbers and sampling frequencies. This additional freedom allows a test engineer to adjust M_x in a way that assigns the favorite frequency bin location to the target signal by selecting the appropriate F_s and K combination that supports the desired M_x .

4.3.3 Capturing Digital Waveforms

Let's start by analyzing how a digitizer captures a waveform. Figure 4.26(a) shows a clock waveform with a frequency F_t . Three cycles of this clock waveform are digitized with 16 points of data so that the frequency condition is $F_t/F_s = 3/16$ and $M/N = 3/16$, where F_s is the sampling (digitizing) rate, M is the number of test signal cycles, and N is the number of data sampling points. The captured data consists of three cycles of the primitive waveform as Figure 4.26(b) shows. The samples are reshuffled as described earlier so that a single cycle of the precise waveform is obtained as shown in Figure 4.26(c).

Moving to a sampler in undersampling condition, the same waveform, for example, would be sampled over 35 cycles with 16 points of data, as shown in Figure 4.27(a). The frequency condition in this example can be described as $F_t/F_s = 35/16 = 2 + 3/16$, with $M_x = 3$. This setup samples three cycles of the primitive waveform as shown in Figure 4.27(b). Reshuffling the three-cycle waveform can replicate the single cycle of the original waveform shown in Figure 4.27(c).

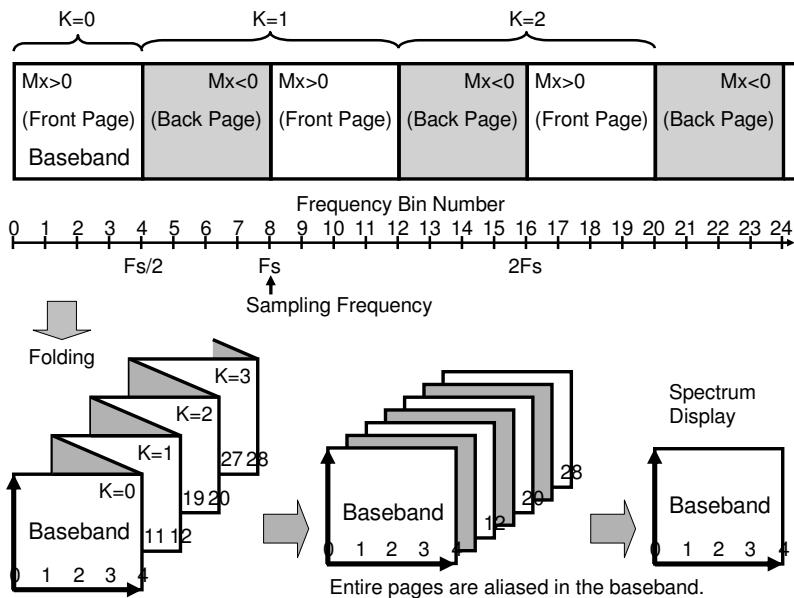


Figure 4.25 Coherency equation versus folded spectrum.

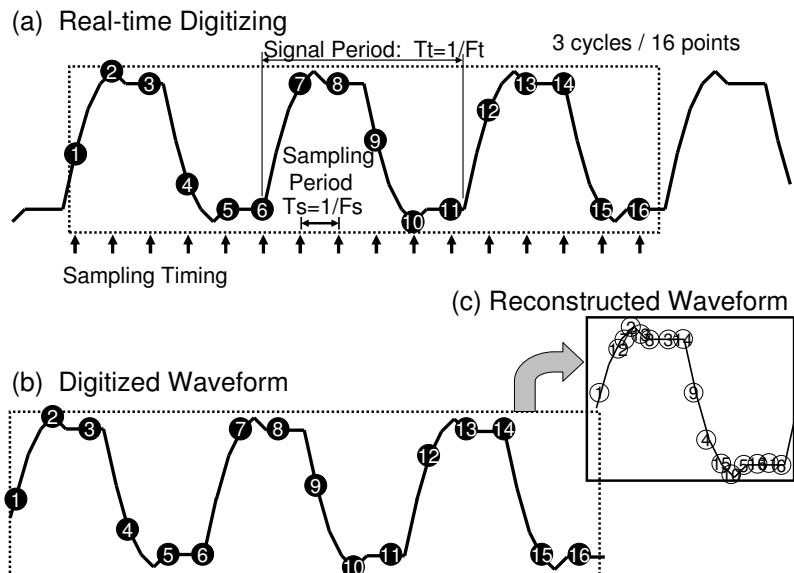


Figure 4.26 (a–c) Waveform sampled by a digitizer $F_t/F_s = 3/16$.

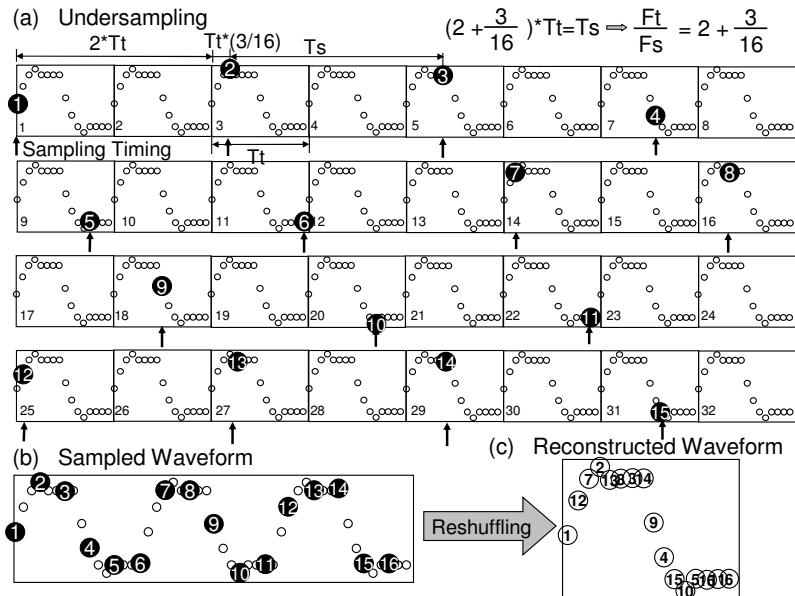


Figure 4.27 (a–c) Undersampling $F_t/F_s = 3/16$.

Figure 4.28 shows the case of a 31-cycle waveform that is sampled with 16 points, that is, $F_t/F_s = 31/16 = K + M_x/N = 2 - 1/16$ so that $M_x = -1$. The sampled data directly replicates a single cycle of the original waveform. However, it appears inverted. When M_x is negative, the waveform is reconstructed inverted. This does not pose a problem for analog signals which in most cases are analyzed in the frequency domain using spectrum analysis methodologies. The signal reversion does not affect the signal spectrum. However, if such sampling conditions with negative M_x values are applied to digital signals, the reconstructed signal has to be corrected by changing the index of the waveform data array to obtain correct measurement results.

In real digital applications, usually longer pattern sequences than just the digital clock cycle considered in our previous description need to be captured for certain parameter measurements. Such real life sequences, for example, are PRBS patterns or the PCI Express compliance pattern. Coherent undersampling can be used also for these kinds of patterns as long as they are periodic and a DUT can be operated in a way that the desired pattern is generated repeatedly. In such a case F_t represents the frequency of the pattern repetition and the UTP represents the length of one complete pattern sequence that is repeated. Since coherent sampling increases the effective sampling rate at the cost of the number of required pattern repetitions, coherent sampling of

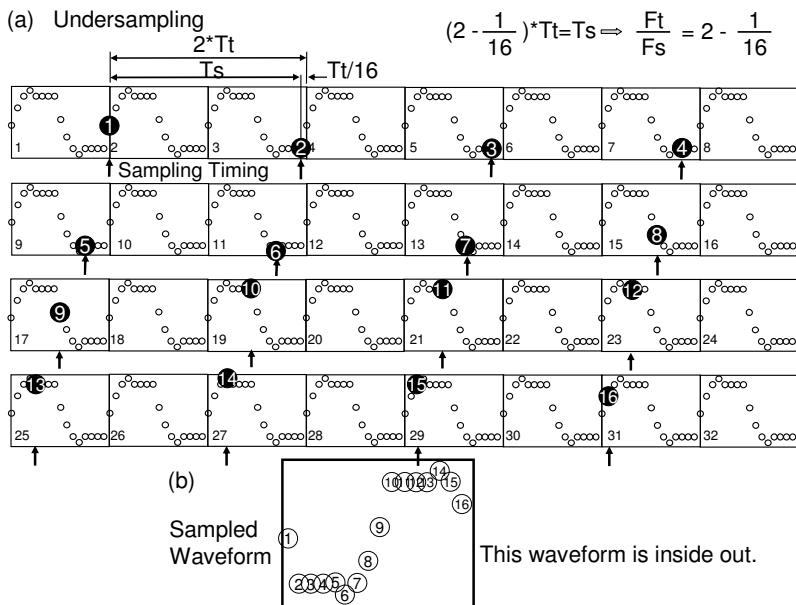


Figure 4.28 (a, b) Undersampling $F_t/F_s = 31/16$.

long pattern sequences with high resolution can increase the required pattern execution time significantly.

Example Let's assume that one wants to acquire one PRBS7 sequence for a 5-Gbps signal. The execution time for a single PRBS7 sequence consisting of 127 bits would be 25.4 ns. This time represents one UTP in our case and thus results in $F_t = 39.37$ MHz. If this PRBS sequence is to be sampled with a 10-ps resolution, $N = 25.4\text{ ns}/10\text{ ps} = 2,540$ sampling points are required. If M is selected to be 2,541, the pattern run time for the sampling execution would be about 64.5 μ s. If more than one PRBS sequence is to be sampled this pattern run time would increase accordingly.

4.3.4 Special Considerations for Coherent Sampling with Digital ATE Channels

The advantage of coherent sampling with digital ATE channels versus compare strobe movements in sampling resolution increments via the ATE timing system is the fact that a single functional test delivers data for all sampling points regardless of the sampling resolution. Of course, this comes at the cost of longer pattern execution times because the number of required

waveform repetitions depends on the desired sampling resolution according to the coherence formula. However, executing a single functional test with a longer pattern typically is faster than executing multiple functional tests with timing changes in between. Also, coherent sampling allows high resolution sampling beyond the resolution of the timing system resolutions and without the negative influence of delay line nonlinearities.

Since the coherency requirement with M and N being prime to each other means that also F_t and F_s do not have friendly fractions, digital ATE channels that are used for coherent measurements need to run in different clock domains with a frequency offset to the channels that stimulate the DUT.

4.4 Power Supplies

ATE power supplies, also referred to as device power supplies (DPS), not only provide the needed voltage/current to the DUT but also typically offer the ability to measure important DUT parameters like the quiescent current (IDDQ) or the steady state current (IDD). In terms of requirements, power supplies are typically divided into subcategories like high pin-count, low noise, high current, and so on.

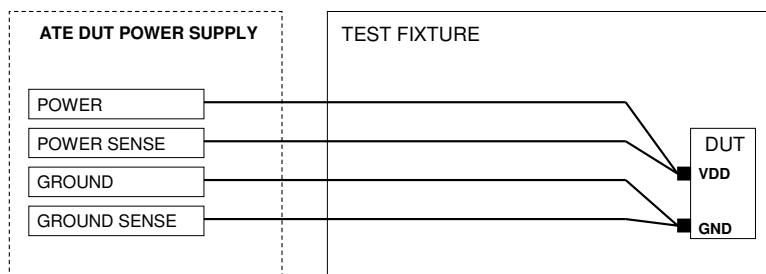


Figure 4.29 Block diagram describing the force/sense connections between an ATE DUT power supply and the DUT.

High-current power supplies typically have to provide currents in the tens or hundreds of amperes since some applications like a microprocessor might require very high current values [14]. If a single power supply channel cannot provide the required amount of current, there is the option to gang multiple channels to obtain the required current value.

Another possible classification of DUT power supplies in ATE systems is between analog and digital power supplies: analog power supplies are intended for analog application or mixed signal application where a very low noise floor is needed, while digital power supplies are intended for digital applications where the noise floor requirements are not as high.

An ATE power supply contains two sets of connections for the DUT as shown in Figure 4.29. One is the power/ground pair (sometimes named force pair) that provides the power to the DUT. The other is the power and ground sense connections. These sense connections are used by the power supply control circuitry to measure the voltage at the DUT.

If the value at the DUT is different from the programmed value (note that the voltage at the DUT might be different from the voltage at the power supply output in certain time instants), the power supply control circuitry will try to get the measured value equal to the programmed value by increasing/decreasing its own output voltage [15]. Because of this it is important that the power and sense lines are connected together as close as possible to the DUT power terminals. However, it is also important to note that this feedback circuit is slow and cannot compensate for the fast changes at the DUT power pins due to transistor switching. This is one of the design targets of the power distribution network (PDN) on the test fixture which is discussed in Section 8.11.

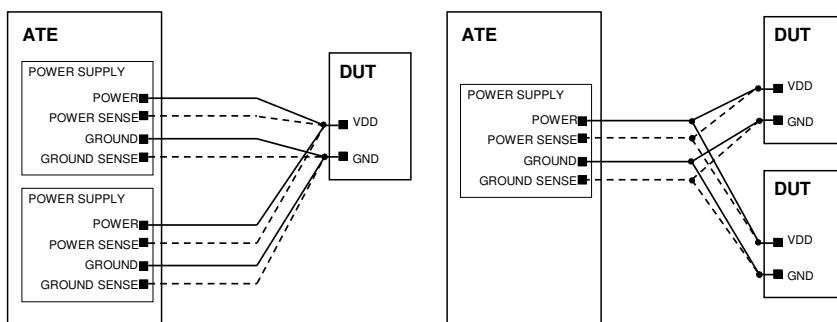


Figure 4.30 Block diagram describing the force/sense connections for two power supplies connected to a single DUT (left) and one power supply powering two DUTs (right).

In some situations it is necessary to use multiple power supplies to provide the required current to a DUT (power supply ganging) while in other situations like on a multisite application one power supply is used to power multiple DUTs. Both of these options require proper connections of the power supplies' power and sense lines as illustrated on Figure 4.30. It is also important in both situations that the force lines ohmic loss is as balanced as possible.

References

- [1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Springer, 2000.
- [2] I. A. Grout, *Integrated Circuit Test Engineering*. Springer, 2006.
- [3] R. Rajsuman, N. Masuda, and K. Yamashita, “Architecture and Design of an Open ATE to Incubate the Development of Third-Party Instruments,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, Oct. 2005.
- [4] J. Kelly and M. Engelhardt, *Advanced Production Testing of RF, SoC and SiP Devices*. Artech House, 2007.
- [5] J. Moreira, G. Haensel, and F. Koban, “Addressing the Challenges of Implementing an At-Speed Production Test-Cell for 10Gb/s Wafer Probing,” *DesignCon*, 2005.
- [6] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [7] W. Maichen, *Digital Timing Measurements*. Springer, 2006.
- [8] U. Shankar, “Test Challenges for SONET/SDH Physical Layer OC3 Devices and Beyond,” *IEEE International Test Conference*, 2001.
- [9] D. C. Keezer and Q. Zhou, “Alternative Interface Methods for Testing High-Speed Bidirectional Signals,” *IEEE International Test Conference*, 1998.
- [10] J. Moreira, H. Barnes, and H. Werkmann, “DUT Loadboard Layout for Multi-Gigabit Bi-Directional Interfaces,” *Verigy Users Conference*, 2007.
- [11] M. Burns and G. W. Roberts, *An Introduction to Mixed-Signal IC Test and Measurement*. Oxford University Press, 2000.
- [12] M. Baker, *Demystifying Mixed Signal Test Methods*. Elsevier Science, 2003.
- [13] M. Mahoney, *DSP-Based Testing*. IEEE Computer Society Press, 1987.
- [14] K. Aygün, M. J. Hill, K. Eilert, K. Radhakrishnan, and A. Levin, “Power Delivery for High-Performance Microprocessors,” *Intel Technology Journal*, vol. 9, Nov. 2005.
- [15] D. Gizopoulos, *Advances in Electronic Testing: Challenges and Methodologies*. Springer, 2006.

5

Tests and Measurements

This chapter presents a discussion on the typical tests associated with the driver and receiver parts of a high-speed digital interface. It is important to understand that all of these tests have one objective, to guarantee that the number of errors on a digital interface transmission will be below a certain threshold. Any digital interface has the possibility of having a transmission error at some point in time (independent of how good the physical layer design is). This is due to the physics associated with semiconductor devices and transmission over a nonideal medium. Note that at higher layers of the transmission protocol, some of these errors can be corrected by error correction methodologies embedded in the protocol. But there is always a price to be paid since more robust error correction methodologies imply a loss on bandwidth.

5.1 Bit and Pattern Alignment

As we have already described in Section 2.6, one of the key building blocks for high-speed I/O drivers are PLL circuits. One characteristic of high-speed I/Os that use PLLs to generate their high-speed clocks is that the phase offset between the reference input to the PLLs and the high-speed clocks generated by the PLLs that serve as timing reference for the high-speed I/O data of the device are typically not deterministic. Thus, the latencies between the reference clock or other ATE stimuli provided to the DUT and the high-speed signals generated by the DUT are not deterministic. The problem with this unknown phase and latency relationship is twofold.

The first aspect is the uncertainty where the compare strobe edges of the ATE system are located in relationship to the data bits the ATE comparator receives from the DUT (bit uncertainty). Without taking care of this bit

uncertainty it is possible that the ATE compares the received data bits in an unsafe area at or close to the bit transitions.

The other aspect is the uncertainty for the pattern offset between the data pattern generated by the DUT and the sequencer controlled compare vector data on the ATE system (pattern uncertainty) as shown in Figure 5.1.

The problem is further exacerbated by the fact that the latency between the stimuli of the DUT and the reaction of the DUT to these stimuli usually is not predictable exactly and might vary each time the DUT is powered up. The same effect might be observed in the case of a re-start of the PLL input (e.g., the reference clock or the clock extracted from the incoming data) which also could change the phase and latency the device introduces between incoming clock/data and outgoing data to the ATE system.

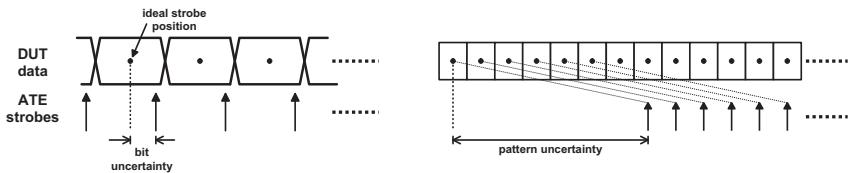


Figure 5.1 The alignment challenge for correctly performing a functional test.

The most straightforward way to solve this challenge is to perform a timing search on the ATE side until one finds the exact timing delay between the signal generated by the DUT and the ATE pin electronics receivers. Although this procedure is straightforward, it can be extremely time consuming because the pattern uncertainty that sets the requirements for the timing search range can be large while the bit uncertainty forces search steps that are significantly smaller than a single bit time (a single UI). In some situations the timing architecture of the ATE system might not even be able to handle the large delays associated with the pattern uncertainty.

Thus, a more general and faster way to overcome this alignment challenge is to compensate bit uncertainty and pattern uncertainty in two separate steps. In the first step, the bit uncertainty is compensated by bit alignment and in the second step, pattern alignment compensates for pattern uncertainty.

In a system environment, these uncertainties are usually handled by training sequences that usually have to be applied between two link partners as part of the power-up sequence or after the change of operating conditions. Although the details of this training vary between high-speed I/O standards (e.g., PCI Express, GDDR5), the fundamental target of the training sequences is to achieve bit and pattern alignment to guarantee correct link partner communication with maximum signal timing margins.

5.1.1 Bit Alignment

As mentioned before, bit alignment is the task of placing the ATE's strobe edge timing in the center of the data eye. The resulting values for this timing shift are less than one UI or bit time interval as the expression "bit alignment" implies. The first step in this regard is to identify the phase of the incoming data stream or, in other words, identify where the bit transitions in the incoming data stream occur relative to the ATE timing. Once this phase relationship is identified, the timing of the ATE comparators is offset by half a UI from the identified phase alignment point to set up the compare timing to the center of the received data eye.

ATE systems offer several ways to accomplish bit alignment. Timing search based approaches search either for a single bit transition or for the transitions of multiple bits to identify the bit boundaries. A more accurate bit alignment is possible with an approach that scans the bit error rate (BER). Finally, some ATE platforms offer integrated hardware-based bit alignment.

5.1.1.1 Single Bit Transition Search

The standard way to implement bit alignment on ATE is based on timing searches. Since a global fail-to-pass (or pass-to-fail) search with a compare to the full expected pattern only yields a result if potentially existing pattern offsets are also covered by the timing search range, this would lead to the problem of a large search range with a high timing resolution as mentioned above. In order to be able to use a timing search with a minimum search range, the compare pattern is set up with a single compare to high or low. With such a timing search setup, the closest bit transition in the direction of the search will be identified on the received pattern regardless of a potential pattern offset between received and expected data. For this search, a single compare threshold is set to the center of the expected level range in order to get the functional test result transition as good as possible at the center of the signal waveform transition as shown in Figure 5.2.

Taking into account that the used compare pattern only compares for a single bit within the received data stream, it is obvious that the search result obtained for the phase relationship between the DUT and the ATE system does not take into account the timing jitter and amplitude noise of the received signal. Thus, the accuracy achieved with this kind of phase identification is relatively low and might be acceptable for functional testing only.

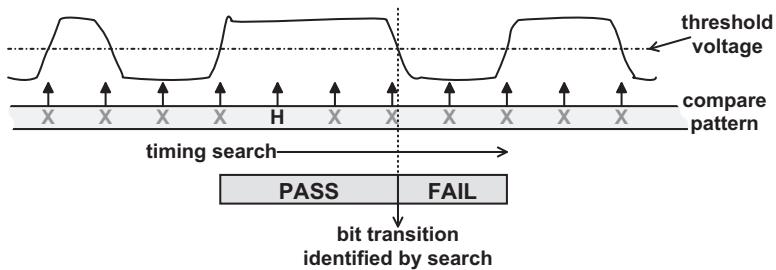


Figure 5.2 Bit alignment using a transition search.

5.1.1.2 Multiple Bit Transition Search

One improvement of the result accuracy for a purely pass-fail based search would be to use a pattern that consists of a pattern sequence that is repeated by the DUT. Representatives for such patterns are, for example, the PCI Express compliance pattern or a PRBS pattern generated by the DUT. Such a pattern can hide the pattern uncertainty from the search to some extent because the compared pattern is reoccurring latest after the length of the repeated pattern sequence.

With this, a full compare to the pattern sequences is possible (with some mask margins at the start and the end of the pattern) for the search with a search range that corresponds to the length of one of the pattern sequences used. The extreme example for a pattern with these characteristics would be a bit clock pattern that consists of the pattern sequence 01. It is obvious that a pass-fail based search will yield a search result within a search range of two UIs even if the compare on the ATE is done for more than just a single bit. With this technique it is possible to take at least some of the jitter components into account that will be present with the functional pattern to improve the phase search accuracy.

5.1.1.3 BER-Scan

A more sophisticated and accurate phase identification using a search approach can be accomplished, if the result is not just a simple fail-to-pass or pass-to-fail decision but collects the functional test errors observed for every search step. Such a measurement implementation delivers constant error count numbers for consecutive search steps inside the data eye. If the search steps are in the range of a data transition, the error count will vary from search step to search step.

Using error count results during the search has the advantage that the bit alignment can happen on a real functional pattern with a minimum search

range or, more correctly, with a minimum error count scan range. Another advantage is that the full jitter characteristic of the received data pattern is obtained and can be analyzed. With an ATE channel that has the ability to measure the number of functional test errors for a single functional test, the compare timing is moved over a range of two UIs and for each timing step the number of errors for each pin to be analyzed is recorded.

The recorded error count signature (error density) per pin will show a variation of the error count over time regardless of what data is compared to. Note that we will not have a zero error region since we so far do not have a pattern alignment, but in the timing edges of the data eye the error density will vary due to jitter. Also the stable error count regions will vary in the number of errors observed due to the movement of the compare strobes for the same compare data bits from one data eye into the next data eye. The maximum values of the derivative of this error density provide the right and left edges of the data eye. The optimal compare strobe timing (with regard to the ATE timing period) will be in the center of these two points as shown in Figure 5.3.

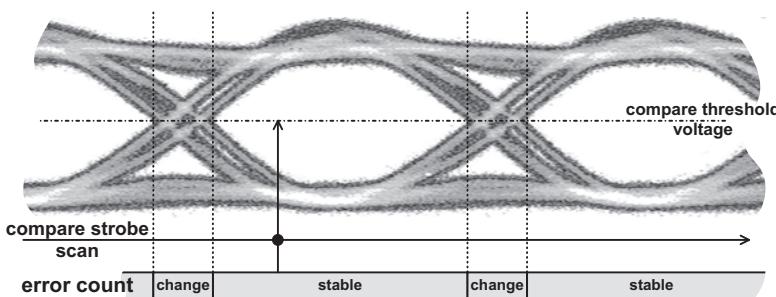


Figure 5.3 Bit alignment with measurement of the error density.

5.1.1.4 ATE-Integrated

In order to simplify this task for the test engineer, the latest high-speed ATE pin electronics channels have embedded support (phase tracking and/or CDR) for the identification of the phase of an incoming data stream as described in Section 4.1.1. In addition to the identification of the phase, these channels usually also provide an automatic phase adjustment capability, which aligns the compare strobe edge to the center of the received data eye by setting an automatically calculated or user-defined timing offset to the identified data stream phase.

5.1.2 Pattern Alignment

After a successful bit alignment, it is necessary to perform a pattern alignment to obtain a functionally passing test. As for the bit alignment, some ATE cards have embedded support to achieve this task [1]. If such a feature is not available, there are several other possibilities to achieve pattern alignment. Pattern alignment using match loop constructs is one of these. Of course, also an alignment based on timing searches is possible in some cases. When timing search and programming ranges are not sufficient to cover the expected pattern offset, for some ATE systems an alignment using controlled ATE sequencer offsets is possible.

For pattern alignment, the main challenge is not necessarily the identification and compensation of the pattern offset between the DUT and the ATE system itself. A bigger challenge usually is to ensure that the relative timing offset between the DUT pattern start and the start of the sequencer(s) of the ATE stays constant for repetitive pattern executions that might be required to quantify the pattern uncertainty.

5.1.2.1 Match Loop

The classical way for ATE to adjust to unknown pattern latencies is the use of a match loop. A match loop is a feature of some ATE systems where the user defines a pattern that the ATE system loops around until it gets a functional pass. After this functional pass, the ATE will execute the target pattern that is used for the final pass-fail decision.

The advantage of such a loop construct is that the ATE sequencer(s) keep running and therefore no uncontrolled pattern offset change originating from sequencer restarts occurs. In order to achieve pattern alignment, a match loop is set up that compares to the pattern expected from the external pattern source (external instrument or DUT) plus one potentially masked compare bit as illustrated in Figure 5.4. Thus, a sliding compare window on the ATE is generated that advances by one bit relative to the pattern generated by the DUT with each match-loop iteration as shown in Figure 5.4.

It is obvious that at one of the loop iterations, the expected data matches the received data. If the match loop is left right after the matching loop iteration, pattern alignment is achieved [2]. It is important to note that a necessary requirement for the application of a match loop is the ability to implement a pattern offset between expected and received data in the match-loop body (in Figure 5.4 this offset is implemented by the additional compare bit). While this is not an issue when comparing to bit streams that run freely, like, for example, data originating from an external instrument or stimulus data generated autonomously by the DUT (e.g., via PRBS or other internal

pattern generators), it can be a challenge if there is a one-to-one relationship between the stimulus data sent to the DUT and the data received from the DUT.

The problem in such a case is that both stimulus and compare data for the DUT are defined inside the match-loop body. If there is a fixed relationship between stimulus and compare data, the required pattern offset to implement the sliding compare window cannot be set up because either the stimulus or the compare data would have to be changed for each match-loop iteration for this purpose.

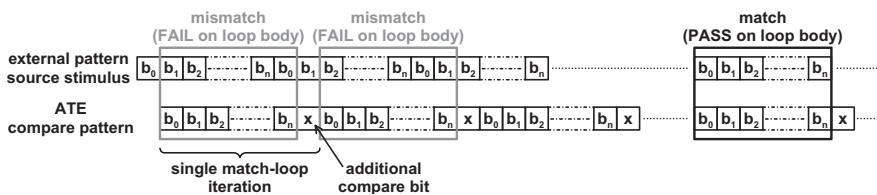


Figure 5.4 Sliding compare window implemented by a match loop.

5.1.2.2 Timing Search

Another way to achieve pattern alignment, of course, is to do a timing search and to compensate the measured pattern offset by a timing adjustment on the ATE. Since bit alignment was done beforehand and compare strobes are positioned in the center of the data eyes, the search steps are set to one UI.

The problem with the timing search, however, in many cases is that the pattern offset can extend beyond the boundaries of the timing programming ranges of the ATE. If a timing search is possible (e.g., because the pattern repeated by DUT is shorter than the allowed timing programming range), one has to take care that the relative pattern offset between DUT and ATE stays constant for the repetitive sequencer starts that will be caused by this timing search. On ATE systems that support cycle times that are prime to each other and have synchronization mechanisms available to synchronize the different clock domains that are set up, a constant offset can be achieved by setting up a dummy clock domain.

This dummy clock domain runs at a cycle time that corresponds to the length of the pattern that the DUT loops on. The ATE synchronization mechanisms then take care that the sequencer(s) of the ATE channels that do the timing search always are restarted on a timing grid that has a spacing of the least common multiple of the two cycle times of the clock domains as shown in Figure 5.5. Thus, the sequencers of the searching pins always are restarted with the same relative pattern offset to the pattern generated by the DUT.

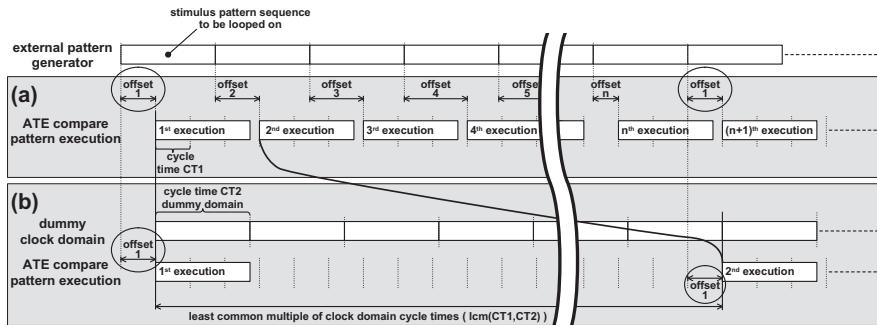


Figure 5.5 Relative ATE sequencer restart offset to external pattern (a) without and (b) with usage of clock domain synchronization.

Sequencer Offset

A final possibility for pattern alignment that even works with offsets bigger than the allowed timing programming range is to delay the generation of the compare pattern data on the ATE by inserting no operation (NOP) or dummy sequencer operations between the start of the sequencer(s) and the sequencer operations that generate the actual compare data. In order to calculate the necessary amount of NOP operations that need to be inserted, a portion of the received data is captured and its position within the expected data is analyzed.

The difference between the position of the captured data in the expected data and the position it actually was captured at represents the pattern offset that needs to be compensated by NOP operations to achieve pattern alignment. Of course, it is required that the captured data is a unique bit sequence within the expected data to be able to identify the pattern offset between the received and expected data. For PRBS patterns, this is achieved by capturing the number of bits that is used by the linear feedback shift register (LFSR) that generates the PRBS sequence. For other patterns the number of bits captured needs to be set in a way that guarantees that a unique bit sequence is captured in any case.

5.2 Functional Test

At-speed functional tests are not only at the core of ATE based testing in general, but also are an important factor for testing high-speed interfaces. Some of the measurement algorithms that are used to characterize important high-speed parameters are based on functional tests. During a functional test, stimulus data is applied to the DUT by the ATE drivers, external instruments,

or a combination thereof. The ATE receivers compare the data generated by the DUT as a reaction on the stimulating data to a data pattern defined inside the ATE system.

The pattern applied and compared by the ATE at a single timing instance also is called pattern vector or simply test vector. The minimum result a functional test delivers is whether the data pattern received from the DUT matches the ATE compare data or not (pattern pass or fail). The way the data received from the DUT is interpreted with regard to the nomenclature used for the compare pattern is determined by the compare threshold voltage(s) used by the receiver pin electronics front end. For high-speed pin electronics it is quite common that only a single threshold voltage exists, and thus, the data received can be distinguished between high and low only (no intermediate state is usually available for lower speed dual threshold comparators).

The generation of test vectors for functional tests can happen in various ways. Typically, test vectors are derived from logical device simulations or other means that result in predefined static stimulus and compare data stored in the vector memory of the ATE. In the memory test area, huge amounts of vectors are required to do exhaustive functional testing. Since it would not be affordable to keep these test vectors statically stored in a vector memory, memory test systems usually generate their test vectors dynamically with so-called algorithmic pattern generators (APG) that allow a high level definition of the test patterns using a programming language style.

The actual test patterns are generated from this description either directly on-the-fly using dedicated APG hardware on some ATE systems or during a compilation process that generates test patterns optimized for the sequencing capabilities on other ATE systems. Recently, similar approaches also found their way into nonmemory high-speed ATE systems and allow a mixture of the classical statically predefined vector pattern storage in vector memory and the usage of patterns that are generated algorithmically on-the-fly. One example for such algorithmically generated patterns are pseudo-random bit sequences (see Appendix C).

During a single functional test, there is only limited user control on the applied timing and level settings. Once the functional test is started, the predefined settings for these parameters are used. These may only be modified at predefined positions within the pattern to predefined values but not in a dynamic manner. Although some ATE systems offer the capability to dynamically adjust these parameters within their data pattern definitions depending on the test result of the previously executed pattern vectors, the execution of such a pattern is not considered a single functional test. The reason for this interpretation is the fact that the evaluation of the pass/fail result and the parameter adjustment depending on this result interrupt the at-speed

pattern execution and thus introduce a pause in the at-speed pattern.

Besides just a single overall pass/fail result for the complete functional test, ATE systems have various other capabilities to deliver more information about the results of a functional test execution. Another level of information, for example, is the exact number of errors observed during a functional test or even the exact positions of these errors within the test pattern. All of this information might be available also on a per-pin basis depending on the ATE system. In general, the more information on a functional test execution that is made available, the better a functional test can serve as a starting point for parametric measurements that use this information in post-processing steps like, for example, the measurement of a BER (see Section 2.3) which necessarily requires the exact error count information of a functional test.

Another aspect of functional testing that gained relevance with the increasing amount of packet-oriented, encoded high-speed interface standards is the topic of protocol-aware testing. In some sense this also can be interpreted as algorithmic pattern based functional testing because the test engineer does not work on a physical layer presentation for the vector data anymore but on a layer that shows the payload data transferred from and to the DUT. The algorithmic part of the pattern generation in this case is the transfer of the payload data into actual physical data and vice versa. We will discuss protocol-aware testing in more detail in Section 9.6.

5.3 Shmoo Tests

Shmoo tests play a central role in the characterization and production ramp phase of a device. With the shmoo test methodology, test and device parameter margins can be identified and a test engineer can draw conclusions on the actual performance of a DUT and on the test repeatability and stability for the selected test conditions.

For a shmoo test, one or more of the test parameters controlled by the ATE (e.g., timing or level parameters) are swept over a range defined by the test engineer. The resolution of the single sweep steps also is defined as a shmoo test parameter by the test engineer. For each of the sweep steps, a functional test is executed and the result of the test is recorded. As already described in Section 5.2, this result might not just be the pass or fail information of the functional test but also the number of errors seen for the sweep point or any other result supported by the ATE. With this, a shmoo test allows identification of the test margin a certain parameter has compared to the actual device performance.

While shmoo tests that sweep a single test parameter already deliver important insights into the DUT operation boundaries, the real power of

shmoo tests is unleashed if multidimensional shmoos are applied to a DUT. For such a multidimensional shmoo, sweep settings are applied to several test parameters. During the shmoo test, a functional test is applied for all combinations of parameter settings that are defined by the single sweeps. The tests for all shmoo parameter combinations easily allow identification of parameter dependencies and parameter settings for a safe device operation. However, it has to be taken into account that the overall number of functional tests that are executed for such a multidimensional shmoo grows exponentially with the number of shmoo dimensions.

Thus, test engineers typically limit the number of dimensions in real test implementations to two or three to achieve reasonable test times. There might be cases, however, where it makes sense to tolerate long test times and use shmoo tests with more than three dimensions. This mainly is true for characterization measurements that have the target to gather data that allows distinguishing critical parameters that need to be considered during production testing from noncritical test parameters that might not even be checked in a production test flow.

Besides reasonable test times, shmoo tests with three or less dimensions also have the advantage that they can be displayed relatively easily in a graphical manner. Graphical representations of shmoo test results often are referred to as shmoo-plots.

Shmoo-plots usually simplify the interpretation of the shmoo test results for a test engineer. In order to speed up the execution of a multidimensional shmoo, shmoo algorithms like fast-shmoo are applied [3]. In a fast-shmoo, an initial parameter sweep is executed that does not use the minimum sweep resolution defined by the test engineer but a lower resolution to scan the complete sweep area. After this initial measurement, only the areas that are located between parameter settings that caused a change in the test result are scanned with the desired resolution. A typical example of such a fast-shmoo is shown in Figure 5.6 for a data eye measurement.

When setting up shmoo tests, one has to specifically take care of the fact that sweep parameters selected for one of the shmoo dimensions have to be independent of other test parameters. If this is not the case (e.g., a shmoo that sweeps the supply voltage typically affects DUT input voltages and compare thresholds), the test engineer has to use a shmoo setup that tracks these dependent parameters together with the parameter that is swept on one of the shmoo dimensions. Such a shmoo setup is often referred to as tracking-shmoo. The parameter tracking usually is done via a user defined dependency equation that describes how the tracked parameter has to behave in dependency of the parameter used in the shmoo setup. One example for such a tracked dependency, for example, would be the ATE drive and comparator threshold

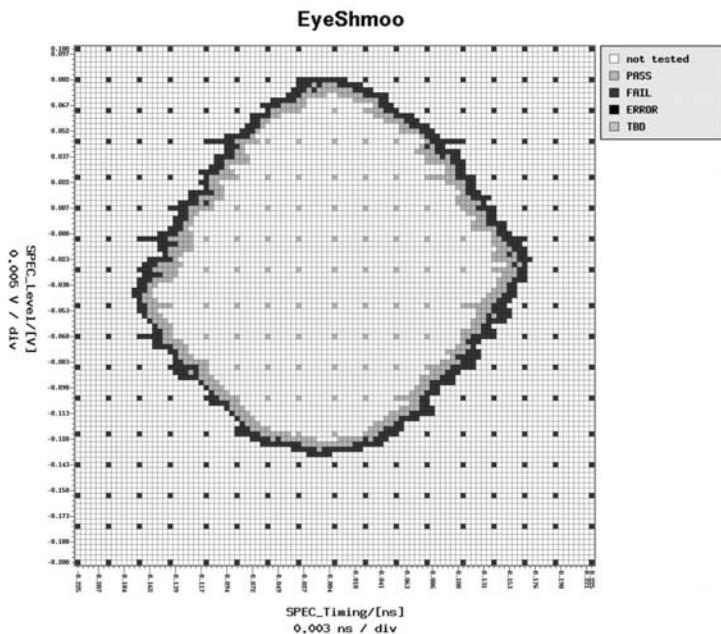


Figure 5.6 Fast-shmoo of device data eye.

levels used during a shmoo test that sweeps the supply voltage of the DUT.

Another boundary condition that has to be taken into account is the potential dependency of the overall DUT functionality on one or more of the parameters used in the shmoo dimensions. This is especially true for high-speed interfaces that, for example, might require retraining after the change of parameters that have influence on the PLL phase locking of the high-speed I/O signals (e.g., reference clock frequency or supply voltage). If such a critical parameter is swept during a shmoo, the test engineer has to take care that the appropriate steps, for example, device retrainings, are performed at the correct instances during the shmoo execution.

A few examples for typical two-dimensional shmoo setups that are widely used for high-speed I/O devices are frequency shmoos that sweep the device operating frequency versus a data eye width measurement, data-eye shmoos that sweep compare strobe timing versus compare level threshold, and VDD shmoos that show the dependency of the maximum operating speed of a device on the supply voltage level.

5.4 Fundamental Driver Tests

This section presents some of the typical measurements for testing an I/O cell driver and how these tests can be implemented on an ATE system.

5.4.1 Rise/Fall Time

The rise and fall time measurement delivers the amount of time the DUT driver takes to achieve a logic one voltage level starting at logic zero (for rise time) or starting at the logic one voltage level to achieve logic zero (for fall time). Typically rise or fall time is not measured from one voltage rail to another but from a given percentage from the rails voltage level, normally 10–90% or 20–80% as shown in Figure 5.7.

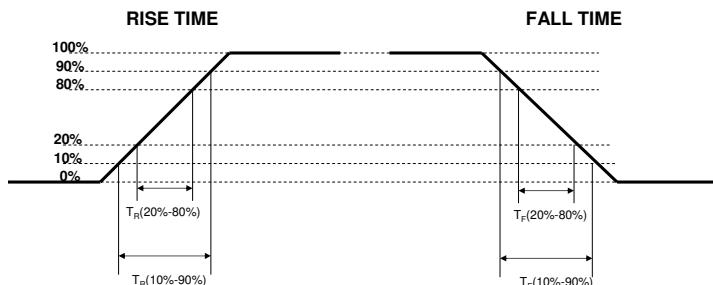


Figure 5.7 Rise and fall time measurement definitions (10%-90%) and (20%-80%).

The rise and fall time measurement results will depend significantly on the signal path loss of the test fixture and the measurement instrument bandwidth. The reason is that the test fixture bandwidth and the measurement instrument bandwidth will behave like a lowpass filter on the signal rise time, slowing it down. One way to address this issue is to use (5.1) to remove the effects of test fixture and instrument bandwidth. The formula requires the intrinsic rise time of the test fixture signal path and receiver bandwidth to be known so that it can be subtracted to the measured rise time to obtain the rise time at the DUT output. Note that this formula is only exact in special cases like Gaussian filters. In cases where the degradation is due to other causes like the skin effect in a coaxial cable, the formula is no longer exact [4].

$$Tr_{\text{MEASURED}} = \sqrt{(Tr_{\text{DUT}})^2 - (Tr_{\text{SIGNAL PATH AND RECEIVER}})^2} \quad (5.1)$$

Another important topic when measuring rise/fall times is the effect of jitter on the measurement. Jitter “spreads” the signal edge creating the

problem of which points should be used for the time instants on the rise/fall time computation. The typical approach to address this challenge is to perform a histogram measurement on the two voltage levels intended to be used for measuring the rise time and then select the average value for the measurement as shown in Figure 5.8. This is equivalent to performing a waveform averaging in a real-time oscilloscope. This approach works very well in cases where the signal jitter is dominated by random jitter [5].

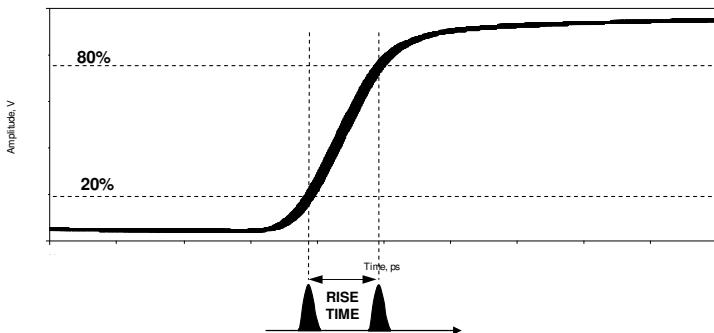


Figure 5.8 Using a histogram approach in a rise/fall time measurement.

5.4.2 Data Eye Diagram

The objective of the data eye diagram is to measure the degradation of a waveform from the DUT in comparison to the ideal one. In the ideal case, each bit in the data stream should have the time length at the optimal threshold voltage point equivalent to the bit rate period T_{BIT} (e.g., 1 ns for a 1-Gbps NRZ signal) and the signal amplitude between the high and low levels should be equivalent to the expected ideal amplitude. Unfortunately a real signal will have jitter, amplitude noise, and limited rise and fall time as shown in Figure 5.9.

As already discussed in Section 2.1.2, a data eye diagram provides a visual representation of the parametric performance of the DUT driver. From a data eye diagram several parametric performance values can be obtained like the rise/fall time, data eye width, data eye height, jitter, and the amplitude noise.

5.4.2.1 Measuring the Data Eye Diagram with a Digital Pin Electronics Receiver

To measure the data eye diagram with a digital pin electronics receiver, it is necessary to perform a shmoo test across the compare strobe timing and the

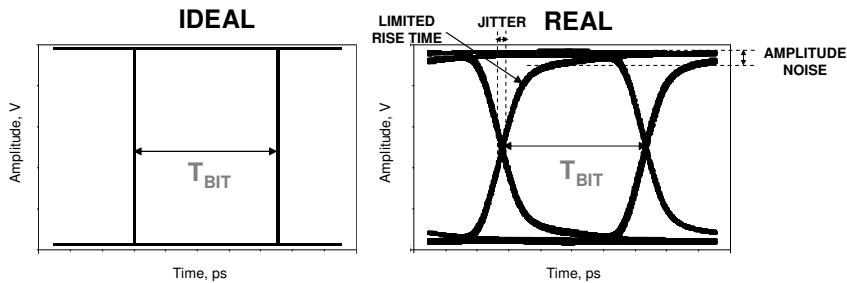


Figure 5.9 Ideal versus a real data eye.

voltage compare threshold. For this purpose the horizontal axis representing the strobe timing is divided into, for example, 100 steps. This is repeated for the vertical axis showing the compare voltage. For each combination of these steps ($100 \times 100 = 10,000$), a functional test is executed by the digital pin electronics receiver.

The result of the functional test can either be a pass/fail or the number of failed bits if the pin electronics has the ability to count the number of failed bits. Figure 5.10 shows one example of a shmoo plot that corresponds to a data eye diagram measured with an ATE system. The data eye diagram in this case is plotted with different colors representing the different number of failed bits on the functional test. The test time will depend on the resolution used for the timing and voltage steps and the number of compared bits. The measured accuracy will depend on the digital pin electronics receiver bandwidth, the number of compared bits, the jitter and amplitude noise on the receiver compare strobe, and also on the compare strobe linearity associated with its movement along the time and voltage axis on the data eye diagram shmooplot.

One of the major advantages of an ATE system is the ability to measure multiple I/O channels in parallel. This allows, for example, the overlay of multiple data eye diagrams corresponding to different pins in a I/O bus as shown in Figure 5.11. The 100% values correspond to a combination of strobe timing and compare voltages where the functional test for all pins in the interface bus results in a pass. Percentages below 100% indicate timing and voltage settings where at least one of the pins in the interface bus had a failure during the functional test. This type of overlay shmooplot provides an overview of the performance difference between the different pins that compose the interface bus including any timing skew.

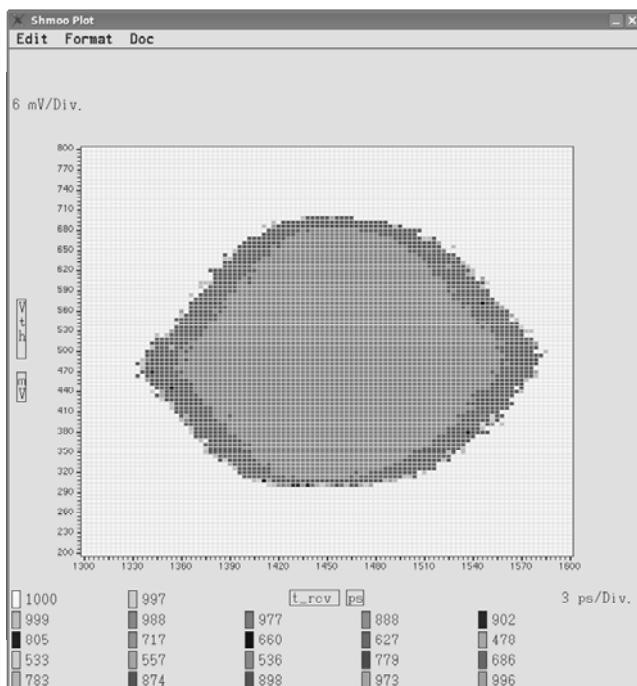


Figure 5.10 Example of measuring a data eye diagram with an ATE digital pin electronics receiver (courtesy of Verigry).

5.4.2.2 Measuring the Data Eye Diagram with a Sampler¹

A sampler can also be used to generate a data eye diagram by applying the techniques described in Section 4.3. Figure 5.12 presents an example of a test configuration for performing a data eye diagram measurement with a sampler where the DUT drives a $2^7 - 1$ PRBS sequence at 2.5 Gbps.

In this example, the sampler is programmed to capture two sets of the entire PRBS pattern (i.e., a total 254 bits with $N = 65,536$ points of data). The sampler runs at the rate of 9.8423695 Msps by using the coherency configuration shown in Figure 5.12.

Figure 5.13(a) shows the captured waveform containing two sets of the PRBS sequence. By reshuffling the waveform with the key number 127 instead of 254, the data eye pattern can be reconstructed as shown in Figure 5.13(b) [6].

¹Contributed by Hideo Okawara.

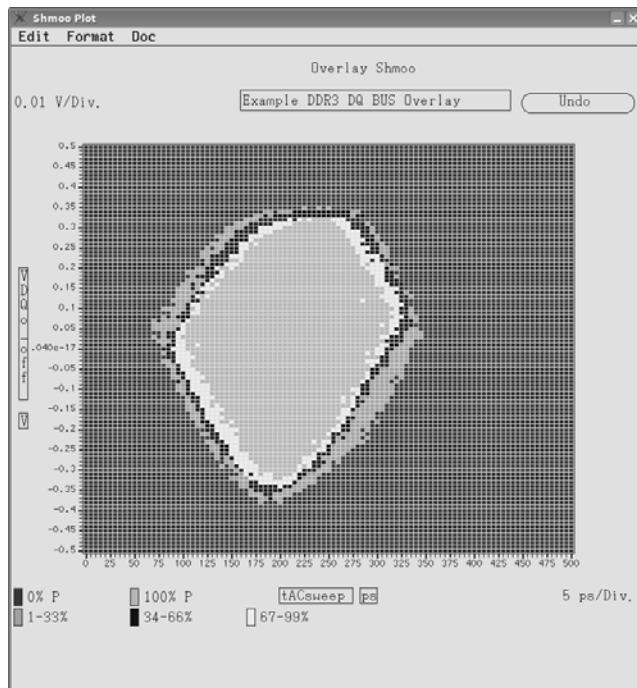


Figure 5.11 Overlay of multiple data eye diagrams from a DDR3 bus into a single data eye diagram (courtesy of Verigty).

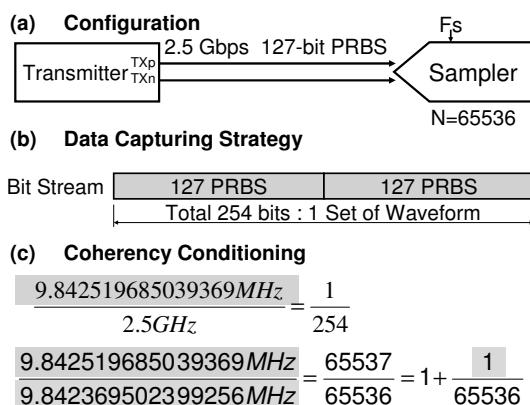


Figure 5.12 (a–c) Data eye measurement with a sampler configuration.

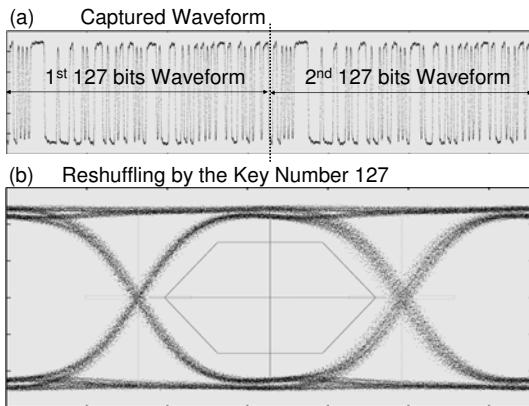


Figure 5.13 (a, b) Measured waveform and reconstructed data eye diagram.

5.4.2.3 Data Eye Width

The objective of the data eye width test is to measure the percentage of the ideal bit period where the setting of the strobe timing results in a functional pass (i.e., how much time margin is available for a device receiver connected to the DUT transmitter). This can be a complex task depending on the link implementation (e.g., some receiver designs have their own CDR circuit that is able to remove some of the timing jitter on the data). The question whether this circuit should be taken into account when measuring the eye width depends on the test requirements (e.g., some standards require a specific CDR to be used on all driver parametric measurements). This topic is addressed in Section 9.5 in more detail. In the remaining part of this chapter we assume no CDR circuit is used on the measurements.

Another issue is the fact that the peak-to-peak value of the random jitter increases over time (i.e., if the observation period is long enough, the data eye width will theoretically decrease to zero). This means that the measured data eye width depends on the number of acquired samples. This will be discussed more in detail in Section 5.4.3 with the BER bathtub curve.

Assuming that a data eye diagram is measured using a given amount of samples or bit compares, the data eye width is measured by computing the distance between the left and right edges of the data eye diagram at the optimum voltage threshold point as shown in Figure 5.14. In the figure the average values of the timing histograms at the optimal threshold point provide the data eye period but the data eye width will be the distance between the edges of the left and right histograms. Figure 5.15 shows an example of how the eye width is represented on an oscilloscope and the data eye shmoo on an ATE system.

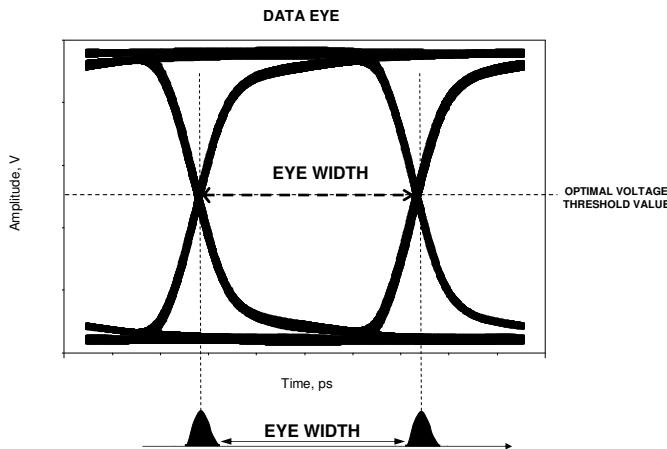


Figure 5.14 Measuring the data eye width.

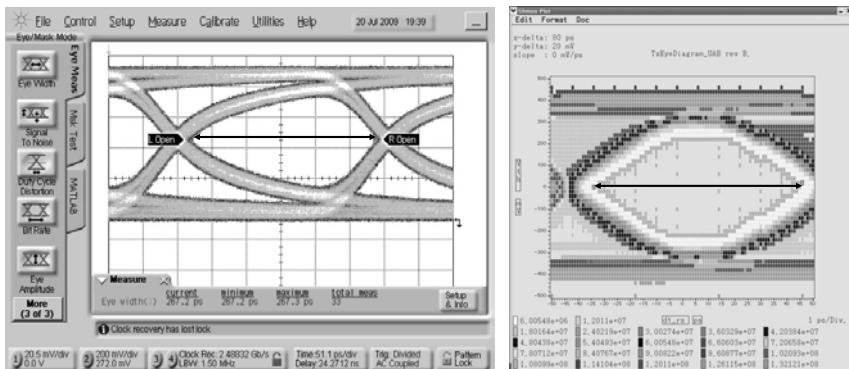


Figure 5.15 Examples of measuring the data eye width (left: Agilent DCA; right: shmoo with Verigy 93000 ATE).

5.4.2.4 Data Eye Height

The objective of the data eye height test is to measure the level margin available on the data eye from the DUT driver. This is important because the receiver on the link partner will receive a data eye that is degraded by the transmission channel. Therefore it is critical to not only achieve enough timing margin but also ensure that the amplitude margin on the DUT driver data eye is sufficient for the link partner receiver to correctly identify the transmitted bits. The reasons why the data eye height does not correspond to the expected DC amplitude levels are the random noise inherent to the voltage levels and the bandwidth limitation on the DUT driver and the test fixture for a given

transmitted bit, which depends on the preceding bits as shown in Figure 5.16.

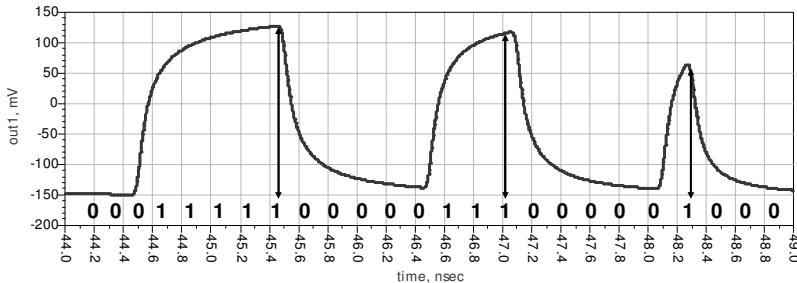


Figure 5.16 Example of a digital waveform after a lossy channel showing how the maximum amplitude reached by each bit depends on the previous bits.

When looking at a data eye diagram as a representation of a waveform, one can observe that the amplitudes of the various data bits on the data pattern will be superimposed. A data height measurement reveals the available voltage margin for the worst case bit in a given data pattern. This can be accomplished by computing the difference between the top and bottom inner edges of the data eye diagram. There are several possible approaches as shown in Figure 5.17.

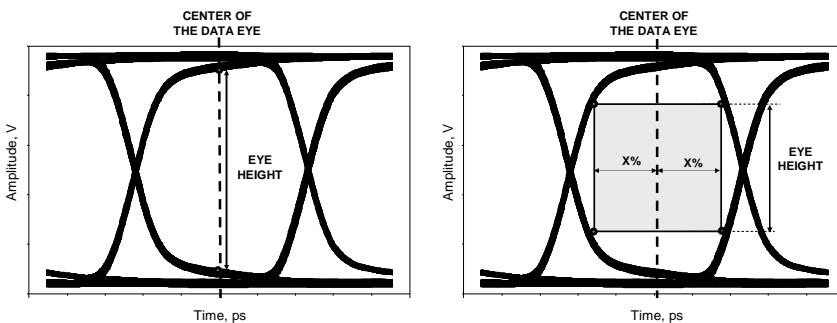


Figure 5.17 Possible approaches for measuring the data eye height.

One approach is to measure the voltage interval between the top and bottom edges of the pass/fail region taken at the center (optimal timing strobing point) of the data eye as shown in Figure 5.17 (left). Figure 5.18 (left) shows an example of this approach with an equivalent-time oscilloscope, and 5.18 (right) shows the same approach with an ATE based data eye diagram.

Another option is to define a rectangular region around the center of the data eye where the top and bottom level rails must not cross as shown in Figure 5.17 (right). This approach takes into account that the receiver might not be

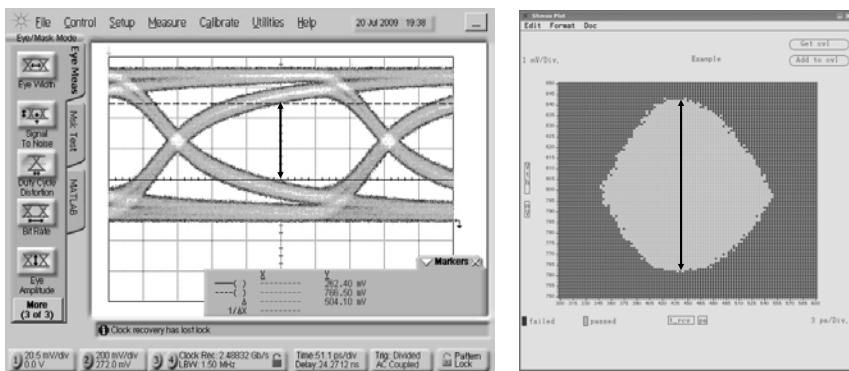


Figure 5.18 Examples of measuring the data eye height (left: Agilent DCA; right: shmoo with Verigy 93000 ATE).

ideal and that the strobing point might not be exactly at the center of the data eye. Figure 5.19 shows an example of this approach using an equivalent-time oscilloscope.

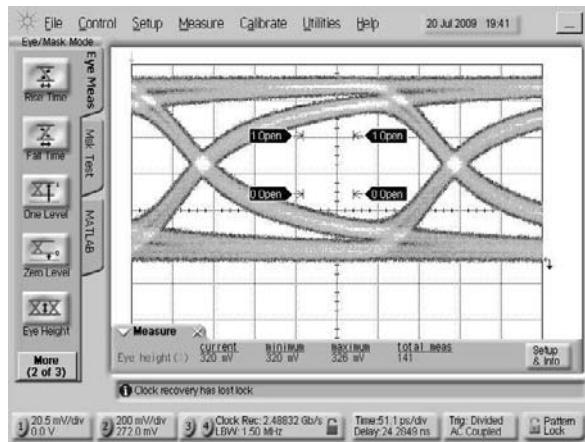


Figure 5.19 Example of measuring the data eye height with an equivalent-time oscilloscope using a pass region that is 20% of the UI around the data eye center.

5.4.2.5 Data Eye Mask Test

The data eye mask test is an approach to measure with a short test time if the data eye has enough performance for a given application or standard. Instead of measuring the data eye parameters like the data eye height or width

separately, a data eye mask is verified. Such a data eye mask consists of one or multiple geometric polygons (e.g., rectangles and hexagons) with a specific geometric configuration that is used to specify the minimum performance requirements on the DUT data eye. The data eye diagram must not “touch” any of the data eye mask polygons as shown in Figure 5.20. If a DUT I/O driver passes a data eye mask test defined in a standard, it means that the driver has the necessary level and timing margin to be compliant with the standard requirements.

With most measurement instruments (e.g., an equivalent-time sampler or an ATE sampler), the violations of the data eye mask are computed after the data eye diagram has been acquired. Applying digital pin electronics, the test time can be shortened by simply executing a functional test at the points specified by the data eye mask instead of acquiring the complete data eye through a shmoo-plot. In Figure 5.20(b), for example, it is sufficient to check if any of the points for the data eye mask test polygon inside the data eye provide a fail during a functional test. For the polygon above the data eye also a simple functional test is performed which is exclusively compared to logical “high” and checks for errors. The test time of a data eye mask test can be further reduced by performing a fast eye mask test, which will be explained in Section 6.3.2.1.

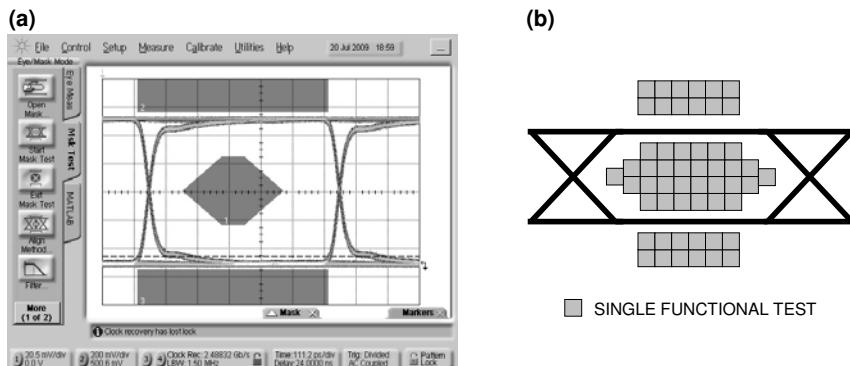


Figure 5.20 Examples of performing a data eye mask test. (a) Using an equivalent-time oscilloscope. (b) Using a digital pin electronics receiver.

5.4.3 BER Bathtub Curve

One fundamental measure of the performance of a DUT driver is the BER bathtub curve² [7]. The concept of BER was already defined in Section 2.3. The BER bathtub curve measures the BER of the DUT driver by an ideal receiver at the different strobing instants across the data eye.

One way to think about it is to imagine that we have an ideal receiver (no intrinsic noise) where the strobing position of the receiver can be changed from its optimal position in the center of the data eye. If the DUT driver were also perfect without any intrinsic noise, then the “perfect” receiver should be able to set its strobe timing at any point of the bit period obtaining a zero BER at each point. In this case we would obtain the graph shown in Figure 5.21 where in every point inside of the data eye the BER is zero.

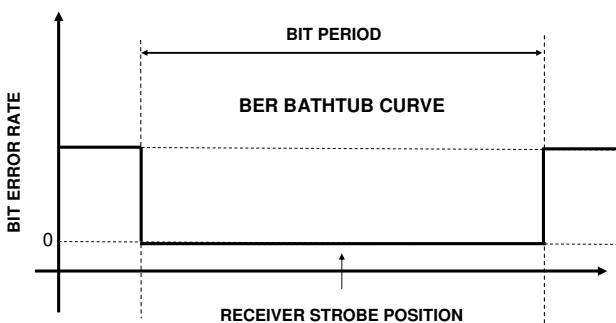


Figure 5.21 The BER bathtub curve assuming a “perfect” DUT driver.

Real DUT drivers have inherent noise; therefore, even a perfect receiver will not be able to correctly strobe the incoming data pattern for timing strobes at the edges of the data eye bit period. Figure 5.22 illustrates this fact by showing that the BER no longer goes abruptly to zero at the edges of the data eye. The most straightforward way to obtain a BER bathtub curve is to use a measurement instrument as the “ideal receiver” and then do a functional test with the receiver strobe covering the entire bit period in a given number of steps. In an ATE environment, a digital pin electronics receiver with the ability to record the number of errors in an at-speed functional test would suffice. It is important to note that the performance of the receiver in terms of bandwidth and jitter has a direct impact on the measurement accuracy since an ideal BER bathtub curve measurement would require an ideal receiver. Another important point is the linearity of the receiver strobing edge (see Section 9.1.3).

²Sometimes the measurement of the BER bathtub curve is referred to as a BERT scan.

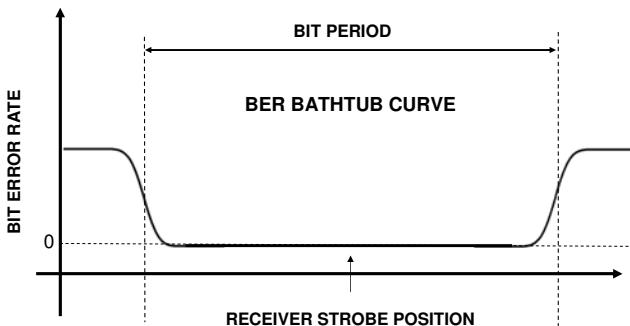


Figure 5.22 The BER bathtub curve for a realistic (not perfect) DUT driver.

The BER bathtub curve is usually plotted using a logarithmic scale for the BER instead of a linear scale as shown in Figure 5.23 to allow a more easy visualization of points with a very low BER value. The test time to obtain a BER bathtub curve measurement is given by (5.2).

$$\text{Test Time} = \text{Number of Compared Bits} \times \text{Bit Period} \times \frac{\text{Bit Period}}{\text{Step Resolution}} \quad (5.2)$$

For example, for a 10-Gbps pattern where 10^{12} bits are compared in each functional test and a resolution of 5 ps is used, the test time of just running the pattern for the complete BER bathtub curve measurement would be around 33 minutes.

Given the large time that it can take to acquire a BER bathtub curve, the test engineer is forced in some situations to reduce the test time by changing the two variables under his control that have a direct impact on the test time: the step resolution and the number of compared bits. It is important to understand that this change might have an impact on the measurement results depending on the jitter characteristics of the DUT. Figure 5.23 shows two BER bathtub curves from the same DUT where a different number of compared bits is used.

The figures show that by using a reduced number of points, the smallest BER value that is measured will be different. This is expected since to measure a BER of 10^{-10} we would need to measure at least 10^{10} bits and assume that one of those bits would be a bit error. A more realistic approach would be to make the compare bit pattern at least 10 times larger than the minimum BER value one intends to measure (Section 5.10 presents some more discussion on this topic).

The main objective of measuring a BER bathtub curve is to find out how much timing margin the DUT driver has. This is similar to the objective of the

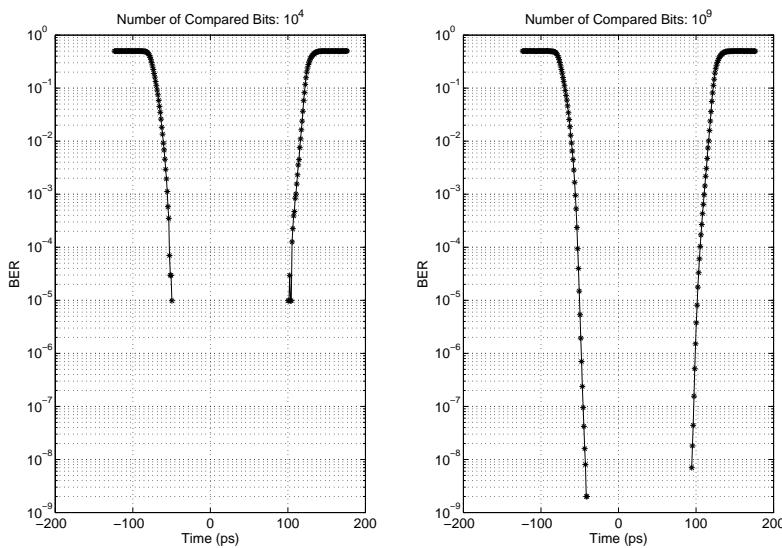


Figure 5.23 Influence of the number of compared bits on the BER bathtub curve (left: 10^4 compared bits; right: 10^9 compared bits).

data eye width measurement but the BER bathtub curve provides a more deep understanding of the effect of the random jitter through the compare pattern size on the data eye width. The reason is that the BER bathtub curve provides the timing margin for each possible BER value. For example, if the I/O design target is to have a maximum of one error per 10^8 transmitted bits, then the critical eye width value is the one obtained for a BER of 10^{-8} as shown in Figure 5.24.

Associated to the data eye width is the value of total jitter. Basically the total jitter at a certain BER threshold is equivalent to the bit period minus the eye width.

$$\text{Total Jitter (BER)} = UI - \text{Eye Width} \quad (5.3)$$

In most cases the design target is defined in terms of the amount of total jitter at a given BER threshold. The challenge is that most standards define the total jitter value for a BER of 10^{-12} which would imply a very large test time due to the required pattern size (e.g., 10^{13}). Section 5.5.5 discusses a different and faster approach to estimate the total jitter at a certain BER through random and deterministic jitter separation. In [8] an approach is proposed for a faster measurement of the total jitter through a BER measurement but avoiding the long test time of a standard BER bathtub curve.

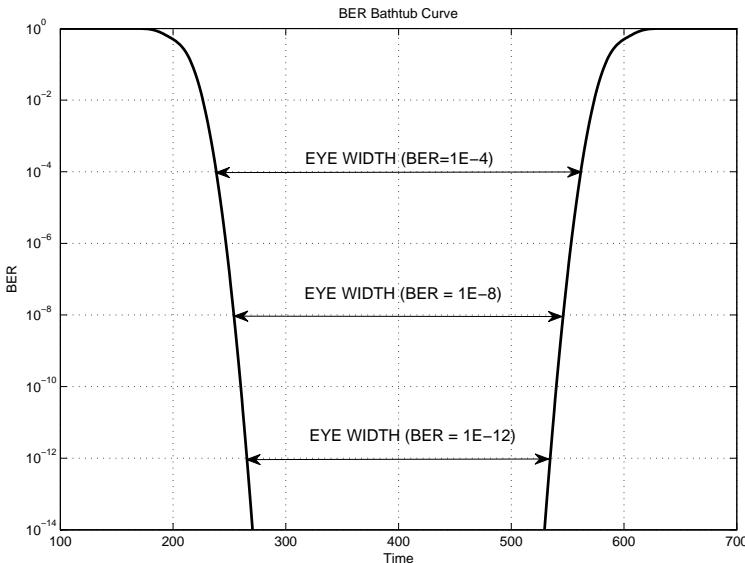


Figure 5.24 Measuring the data eye width value at different BER thresholds.

5.4.4 Skew

For at-cycle source-synchronous interfaces, transmitter or driver skew together with the receiver setup/hold time (described in Section 5.6.1) is one of the most critical test and characterization parameters. The skew between the transmitted clock and its associated data signals determines the margins a receiver will have best case in regard to the required setup/hold times to correctly recognize the transmitted signal. The skew within the data signals determines the best case available valid data eye and thus the overall available margin for the sum of setup and hold time that is left for a receiver as shown in Figure 5.25.

It has to be noted that skew and level loss of the device interconnections will further reduce the margins an at-cycle source synchronous receiver will have available to recognize the transmitted data correctly. Thus, taking into account the overall timing budget of such a connection, the specified skew values for the transmit interface of a device are relatively small compared to the nominal data eye width at the specified data rate. This increases the sensitivity of data connections on even small deviations from the tight margins for the driver skew specifications and makes skew a must-test parameter.

Therefore, the skew between ATE channels becomes a critical factor for the accuracy of this measurement because the skew values to be guaranteed are in the range or even below the edge placement accuracies typical ATE

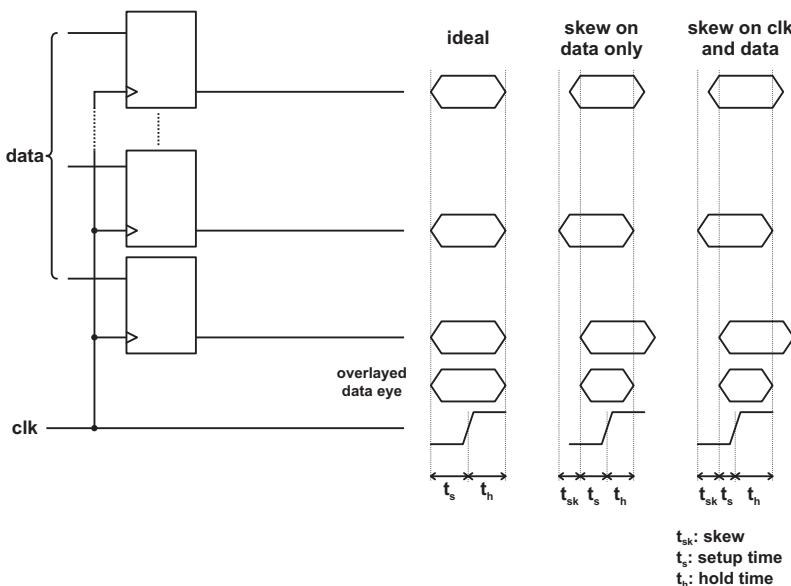


Figure 5.25 Transmitter or driver skew.

systems provide with their standard calibration procedure. As a consequence, focus calibration techniques as discussed in Section 9.3.1 might need to be applied to enable an ATE to test transmitter skew for at-cycle source-synchronous interfaces with sufficient accuracy.

For I/O interfaces that deploy device training mechanisms to compensate for static skew on the device interconnections (see Section 2.5), skew specifications usually are less tight and skew measurements are less important because the training ensures maximum margins for the receiver setup/hold regardless of potential transmitter and connection skew.

Characterization measurements for transmitter skew usually identify the time for corresponding signal transitions on two or more pins and calculate the skew as the difference between the times at which the transitions were found for the involved signals. Of course, an additional offset might need to be added to the measured skew if the skew specification is not related to signal transitions on the involved signals but, for example, the skew between a clock edge and the center of an associated data bit.

The instrumentation that can be used for skew measurements ranges from dedicated external support instrumentation like time interval analyzers (TIA; see Section 7.3) over digital pin electronics that have integrated TIA or time stamper functionality (Section 4.1.3) to purely digital pin electronics without this special functionality.

With the TIA or time stamper approach, it is important that the measurement resources to be used support relative timing measurements between at least two channels to be able to perform skew measurements. Another side effect that needs to be considered when TIA or time stamper circuitry is applied, is that these circuits usually introduce a latency (arming-latency) between activating the measurement circuitry and performing the actual measurement. Typically the arming-latency is specified with an uncertainty (variation) that is significantly larger than the skew to be measured for high-speed I/O interfaces. This can lead to the effect that corresponding items of the timing measurement tuples³ for the signals to be analyzed are not referring to corresponding edges. A test engineer has to consider that such a case can be identified and the single measurement components are reshuffled in a way to get alignment of the measurement data along corresponding signal edges.

In case of digital pin electronics without TIA or time stamper functionality for skew measurements, the timing measurements to identify the signal transition locations are implemented by means of timing searches. The same search approaches as discussed in the bit alignment part of Section 5.1 are used to identify the timing of the signal transitions on the pins involved in the skew measurement. An important boundary condition a test engineer has to take into account when doing skew measurements is the fact that corresponding transitions between the signals have to be measured. If this requirement is not considered, singular effects that affect the transition timing at the measurement locations in the test pattern differently (e.g., ground bounce or crosstalk) might falsify the measurement. For a timing search that is based on functional tests, this means that not all bits of the pattern should be compared because the search in this case would identify the signal transition location as the overlay of all compared transitions in the pattern. Such an overlayed result per pin does not allow the extraction of the skew for associated signal transitions on the signals for which the skew should be measured. Thus, the pattern used for skew measurements has to only activate compare actions for either a single bit before or after the transition of interest or for the two bits separated by the transition of interest. With this, the associated transitions that are of interest for the skew measurement are marked and are the only ones that influence the search results.

As we have seen in Section 2.4, jitter is another factor that influences the timing of signal transitions and thus also has influence on the results of skew measurements. In order to minimize this influence, a test engineer should avoid using pattern data that introduces additional jitter (e.g., due to

³A set of timing values with one single timing value per measured signal.

bandwidth restrictions on the signal path connecting the ATE pin electronics with the DUT).

5.4.5 Pre-Emphasis and De-Emphasis Measurement

Modern I/O cells incorporate special techniques like pre-emphasis and de-emphasis to address the loss from PCB signal traces at high data rates as discussed in Section 2.6.4. Typically the objective is to measure the level of pre-emphasis or de-emphasis on the output waveform as shown in Figure 5.26 for a de-emphasis application.

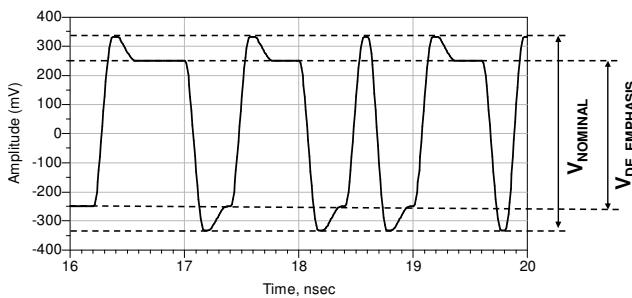


Figure 5.26 Measuring de-emphasis example.

The most straightforward option is to acquire the waveform with the digital pin electronics or with the sampler and measure the pre-emphasis/de-emphasis value on the acquired waveform though post-processing. Although this option is the most generic way to measure any type of pre-emphasis/de-emphasis implementation (there are several techniques with different complexity levels), it consumes a significant amount of test time especially when using digital pin electronics.

A simpler approach with digital pin electronics is to measure the pre-emphasis/de-emphasis value through a two-step approach. First, a level shmoo-test at the center of the data eye is performed only for the bits on the data pattern that are pre-emphasized or de-emphasized to obtain the eye height for the first bit after a transition. In a second step a level shmoo test is performed to obtain the eye height for the rest of the bits. The pre-emphasis/de-emphasis value will be the difference between these two shmoo values. The problem with this approach is that it can only address certain types of pre-emphasis/de-emphasis algorithms, for example, algorithms where only the first in a sequence of equal logic value bits is pre-emphasized.

5.5 Driver Jitter Tests

This section discusses the different types of jitter tests that can be performed on the driver of a DUT I/O cell.

5.5.1 Jitter Histogram

The jitter histogram is the most basic way of analyzing the jitter of a digital signal. The idea is to make a measurement of the timing of each edge of the signal compared to a reference signal (e.g., a low jitter clock source for measuring the time interval error (TIE) jitter) or even in some cases to the previous edge (e.g., when measuring cycle-to-cycle jitter) and generate a histogram of the measurements. The exact way the histogram is generated depends on the specifics of the measurement methodology and measurement instrument. For example, if the result of the measurement is an array of values for the measured jitter at each transition (e.g., using a time interval analyzer as the measurement instrument), then from this array a histogram can be generated as already discussed in Section 2.4.1.

In the case of a digital pin electronics receiver, the histogram can be obtained by the derivative of the shmoo-plot of a functional test error count across the data eye edge transition area as shown in Figure 5.27. Basically one side of the BER bathtub curve can provide the jitter histogram for that transition by using the derivative on the BER bathtub curve.

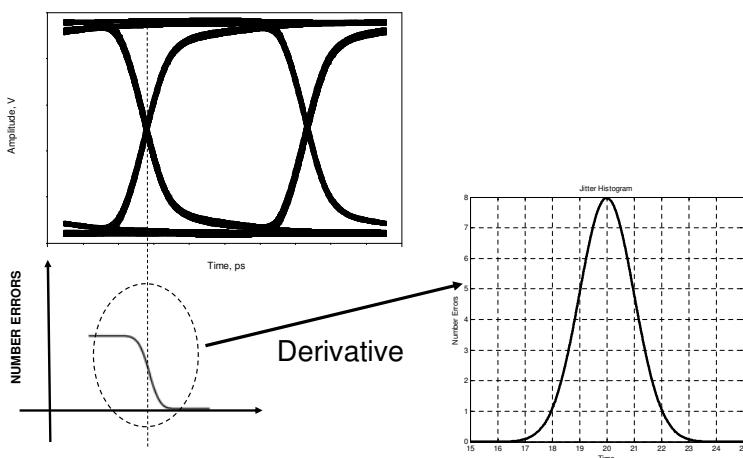


Figure 5.27 Obtaining the jitter histogram from an error count shmoo.

There are two typical values when measuring jitter that are associated with the histogram. They are the jitter RMS value and the jitter peak-to-peak value.

5.5.2 RMS Jitter

The jitter root mean square (RMS) value corresponds to the standard deviation of the jitter measurement (root mean square is typically used as a synonym for standard deviation in physical sciences). If one has access to the array of measured values the best estimate of the standard deviation (RMS jitter) can be computed in the following way:

$$\bar{\sigma} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (5.4)$$

with the average \bar{x} defined as

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.5)$$

In some cases the sample values might not be available and only the histogram is available. In this case the estimate for the standard deviation can be computed by the following formula:

$$\bar{\sigma} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 H_p(x)} \quad (5.6)$$

where $H_p(x)$ is the value of the normalized histogram for the value x_i .

The RMS jitter value (standard deviation of the histogram) provides an idea of how much the measured data is “spread” as shown in Figure 5.28.

For the reader with RF background, there is a relationship between the phase noise measured on a sinusoidal clock signal and the RMS jitter value on that clock signal [9, 10].

In the case of a data signal with only Gaussian jitter, the measured RMS jitter value of a jitter histogram is an estimate of the random jitter value, but in most cases even in the case of a clock signal there will be other jitter components apart from the random jitter. This means that the RMS jitter value is in most cases a very poor estimate of the signal jitter.

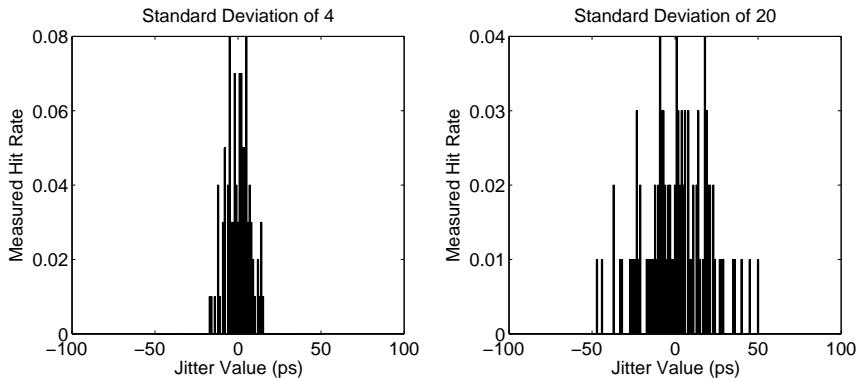


Figure 5.28 Comparing two histograms with different standard deviation (RMS jitter) values (left: $\sigma = 4$ ps; right: $\sigma = 20$ ps).

5.5.3 Peak-to-Peak Jitter

The peak-to-peak jitter value is a very commonly used measure of the jitter from a DUT. It corresponds to the difference between the two extremes of the jitter histogram (see Figure 5.29) or if looking to the array of measurement values it will correspond to the difference between the two extreme jitter values. In this example the peak-to-peak jitter value is 38 ps.

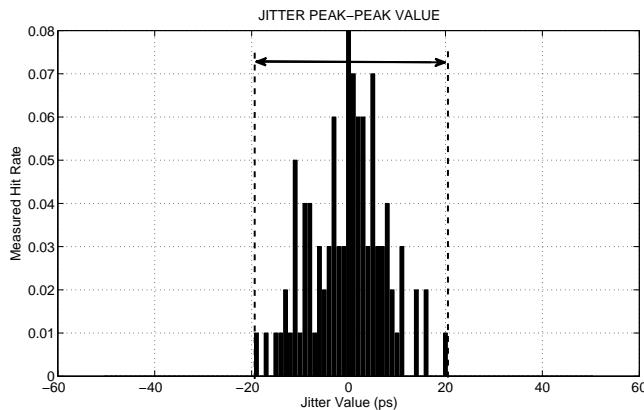


Figure 5.29 The jitter histogram peak-to-peak value.

Although peak-to-peak jitter is a common measure used to evaluate the jitter performance of a DUT driver, it is also the most problematic one. The reason is the fact that the random jitter portion of the driver jitter is theoretically not limited (i.e., its peak-to-peak value will be infinite if one measures it for a sufficiently long period of time). This means that any

peak-to-peak jitter value depends on the number of samples acquired. This can generate several issues, especially when correlating between different measurement setups. Figure 5.30 shows this fact by presenting four different independent measurements of the peak-to-peak jitter of a DUT with an equivalent-time oscilloscope using a different number of acquired samples. The results show that in the case of a small number of acquired samples (10 and 100) the jitter values vary significantly with the 100 samples case even showing a smaller peak-to-peak jitter value (each measurement was done independently of each other). For the 1,000 and 10,000 samples case the peak-to-peak jitter value increases with the number of samples showing an increase of 49% in the peak-to-peak jitter value for the 10,000 samples case when compared to the 100 samples.

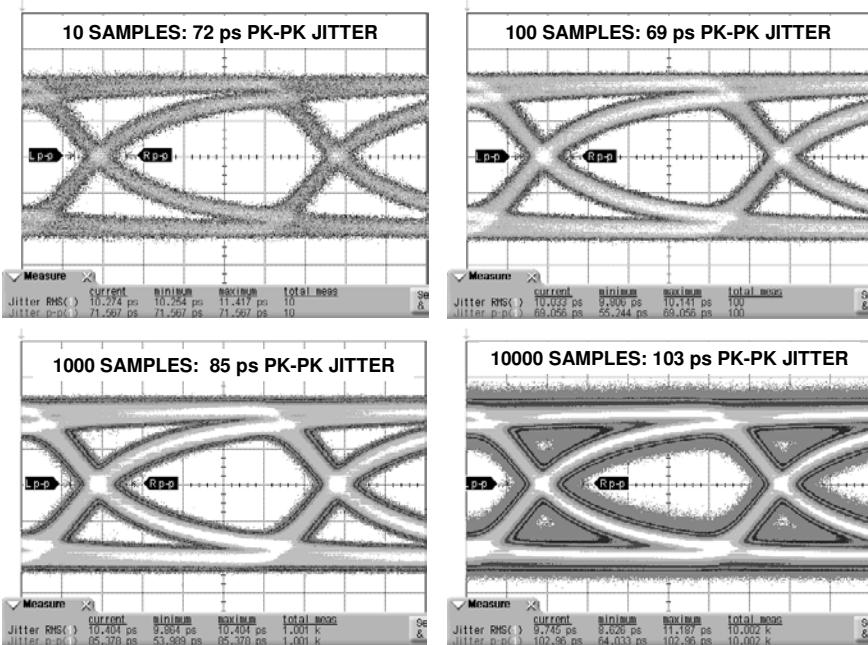


Figure 5.30 Variation on the measured peak-to-peak value with the number of acquired samples. The peak-to-peak jitter is measured with an equivalent-time oscilloscope and the measurement is restarted for each example.

5.5.4 Measuring the Jitter Spectrum

Although most standards define the jitter requirements for the DUT in terms of its random and deterministic jitter components, for design verification it

is important to know the spectral distribution of the jitter generated by the DUT. This information could provide an indication of the root-cause of the measured jitter (e.g., crosstalk from a system clock).

Although obtaining the jitter spectrum can be very simple, it does depend on the measurement approach. For example, with a real-time sampling oscilloscope, the waveform from the DUT driver can be acquired by sampling the signal in real-time and saving it for post-processing. It is then possible to perform a comparison of the pattern transition times with the transition times of an ideal nonjittered transition (since the data rate is known *a priori*). The resulting deviation values will be the jitter waveform sampled at the transition points of the pattern. It is now just a matter of applying the Fourier transform on the jitter waveform to obtain the jitter spectrum.

This procedure is shown in Figure 5.31. The maximum jitter frequency that can be measured corresponds to half of the data rate since it is only possible to sample a jitter value at a bit transition. However, this approach raises another issue due to the fact that apart from a clock pattern, there is no guarantee that a transition will exist for each bit on the pattern. One way to address this issue is to interpolate the jitter waveform between transitions for the bits that had no transition.

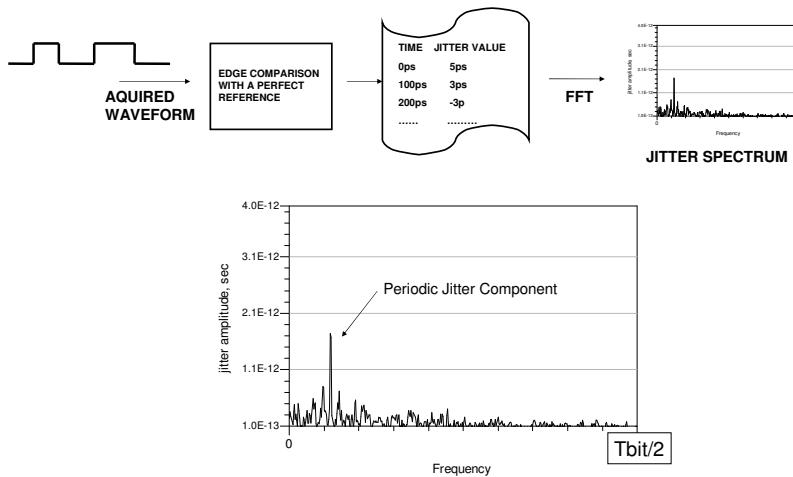


Figure 5.31 Measuring the jitter spectrum with a real-time sampling scope.

It is important to note that obtaining the jitter spectrum from a clock signal or a data signal are two different challenges. An instrument such as a spectrum analyzer or algorithm that is able to measure the jitter spectrum from a bit clock signal might not be able to measure the jitter spectrum from a data

pattern. The reverse is always true since measuring the jitter spectrum of a bit clock is a subset of measuring the jitter spectrum of a data pattern.

In [11] a technique is presented that allows the measurement of the jitter spectrum using a BERT or an ATE digital pin electronics. The main requirement for this technique is the ability to capture the pass/fail for each bit at-speed. The idea is to set the compare strobe timing not at the middle of the data eye as usual for a functional test but at the edge of the data eye. The jitter on the data signal will move the data eye left and right of the strobe point modulating in this way the pass/fail result for each bit on the data pattern as shown in Figure 5.32. The compare error signal can then be post-processed to obtain the jitter spectrum. The drawback is that although the jitter spectrum is obtained, the computed frequency amplitudes do not provide absolute jitter amplitude values but they can be displayed in relative values to the highest peak or to a calibration tone.

5.5.5 Random and Deterministic Jitter Separation

Several standards require the measurement of the random and deterministic jitter components of the jitter generated by the DUT driver (e.g., PCI Express [12]). The reasoning is that under certain conditions these two values allow the computation of the total jitter value for a given BER threshold which provides the timing margin or the data eye width from the DUT driver at that BER. The first assumption is that the random jitter is modeled by a Gaussian distribution with variance σ (see Appendix A). The second assumption is that the deterministic jitter is bounded with a peak value (DJ_{PEAK}) and a peak-to-peak value (DJ_{PK-PK}). With these two assumptions, the total jitter (TJ) value at a given BER can be computed by the following equation:

$$TJ(BER) = DJ_{PK-PK} + \alpha(BER) \sigma \quad (5.7)$$

Equation (5.7) means that the total jitter $TJ(BER)$ at a given BER threshold is equal to the deterministic jitter peak-to-peak value plus the random jitter (RJ) variance (RMS jitter) value multiplied by an $\alpha(BER)$ value that depends on the BER threshold. Note that since the tails of the random jitter are theoretically infinite, if no BER target is defined then the total jitter value would theoretically be infinite. Table 5.1 presents some of the typical values used for $\alpha(BER)$ and Section A.3 shows how the values presented in the table are computed.

Equation (5.7) is of little value without the RJ and DJ values. Typically what is available is a jitter measurement in the form of a histogram, a BER bathtub curve, a jitter waveform, or a jitter spectrum. There are several possible algorithms to obtain the random and deterministic jitter values from

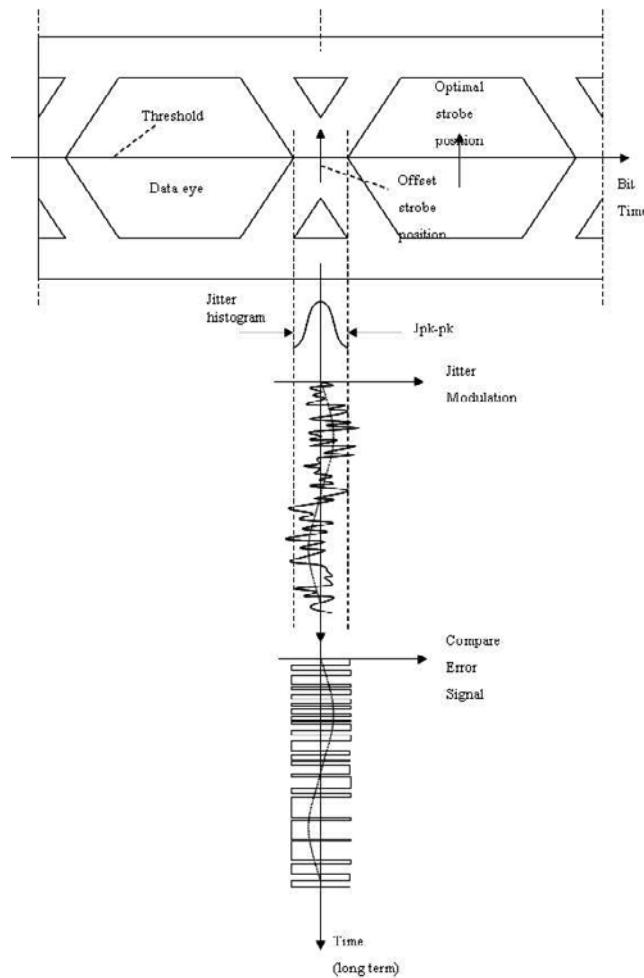


Figure 5.32 Measuring the jitter spectrum with a digital channel using at-speed capture. (From: [11]. ©2004 Bernd Laqua. Reprinted with permission.)

Table 5.1
Some Typical Used Values for $\alpha(BER)$

BER	$\alpha(BER)$
10^{-9}	11.996
10^{-12}	14.069
10^{-16}	16.444

jitter measurement data. The following subsections describe two algorithms: RJ/DJ separation based on the dual Dirac jitter model and RJ/DJ separation based on the jitter spectrum.

5.5.5.1 RJ/DJ Separation Based on the Dual Dirac Jitter Model

The dual Dirac jitter model assumes that the deterministic jitter is modeled by a dual Dirac distribution. The total jitter distribution is a convolution between the dual Dirac deterministic jitter model and the Gaussian distribution for the random jitter as shown in Figure 5.33. Appendix B discusses the dual Dirac model in detail.

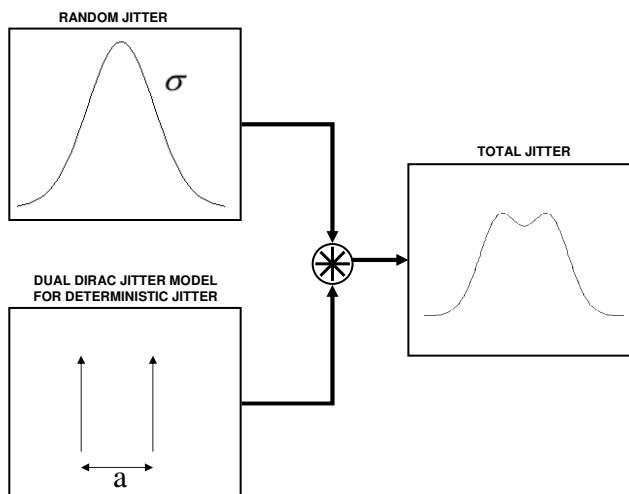


Figure 5.33 The dual Dirac jitter model.

With this model it is now possible to compute the RJ and DJ values from the measured histogram or bathtub curve. Since under this model, the deterministic jitter is bounded between two values, ($-DJ_{PK}$ to DJ_{PK}), it is possible to select a region outside these values (where the jitter is due only to random jitter) and fit a Gaussian curve to it. This is described graphically in Figure 5.34. Note that in the case of a BER bathtub curve the Gaussian distribution will have the form of an inverse erfc function (see Appendix A). Note also that there are two curves to be fitted on both sides of the jitter histogram or BER bathtub curve. Ideally they should be the same but in reality they will usually be slightly different from each other.

Algorithmic details for the RJ/DJ separation algorithm using the bathtub curve can be found in [13] and Appendix B. For the histogram case, the RJ/DJ jitter separation based on the dual Dirac model was presented in [14] although

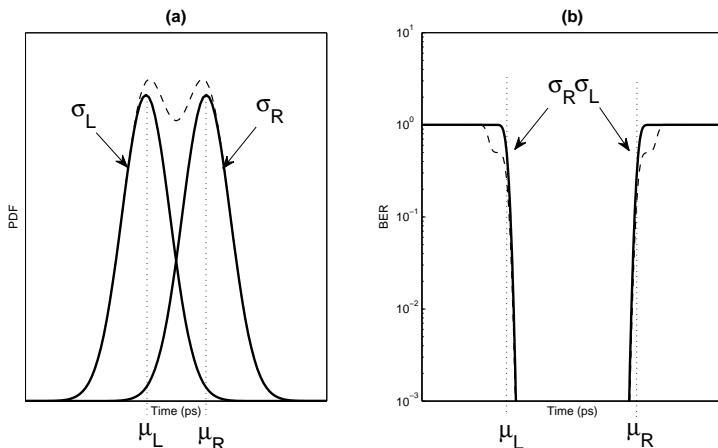


Figure 5.34 (a) Fitting a Gaussian curve to a jitter histogram or (b) fitting an inverse erfc() function to a bathtub curve.

using an automatic procedure to compute the fitting range. The key point is that the σ value (standard deviation or RMS value) of the fitted Gaussian distribution will provide the random jitter value (typically one uses the average of the two fitted Gaussian curves) and the distance between the average of the Gaussian distributions will provide the deterministic jitter value. Figure 5.35 shows one example of a RJ/DJ separation based on the dual Dirac jitter model implemented on an ATE platform.

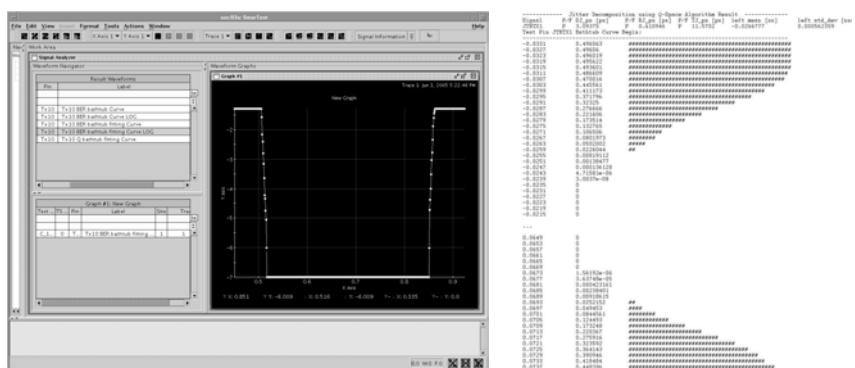


Figure 5.35 Jitter separation on an ATE platform. The left figure shows the GUI that is used during the test program debug and characterization phases to evaluate the jitter separation results, while the right figure shows the datalog that is used on a production environment where multiple pins and DUTs are measured. (Courtesy of Verigy.)

The choice of the fitting region is straightforward if one knows a priori the peak value of the deterministic jitter but unfortunately this is one of the values we are trying to determine. The best strategy is to choose a fitting range that is far from the center in the case of a jitter histogram or with low BER values in the case of a BER bathtub curve. This approach guarantees that the fitting range is outside of the deterministic jitter area, but in turn this requires that the histogram or bathtub curve have measured values in this area which typically requires longer acquisition times which in turn means a longer test time. This is exemplified in Figure 5.36 for a jitter histogram and a BER bathtub curve.

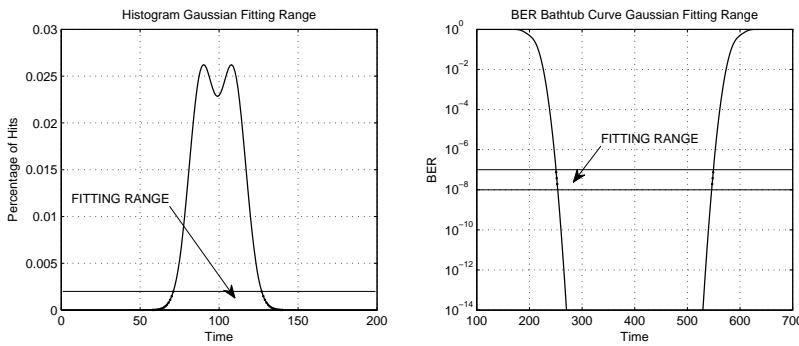


Figure 5.36 Fitting range of the histogram and bathtub curves.

The choice of the fitting range has an impact on the measurement results as shown in the example of Figure 5.37. In this example two different DUTs were measured. In the first DUT random jitter was injected so that the output jitter is dominated by random jitter, and on the second DUT both random and deterministic periodic jitter were injected. In both examples fitting ranges of $[10^{-3}, 10^{-5}]$ and $[10^{-7}, 10^{-9}]$ were used, respectively. The total jitter (TJ) was extrapolated to a BER threshold of 10^{-12} . The results show that in both examples the fitting range does have an impact on both the computed DJ and RJ values. Note that the values obtained with the $[10^{-7}, 10^{-9}]$ fitting range should provide a better accuracy than the ones obtained with the $[10^{-3}, 10^{-5}]$ fitting range.

It is important to note again that the RJ/DJ separation algorithm based on the dual Dirac jitter model is based on a very specific model for the deterministic jitter. Some authors have presented methods to further improve the accuracy of the algorithm by identifying the model in the distribution [15, 16].

Other approaches to compute the total jitter are possible with sometimes better results [17]. For a total jitter measurement a full BER measurement

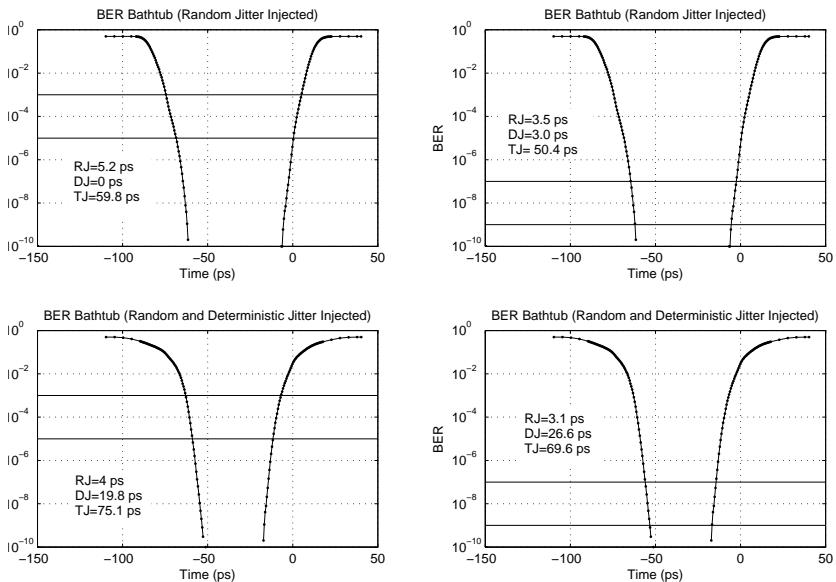


Figure 5.37 Demonstration of the impact of the fitting range in a signal dominated by random jitter (top) and by random and deterministic jitter (bottom). The two fitting ranges used in each example are $[10^{-3}, 10^{-5}]$ and $[10^{-7}, 10^{-9}]$.

at the target BER threshold is the closest one can get to a golden standard although it can take a significant amount of time. This means that when correlating to other bench instruments one needs to be aware of possible differences on the used algorithm before blindly comparing values (more on this topic in Section 5.5.7).

One challenge when using a fitting range for the RJ/DJ separation algorithm is the duality between measurement accuracy and test time. Figure 5.38 shows two bathtub curves acquired from the same DUT but with a different number of compared bits. In this example a fitting range of $[10^{-5}, 10^{-7}]$ was chosen, but if a pattern with only 10^4 bits is used then no measurement values are available in the fitting range. In this case the test engineer needs to change the fitting range, which might imply a reduction on the measurement accuracy as already shown in Figure 5.37.

The best option is to start with a large pattern size and a safe fitting range (i.e., in the case of a BER bathtub curve with very low BER values) and then start to optimize the test time by reducing the pattern size and changing the fitting region in a series of small steps while keeping a close eye on the RJ/DJ separation values with each change.

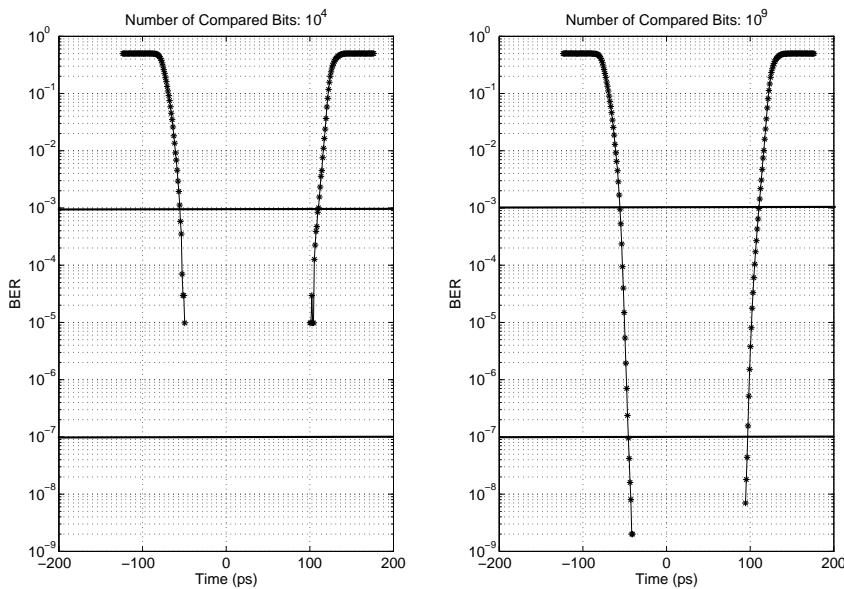


Figure 5.38 Bathtub curve acquired with a different number of compared bits for the same DUT (left: 10^4 bits; right: 10^9 bits).

Although the presented procedure for random/deterministic jitter separation and total jitter extrapolation shows limitations and accuracy issues due to the model assumptions, it is one of the few procedures that is used across the industry, and although when looking for the best accuracy it might not be the most appropriate one, for correlation between different measurement instruments it might be the most appropriate one since it is implemented in the same way in various instruments.

5.5.5.2 RJ/DJ Separation Based on the Jitter Spectrum

Another possible RJ/DJ separation technique is to use the jitter spectrum [7, 18]. The idea is to assume that the random jitter corresponds to the noise floor of the measured jitter frequency spectrum and the deterministic jitter will correspond to the individual spectral lines that are above the jitter spectrum noise floor as shown in Figure 5.39.

One simple approach to accomplish this is to define a threshold value above which a spectral line is considered as deterministic jitter and not part of the random jitter noise floor. Although for a jitter spectrum dominated by random and periodic jitter as in the example of Figure 5.39 this approach seems straightforward, it becomes much more challenging when trying to

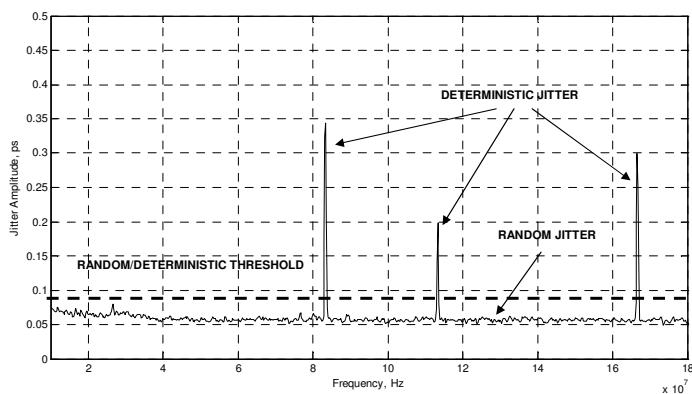


Figure 5.39 Jitter spectrum of a digital signal and the random deterministic jitter separation.

identify the jitter components for other types of deterministic jitter like ISI or BUJ as exemplified in Figure 5.40.

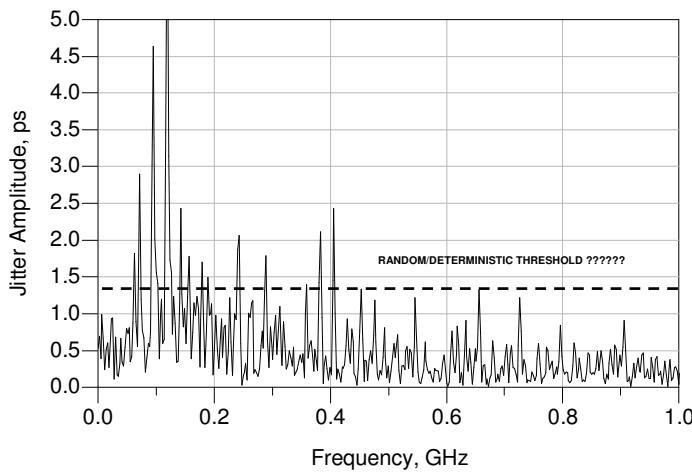


Figure 5.40 Challenge of setting a jitter separation threshold in the jitter spectrum results when multiple types of deterministic jitter are present (e.g., ISI and BUJ).

5.5.6 Measuring the Data Dependent Jitter

In the context of deterministic jitter, data dependent jitter (DDJ) is a critical value because it represents the portion of the deterministic jitter that is correlated with the pattern. Measuring the data dependent jitter requires

knowledge of the pattern under measurement; that is, where the pattern starts and its length. For some measurement instruments this is achieved by a pattern marker trigger capability (i.e., a trigger that indicates where the pattern starts since the DUT driver will be sending the pattern repetitively).

In the case of an ATE system, when using a digital pin electronics receiver, the knowledge of the pattern start and length is readily available in the functional pattern. For example, the receiver uses it for the functional test compare. One approach to measure the deterministic jitter would be to perform a jitter measurement for each bit of the pattern individually as shown in Figure 5.41.

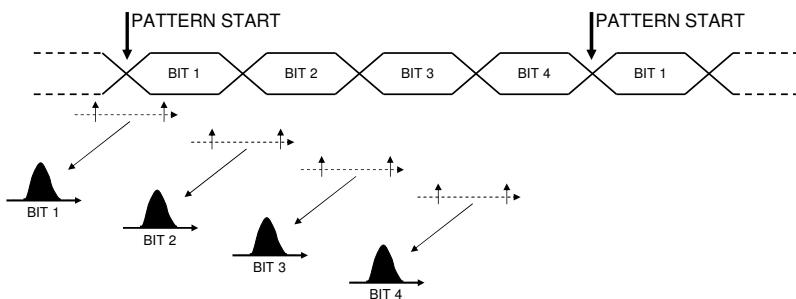


Figure 5.41 Measuring the jitter histogram of each individual bit in the data pattern.

The idea is to only measure the jitter histogram in the transition for bit X ignoring the contributions from the other bits in the pattern (i.e., if using an error count approach to obtain the histogram, only count the errors that happen at the particular bit X position). This procedure is repeated for the other bits in the pattern resulting in a jitter histogram for each bit position as shown in Figure 5.41. Assuming that the measurement setup consists of enough repetitions of the pattern, one would expect the jitter histograms for each bit to be equal if there is no data dependency on the jitter from the DUT driver. But if there are jitter components that are data dependent (e.g., ISI jitter from a lossy test fixture), then it should become visible by comparing the individual histograms for each bit (e.g., comparing the mean value of each histogram in regard to the expected bit transition instant).

5.5.7 Jitter Measurement Correlation

Correlation of jitter measurements can be a complex topic. The reason is that unlike with measurement variables like frequency where there is a clear standard on how the measurement should be done and measurement differences between different measurement setups can be traced back to the

instruments' accuracy by a well-defined procedure, for jitter measurements this is not the case.

This creates a situation where a test engineer in certain cases might not be able to decide which instrument or technique to use and what is the expected error from a given setup. This is shown in Figure 5.42 where different bench instruments and one ATE pin electronics card were used to measure the deterministic jitter from a pattern generator with sinusoidal and random jitter that was added through a voltage controlled delay line. Note that not only the instruments are different but they also use different jitter separation algorithms or the same algorithm with different parameters.

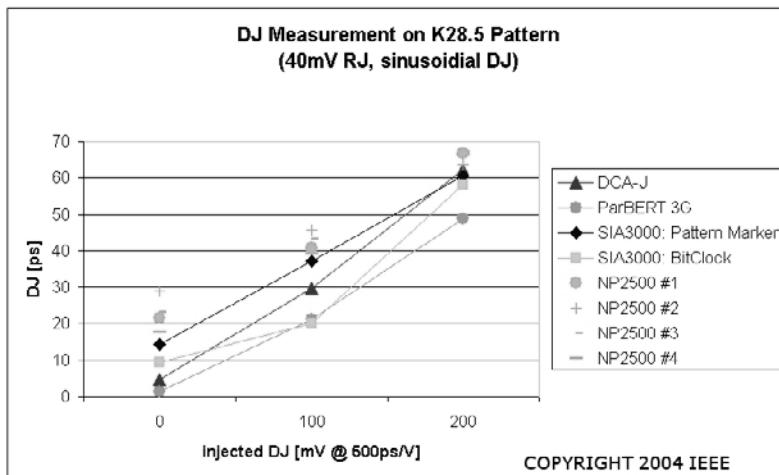


Figure 5.42 Deterministic jitter measurement on a K28.5 pattern from a pattern generator at 2.5 Gbps for different values of injected sinusoidal deterministic jitter (reprinted with permission from [13]).

References [17, 19, 20] present a comparison of selected instruments or algorithms, showing that significant differences can arise on the measured results for exactly the same DUT. This means that when correlating between different types of instruments or algorithms one should expect differences on the measured values that cannot be easily traced back to a difference on the accuracy of the instruments. Some standards have tried to address this challenge with mixed results. This challenge is also present on ATE applications, not only when correlating between ATE and bench instrumentation but also between different ATE systems.

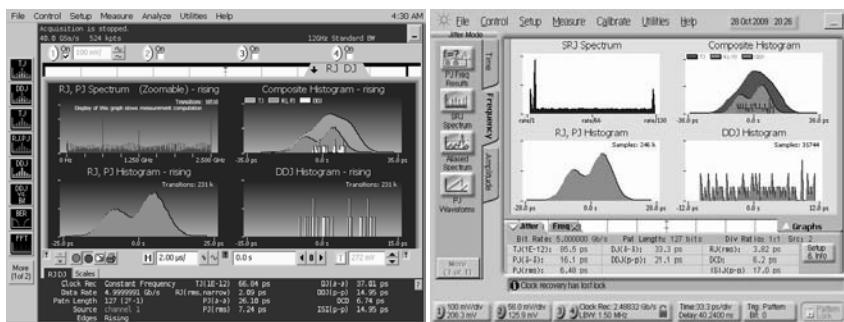
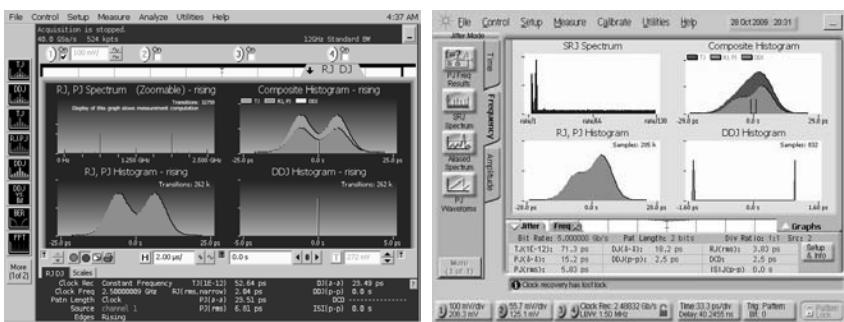
To better illustrate the jitter correlation difficulties, Figure 5.43 shows a jitter measurement of the same DUT using different measurement instruments. A PRBS $2^7 - 1$ and a bit clock data pattern at 5 Gbps are measured

with a real-time oscilloscope and an equivalent-time oscilloscope where the jitter is separated in its components by the jitter separation software of each instrument. The results are summarized in Table 5.2. They show that although the measured values are close they are not equal and therefore show the difficulties of correlating values with a very high accuracy between two different instruments. This difficulty can also be seen in Figure 5.44 where a bit clock is measured with a real-time oscilloscope and a spectrum analyzer with phase noise measurement software. In this case the correlation is more complicated because the phase noise measurement includes not only the random jitter but also any periodic jitter in the measurement range while on the real-time oscilloscope the random jitter is separated from the periodic jitter. The real-time oscilloscope measured a value of 2.14 ps while the spectrum analyzer measured a value of 3.76 ps. But if one analyzes the spectrum measured by both instruments (in the case of the real-time oscilloscope the jitter spectrum was computed by the jitter software) that is shown in Figure 5.45, it is possible to see that there is a good correlation on the periodic jitter component that is present on the jitter spectrum and this also explains the measured random jitter value differences since this periodic jitter component will be present on the phase noise measurement range. This is a good example to demonstrate the possibility of inferring the reasons for the two jitter measurement differences by a deeper analysis of the jitter results of two instruments.

Table 5.2Comparison of Jitter Measurements on a Bit Clock and a PRBS $2^7 - 1$ Pattern

Jitter Value	Real-Time Oscilloscope	Equivalent-Time Oscilloscope
RJ (CLOCK)	2.09 ps	3.82 ps
DJ (CLOCK)	37.01 ps	33.3 ps
RJ (PRBS $2^7 - 1$)	2.04 ps	3.83 ps
DJ (PRBS $2^7 - 1$)	23.49 ps	18.2 ps

Unfortunately the solution to the jitter measurement correlation challenge is not as easy as choosing a “golden” jitter measurement instrument since there is none. The best option is to compare the different measurement setups one intends to correlate using the same jittered signal and if possible with a calibrated amount of the different types of jitter. This topic is further discussed in Appendix I.

(a) PRBS $2^7 - 1$ pattern.

(b) Bit clock pattern.

Figure 5.43 (a, b) Comparison of a jitter separation measurement of a 5-Gbps PRBS $2^7 - 1$ and bit clock signal between an equivalent-time oscilloscope with a 70-GHz input bandwidth and a real-time oscilloscope with a 13-GHz input bandwidth and 40-Gsps sampling rate. Both instruments are from the same manufacturer and include jitter separation software.

5.5.8 Driver Amplitude Noise

The driver performance is also defined by the amount of amplitude noise present. While noise in the time domain is referred as jitter, in the voltage domain it is defined as amplitude noise. As already shown in Figure 2.33, amplitude noise will increase the amount of jitter and in this way not only decrease the timing margin on the driver signal but also the level margin in the form of a reduced data eye height. Like in the case of jitter, it is possible to analyze the amplitude noise using a histogram as shown in Figure 5.46. Amplitude noise also allows itself to categorization in different types like random and deterministic amplitude noise. In this case a lossy signal path not only adds deterministic jitter but also deterministic amplitude noise.

Like for the timing case, it is also possible to generate a bathtub curve for

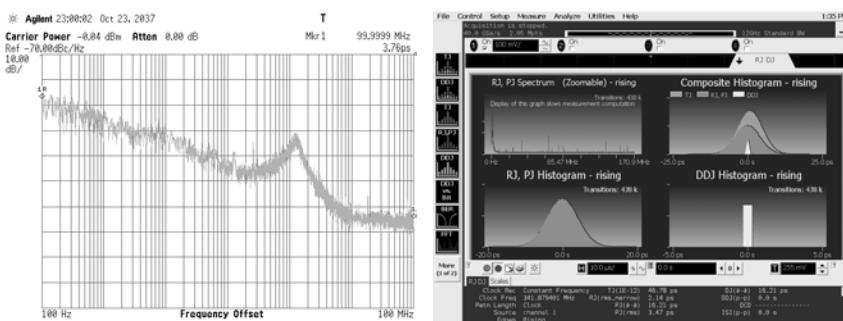


Figure 5.44 Comparison of a jitter measurement for a bit clock between a real-time oscilloscope with 13-GHz bandwidth and 40-Gsps sampling rate with jitter separation software and a spectrum analyzer with a frequency band between 3 kHz and 26.5 GHz with phase noise measurement software. Both instruments are from the same manufacturer. The real-time oscilloscope measured a RJ value of 2.14 ps and the spectrum analyzer measured a value of 3.76 ps.

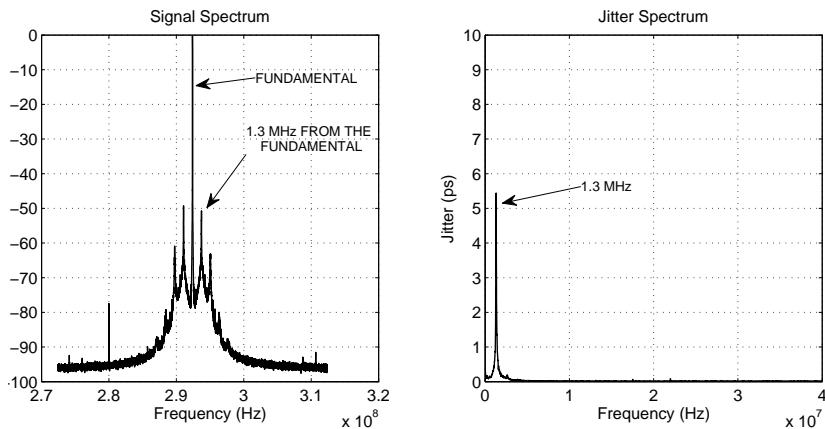


Figure 5.45 Comparing a measurement of a bit clock signal with regard to its jitter spectrum. The jitter spectrum obtained with real-time sampling oscilloscope (right) shows a periodic jitter component at 1.3 MHz that correlates with the phase modulation spur measured by the spectrum analyzer (left) that is located 1.3 MHz from the bit clock signal fundamental tone.

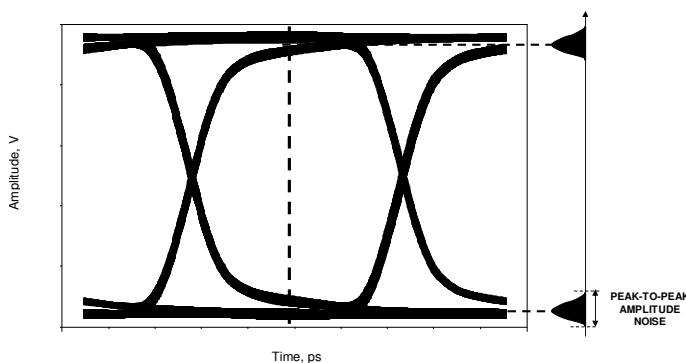


Figure 5.46 Histogram approach for the driver amplitude noise measurement.

the signal level. This fact allows one to generate a two-dimensional plot that provides a visualization of the effects of jitter and amplitude noise on the BER simultaneously as shown in Figure 5.47. The plot is also called a BER contour plot since each contour represents a collection of the same BER threshold.

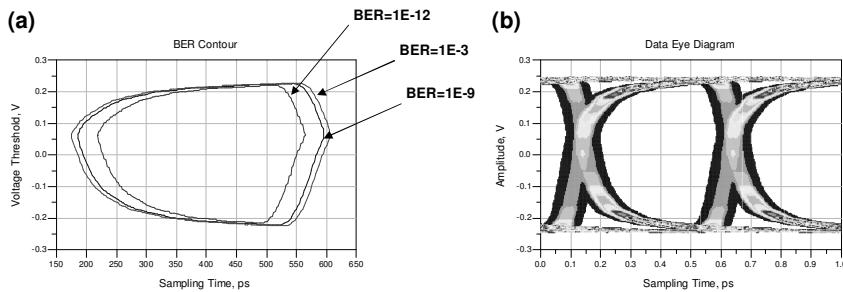


Figure 5.47 (a) Example of a two-dimensional BER plot (sampling time versus threshold voltage) also known as a BER contour plot and (b) the corresponding data eye diagram.

5.6 Fundamental Receiver Tests

This section presents some of the typical measurements for testing the receiver in a DUT I/O cell, and how these tests can be implemented in an ATE system.

5.6.1 Setup and Hold

The measurement on the DUT receiver side that corresponds to the driver skew measurement is the setup and hold measurement. Setup and hold time measurements determine the minimum time a data signal has to be stable before (minimum setup time) or after (minimum hold time) the corresponding latching clock edge. Figure 5.48 shows this relationship graphically.

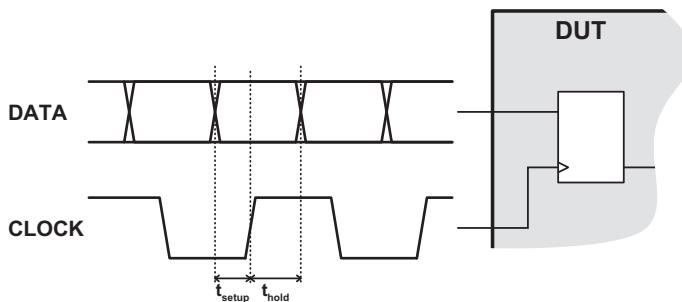


Figure 5.48 Receiver setup and hold measurements.

Like the skew measurements, setup/hold time measurements are implemented using the timing control and search capabilities of the digital pin electronics. Since the ATE drivers do not have the same timing deviations as might be the case for a DUT during the test pattern execution, in contrast to the skew measurement there is no need to restrict the pattern execution/compare on only a part of a full functional pattern set. However, test engineers should avoid the use of patterns that will generate significant data dependent jitter on the signal path from the ATE driver to the DUT because this would cause varying timing offsets between the clock signal and the data signals.

In order not to cause device failure due to high jitter on the data signals and to avoid potential problems due to driver pulse widths below the specified limit, the setup/hold measurement is done with the bits set to their nominal UI width. For the setup/hold time measurement of DUT receivers, all ATE drive edges for each of the data signals are moved uniformly during a timing search. This way, the complete data eyes of the data signals are moved relative to the associated clock signal as shown in Figure 5.49. In order to measure the setup time t_{setup} , the search is started with the active clock edge and the left data bit

edges aligned to each other ($t_{setup} = 0$). This results in a failing functional test. During the search, the data signal edges are moved into the negative direction relative to the active clock edges until a fail-to-pass transition for functional test results is detected.

The relative timing distance between clock and data signal at this fail-to-pass transition represents the minimum setup time that the DUT can handle. For measuring the minimum hold time t_{hold} , the search is started with the right edges of the data bits aligned to the active clock edges ($t_{hold} = 0$). The search for the hold time is done by moving the data signal eyes into the positive direction relative to the active clock edges. Again, the relative timing offset between active clock edge and right data eye edge at this fail-to-pass transition represents the minimum hold time supported by the DUT.

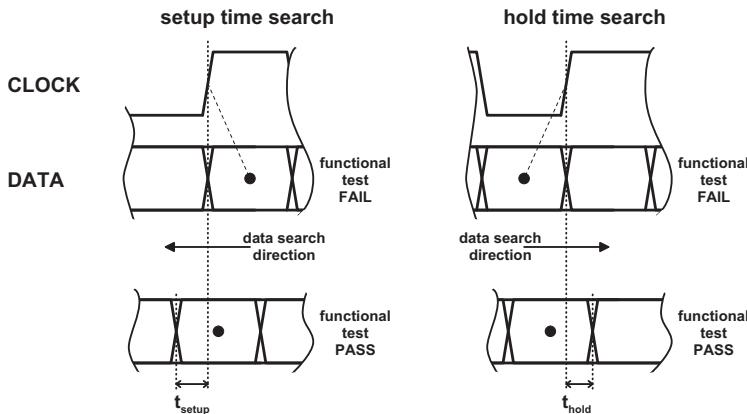


Figure 5.49 ATE search setup for receiver setup hold characterization.

5.6.2 Receiver Sensitivity

The objective of the receiver sensitivity test is to measure the minimum signal amplitude at the DUT's receiver input that still allows the receiver to correctly recover the data on the input signal. In design verification, the signal amplitude at the receiver input is reduced in a level search operation until the receiver starts to fail to recover the correct data.

In order to get the correct receiver sensitivity distribution for all pins and not only the worst-case minimum input amplitude voltage for a group of DUT receivers, this measurement has to be done serially for all DUT receivers to be measured. If each DUT receiver has an associated DUT dedicated output pin (e.g., in a far-end loopback configuration as described in Section 6.3.1.3) that allows a selective judgement whether a single receiver recognized its input

levels correctly, the receiver sensitivity measurement can be done in parallel on all receiver pins as shown in Figure 5.50. Only the pass/fail evaluation for each search step needs to be done separately for each receiver/driver pair in this case.

In production, one simply programs the ATE driver with a specific minimum amplitude to be guaranteed and checks for a passing functional test with this setting. Although this test looks very simple from its definition point of view, it can create significant challenges for a high-speed application, especially if very accurate measurements are required.

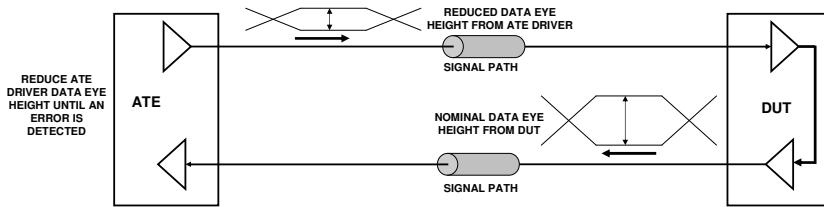


Figure 5.50 Example of a receiver sensitivity test where a far-end loopback is used on the DUT I/O cell.

The challenge starts with the fact that we are dealing with low signal amplitudes. Low amplitude waveforms with a good parametric performance are difficult to generate. Also receiver sensitivity is defined at the receiver input, but in a real setup a test fixture must be used where the signal will be degraded on its travel along the signal path from the ATE driver to the DUT due to the signal path loss and other effects described in Chapter 8. This signal degradation manifests itself in voltage loss, crosstalk from other signals, and amplitude noise from the test fixture (which is pattern dependent).

All these factors make it clear that for receiver sensitivity, one needs to be very careful with the measurement setup. One approach to address the challenge of generating low amplitude signals with good parametric performance is to use high-bandwidth attenuators (Section 7.11.2). This allows the ATE driver to deliver an amplitude value lower than its specifications by programming the driver to an amplitude value within its specifications and then using the high-bandwidth attenuator to obtain an amplitude signal lower than the ATE specifications. Regarding the other challenges, appropriate design of the test fixture (Chapter 8) and focus calibration (Section 9.3) are some of the approaches to address them.

5.6.2.1 Crosstalk Effects on Receiver Sensitivity Testing

Although crosstalk is a topic that is important for any high-speed test, due to the fact that on the receiver sensitivity test, the measurement instrumentation

will be driving very low amplitude signals, the effect of crosstalk from other signal lines might have a significant effect on the test result. Figure 5.51 shows a diagram describing how crosstalk might arise on a receiver sensitivity test. In this case, the ATE driver is driving a low amplitude signal to the DUT receiver while at the same time it is possible that the DUT driver is transmitting a signal with a normal amplitude to the ATE receiver. Crosstalk is proportional to the signal amplitude, rise time, and proximity between the aggressor and victim signals. Since higher data rates imply faster rise times, crosstalk is becoming a considerable effect.

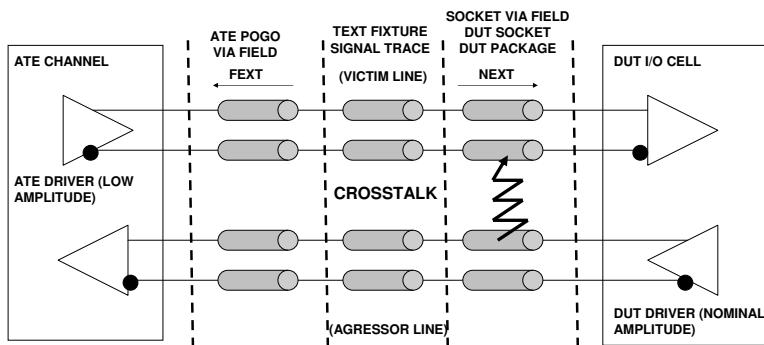


Figure 5.51 Diagram of the crosstalk effect on the receiver sensitivity test.

The fact that the ATE driver is sending a low amplitude signal makes this signal an ideal victim for crosstalk. Since the signal amplitude and rise time from the DUT are usually not under the test engineer's control (it is set by the IC design), the only variable the test engineer can have partial control over is the proximity of the victim and aggressor signals, but this is restricted only to the test fixture since the DUT package and ballout are also predefined and outside the control of the test engineer. Another important point is that crosstalk only occurs in signal transitions, which means that depending on the skew between signals, crosstalk effects might have catastrophic effects or not. This is exemplified in Figure 5.52.

As mentioned before, minimizing crosstalk on the test fixture as well as keeping a tight control on the channel skew are important to prevent effects similar to the one shown in Figure 5.52.

5.7 Receiver Jitter Tolerance

The objective of a receiver jitter tolerance test is to verify the ability of the receiver in a DUT I/O cell to tolerate or compensate for the timing uncertainties (jitter) that can be expected for real signals in a real application.

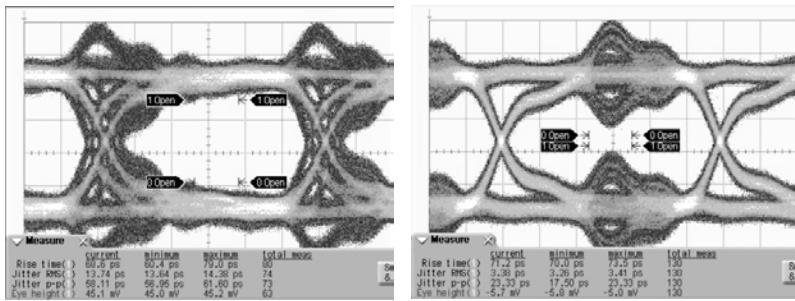


Figure 5.52 Effect of victim/aggressor skew on the crosstalk at the victim line. Note that the crosstalk amplitude is the same on both cases but in one it happens at the victim signal data eye edge with little effect on the eye center (left) and on the other it happens at the data eye center closing the data eye significantly (right).

In Section 2.4 timing jitter was categorized in different categories depending on its characteristics and causes. In a jitter tolerance test the objective is to generate a waveform with some of these jitter categories and test if the receiver is still able to function with a given BER. Figure 5.53 shows the jitter tolerance requirements from the PCI Express standard where it is possible to observe the different jitter categories that the DUT receiver needs to tolerate to be standard compliant. The next subsections will discuss some of the jitter injection categories in detail.

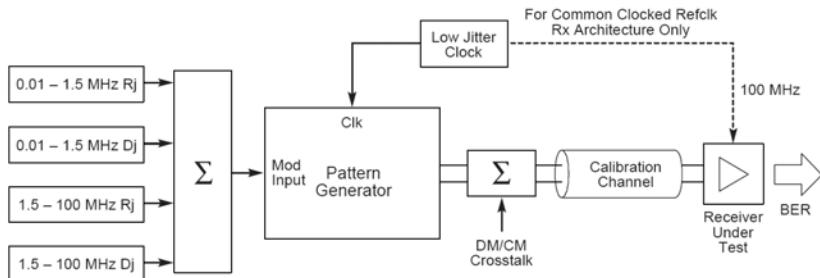


Figure 5.53 The PCI Express jitter tolerance compliance block diagram. (From: [12]. ©2002–2009 PCI-SIG. Reprinted with permission.)

5.7.1 Random Jitter Tolerance

Since any electronic system or part has some intrinsic noise, it is sometimes necessary to evaluate if a high-speed digital receiver is able to correctly

detect the data pattern in the presence of a certain amount of random jitter. As discussed in Appendix A, random jitter is modeled using a Gaussian distribution. The Gaussian distribution probability density function is described by the following formula:

$$p(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \quad (5.8)$$

The main challenge with random jitter tolerance is to provide a stimulus signal with the required random jitter content since on the receiver side it is only necessary to perform a standard functional test and measure if any bits failed. The typical technique is to use a noise source to inject random jitter on the stimulus driver through a voltage controlled delay line. Section 7.7 discusses noise sources in more detail. Figure 5.54 shows an example of a bit clock pattern with added random jitter in the time and frequency domains.

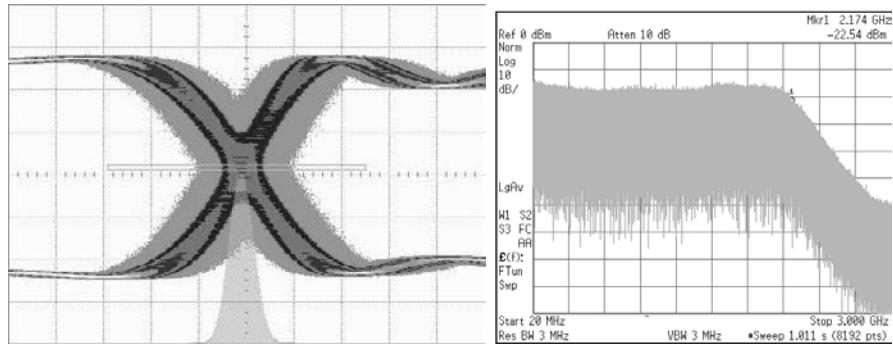


Figure 5.54 Example of a bit clock pattern with Gaussian random jitter injected through a voltage controlled delay line in the time domain by measuring the jitter histogram with an equivalent-time oscilloscope (left) and on the frequency domain with a spectrum analyzer (right).

Section I.2 discusses in detail possible calibration approaches for random jitter injection.

5.7.2 Sinusoidal Jitter Tolerance

Sinusoidal jitter tolerance is the oldest methodology for jitter tolerance testing [21]. One reason is that by changing the frequency of the injected periodic sinusoidal jitter it is possible to observe the behavior of the receiver in the frequency domain as shown in Figure 5.55. The measurement approach is to increase the sinusoidal jitter amplitude for each frequency until a fail is detected during a functional test. Basically it is a shmoof plot of the

periodic sinusoidal jitter frequency versus the sinusoidal jitter amplitude. It is important to note that depending on the pattern size used on the functional test, the measured jitter tolerance curve will correspond to a certain BER threshold. For BER thresholds like 10^{-12} , this test might take a considerable amount of time although faster approaches are possible [22].

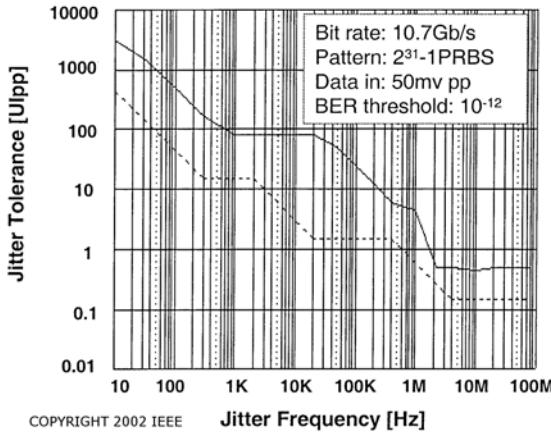


Figure 5.55 Example of a receiver jitter tolerance test with periodic sinusoidal jitter for a OC-192 application (reprinted with permission from [23]).

This type of test provides significant insight on the behavior of the receiver, and standards like OC-192 (SONET) [24] define specific jitter tolerance masks that the receiver must comply with.

Sinusoidal jitter with a peak-to-peak amplitude A is modeled by the following probability distribution function [25, 26]:

$$p_{\text{sinusoidal}}(x) = \begin{cases} \frac{1}{\pi\sqrt{(\frac{A}{2})^2-x^2}} & |x| < \frac{A}{2} \\ 0 & \text{Otherwise} \end{cases} \quad (5.9)$$

Note that the probability distribution is independent of the frequency of the sinusoidal jitter. Figure 5.56 presents a plot of the probability distribution for a periodic sinusoidal jitter signal with a jitter peak-to-peak amplitude of 4 ps ($A = 4$).

Appendix I.1 discusses methodologies for sinusoidal jitter injection calibration. Figure 5.57 shows a measurement on the time and frequency domain of the output of a pattern generator that has sinusoidal injected jitter.

Note that it is not possible to identify the sinusoidal jitter frequency in a histogram. This is only possible with a measurement in the frequency domain. On the frequency domain one can see the two side lobes appearing around

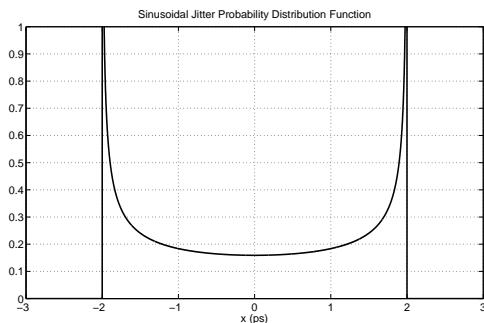


Figure 5.56 Sinusoidal jitter probability distribution ($A = 4$).

each harmonic spectrum of the bit clock signal as expected from modulation theory.

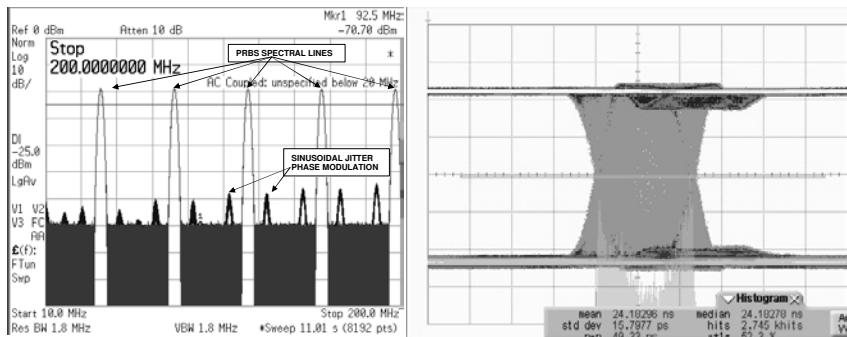


Figure 5.57 Sinusoidal jitter injection of a PRBS data pattern seen on the frequency domain with a spectrum analyzer (left) and time domain with an equivalent-time oscilloscope (right).

5.7.3 DDJ Jitter Tolerance

The ability of a DUT receiver to tolerate the data dependent jitter (DDJ) is a critical parameter for a high-speed I/O cell and is usually specified in modern high-speed standards. Unlike jitter tolerance to sinusoidal jitter that is typically in the low frequency range, DDJ adds high-frequency components to the signal jitter spectrum.

The specifications for DDJ tolerance are sometimes defined using a certain trace length on a printed circuit board with a specific geometry and dielectric material type. Figure 5.58 shows an example of a printed circuit board implemented to address several DDJ standards.



Figure 5.58 Printed circuit board on FR4 containing several traces compliant with different high-speed standards (courtesy of Verigy).

Figure 5.59 shows a 5-Gbps digital signal after traveling though one of the DDJ injection signal traces in Figure 5.58 showing the large amount of DDJ added to it. It is the job of the receiver to correctly recover the signal among all that jitter.

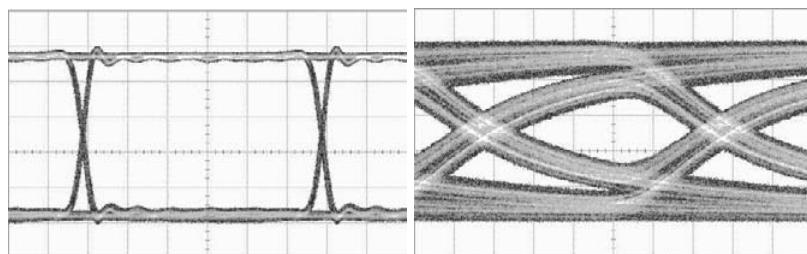


Figure 5.59 A 5-Gbps PCI Express compliance pattern before (left) and after (right) traveling through a DDJ injection PCB signal trace.

The approach of using a printed circuit board to inject the required DDJ, although usable in a bench setup, brings considerable challenges on an ATE environment when trying to test multiple I/O cells at the same time. One approach to address this challenge is to use a filter instead of a real printed

circuit board [27]. This would allow the filter to be directly assembled into the ATE test fixture. In [28] a filter design is presented that tries to mimic the frequency response of the printed circuit board trace. Another approach discussed in Section 9.8.3.4 is to use the equalization capabilities of some ATE pin electronics architectures to inject DDJ.

5.7.4 Testing the Receiver Equalizer

As already discussed in Section 2.6.4, modern high-speed digital receivers contain an equalization circuit that in most cases is controlled by a set of variables (e.g., the tap coefficients of a DFE receiver equalizer). These variables are usually set through registers or in some implementations automatically to an optimal value for the link signal path loss. This optimal value is typically determined by a link initialization phase.

The jitter tolerance test discussed in Section 5.7 assumes that the DUT receiver equalizer is already set to some optimal value. But in the design verification or characterization phase, the objective might be to identify the optimal settings for the equalizer or verify that the settings that are automatically found by the link initialization algorithm are the correct ones as shown in Figure 5.60.

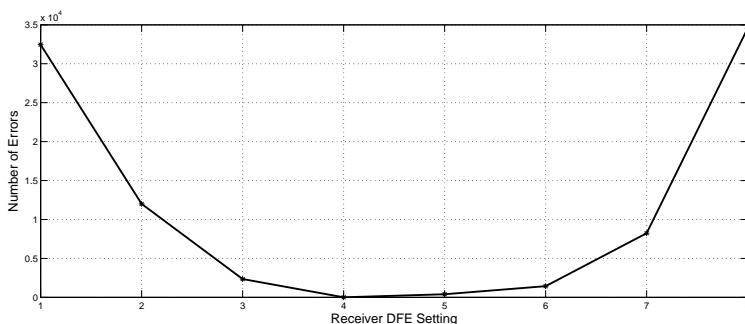


Figure 5.60 Measuring the ideal setting for a receiver DFE by using the number of errors on the received pattern. Note that this measurement is valid for a specific link loss.

In the case of trying to identify the optimal values for the equalization settings, one option is to add the lossy signal path that the equalizer is intended to compensate on the test fixture. For testing the receiver equalizer this signal path could be switched on and off between the ATE pin electronics and the DUT receiver through relays as shown in Figure 5.61. Although this option is straightforward and does not require any special features on the ATE pin electronics, it can result in a complex implementation for high pin-count

devices. It also is limited to a single or a very limited number of lossy signal paths which might be insufficient for some applications that are designed for a large variety of signal path losses.

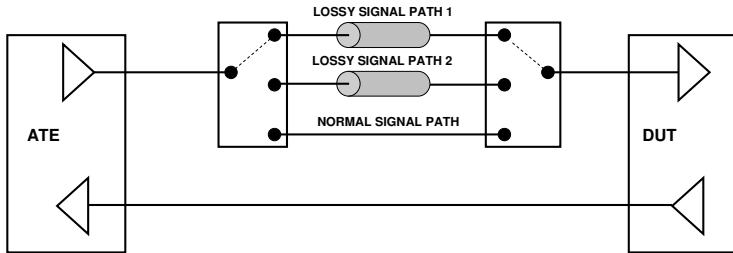


Figure 5.61 Receiver equalization testing by adding a lossy signal trace in the DUT test fixtures.

Another option is to inject the DDJ of the expected lossy signal path on the stimulus waveform to the DUT receiver using the ATE pin electronics (see Section 9.8.3.4). Note that the injected DDJ value and spectrum needs to have a correspondence to the PCB lossy signal path that is expected to be used in the link that the DUT is intended for. This approach provides the maximum flexibility since the DUT receiver equalizer can be tested for multiple different signal path losses without adding complexity to the test fixture. On the other hand it does require the pin electronics driver to have this capability which is not trivial.

In the case of a DUT I/O cell receiver that automatically sets the equalization parameters through a link initialization step, it is possible by implementing appropriate design for test (DfT) blocks on the DUT I/O cell to read the final settings of the equalizer and also overwrite those settings [29]. This would allow the test engineer to find the optimal equalization parameters for a given DDJ profile and compare them with the values obtained through the automatic link initialization process.

5.8 PLL Characterization

This section describes some of the tests that are targeted specifically to the PLL circuitry that is usually part of a high-speed I/O cell.

5.8.1 Jitter Transfer

Jitter transfer is defined as the amount of jitter in the input signal of a PLL circuit that is transferred to the output signal of the PLL. Figure 5.62 shows

a diagram describing a possible jitter transfer measurement setup on the PLL of the DUT reference clock. Typically this measurement is displayed in the frequency domain showing the ratio of measured jitter versus the injected jitter at a given frequency as shown in Figure 5.63.

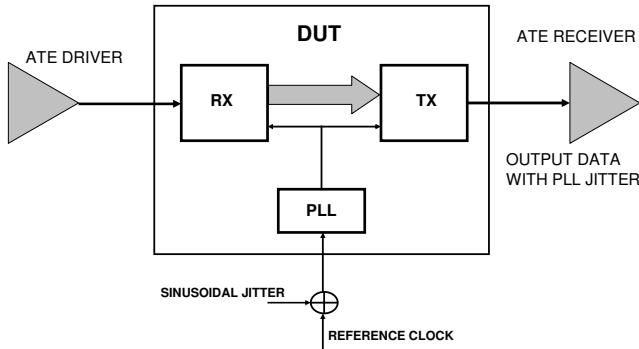


Figure 5.62 Jitter transfer measurement setup.

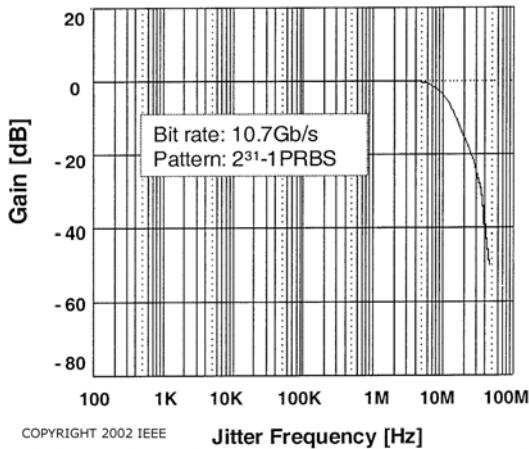


Figure 5.63 Jitter transfer measurement example (reprinted with permission from [23]).

The idea is for each frequency to insert a certain amount of sinusoidal jitter at that frequency in the input signal and measure the sinusoidal jitter amplitude at that frequency on the output signal, displaying the ratio between them using a logarithmic scale as described in the following equation:

$$\text{Jitter Transfer}(f) = 20 \log_{10} \left(\frac{\text{Output Jitter}(f)}{\text{Input Jitter}(f)} \right) \text{ dB} \quad (5.10)$$

The challenge with this measurement is twofold. First, it is important to inject a specific amount of sinusoidal jitter at a specific frequency. This might require a calibration of the sinusoidal jitter amplitude (see Section I.1 for more details). The second challenge is to measure the sinusoidal jitter amplitude at the current measurement frequency step that is present on the output of the DUT. This measurement is not trivial, since the amplitude of the sinusoidal jitter at a specific frequency must be separated from all the other sources of jitter like random jitter or other deterministic jitter sources at different frequencies (e.g., crosstalk from another clock on the DUT). One simple approach when the device output jitter is dominated by random jitter is to use the “RJ subtraction method” that is discussed in Section I.1.2.

Other approaches have been developed for measuring the jitter transfer based on random jitter injection instead of using periodic sinusoidal jitter [30, 31]. The advantage of using random jitter to measure the jitter transfer is that it can be done with a single measurement instead of a shmoof test of different sinusoidal jitter frequencies.

5.8.2 Frequency Offset

Frequency offset is a test designed to measure the ability of the DUT to work properly when the frequency on one of the DUT pins (e.g., the reference clock) does not correspond exactly to the nominal value. Figure 5.64 shows an example of a possible measurement setup for a frequency offset test.

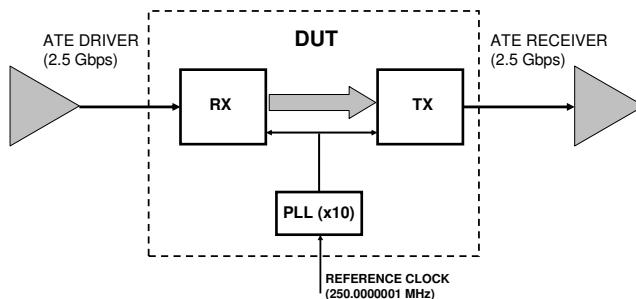


Figure 5.64 Example of a frequency offset measurement setup.

A frequency offset is performed on the DUT reference clock or in another input like the incoming pattern data rate to the DUT I/O receiver, while keeping all the other inputs at the nominal frequency. A functional test is then performed to verify if the DUT is still working correctly with this setup. The challenge with this type of measurement is that typically the amount of frequency offset required is just some tens or hundreds of parts per

million (ppm) of the nominal frequency. This requires that the ATE timing architecture is able to offset the frequency of a driver channel by that amount while keeping the other ATE channels on the nominal frequency.

5.8.3 Spread Spectrum Clocking

Spread spectrum clocking (SSC) [32] is a requirement in several high-speed digital standards (e.g., FBDIMM and Serial ATA) and optional in others (e.g., PCI Express). The objective is to spread the energy of the clock (and therefore data) over a percentage of the frequency band. The reason is to reduce the power at the clock frequency (which is the highest) by spreading it using a specific modulation profile and in this way making compliance with emission standards easier. Figure 5.65 exemplifies how spread spectrum clocking works for the Serial ATA standard [33].

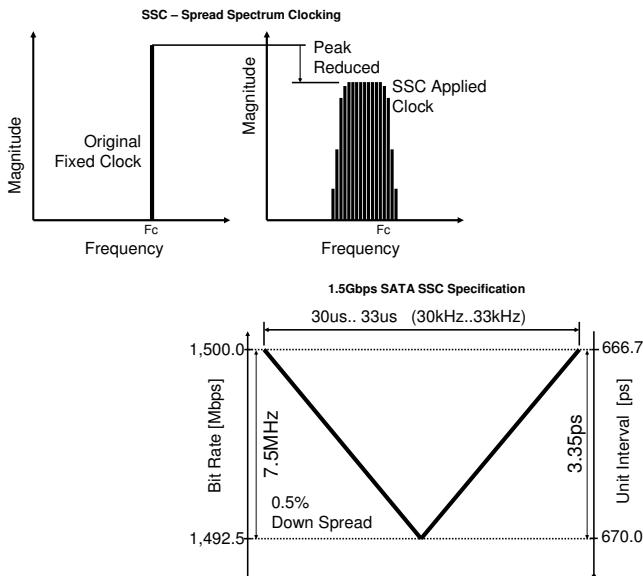


Figure 5.65 Description of how spread spectrum clocking (SSC) works (top: the spectrum peak reduction effect of SSC; bottom: the clock modulation profile).

From the figure it is possible to see that spread spectrum clocking is a kind of frequency modulation (FM) of the reference clock of the device. The bit clock is modulated with an extremely low modulation frequency compared to the bit rate so that the bit rate can be very stable for a short period of time with no harmful effect on the BER.

5.8.3.1 Measuring SSC with a Sampler⁴

A sampler ATE instrument can be used to measure a spread spectrum clock [34]. Figure 5.66 shows an example of a measurement configuration and frequency domain illustration of a spread spectrum clock. The data bit clock pattern is set to 1.5 Gbps to simulate a Serial-ATA (SATA) [33] application clock which is at 750 MHz. SSC for SATA is specified as 0.5% down spread so that the frequency deviation is 3.75 MHz at 750 MHz. Approximately 33 kHz sawtooth-shaped FM is applied. The signal is sampled with 4,096 points at 24.798 Msps so that the baseband is 12.399 MHz wide. The 750 MHz is aliased at around 6 MHz in the 12.399-MHz baseband, and the frequency deviation of 3.75 MHz can be appropriately captured in the baseband, as shown in Figure 5.66(b).

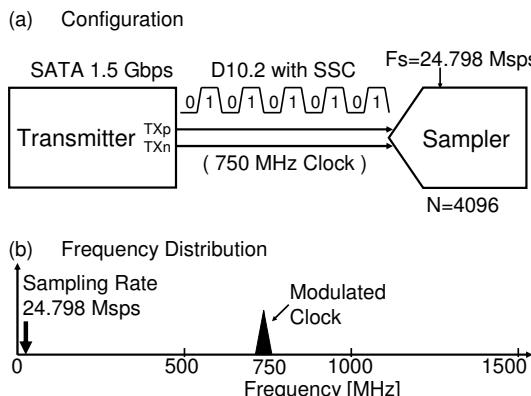


Figure 5.66 (a, b) Spread spectrum clock measurement configuration.

The test signal was measured with a spectrum analyzer and is shown in Figure 5.67 showing the frequency modulation on the clock signal.

The block diagram of the data processing approach to obtain the modulation waveform of the spread spectrum clock is presented in Figure 5.68, which is the application of the orthogonal demodulation [35, 36].

The sampled signal from the measurement setup of Figure 5.66 is shown in Figure 5.69(a) which can also be seen in the frequency domain as the spectrum shown in Figure 5.69(b).

To obtain the modulation signal, the cosine and sine reference (F_{ref}) waveform data are provided and multiplied with the sampled waveform data. The multiplied data (cosine and sine components) split the beat signal and the sum signal components, respectively. Then the data are low-pass filtered.

⁴Contributed by Hideo Okawara.

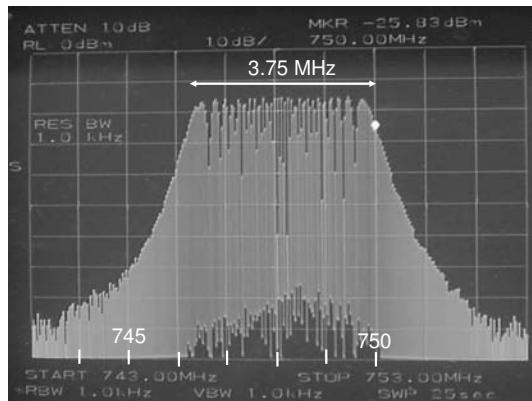


Figure 5.67 Clock with spread spectrum measured with a spectrum analyzer.

The extracted beat signals of cosine and sine components are processed with the arc tangent mathematical operation reconstructing the instantaneous phase trend of the spread spectrum clock. The differential operation to the instantaneous phase trend derives the instantaneous frequency trend. Multiplying the reference frequency with the instantaneous frequency trend reveals the SSC modulation trend as shown in Figure 5.69(c).

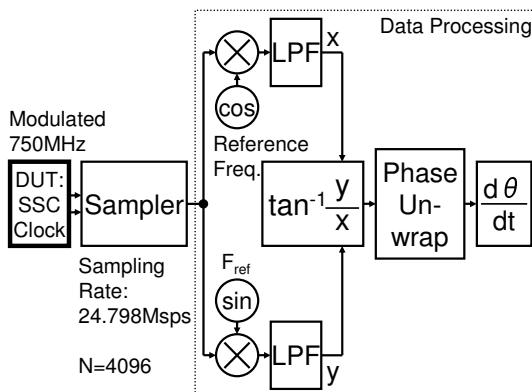


Figure 5.68 SSC analysis with the orthogonal demodulation method.

It is also possible to measure a spread spectrum clock with a digital pin electronics ATE channel as described in [37].

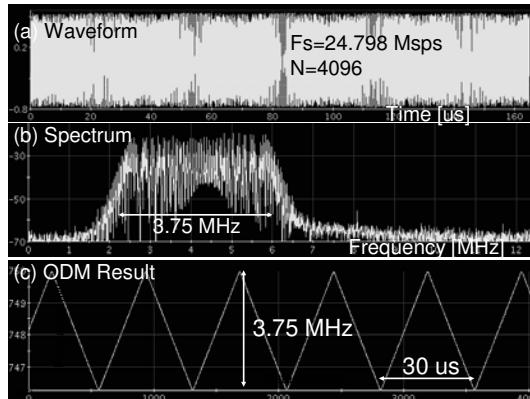


Figure 5.69 (a–c) SSC spectrum and reconstructed frequency trend.

5.9 Other Tests

5.9.1 Impedance Tests

In order to achieve the high data rates of high-speed I/O interfaces, the I/O driver and receiver circuitry of the devices usually has to provide a proper signal termination. For differential interfaces, such a termination is mandatory because it transfers the constant current that flows in different directions for high and low states into a voltage difference that can be recognized by the differential comparator in the device I/O circuit. The impedance of a differential driver has to match the receiver impedance to avoid signal reflections on the signal path. For single ended interfaces that operate at high data rates, proper termination is also important to avoid signal reflections that would disturb data transmission. The different termination schemes that typically are used in high-speed I/O circuits are shown in Figure 5.70.

Since the terminations in the I/O circuitry of a device can have a significant impact on the performance of the high-speed I/O signal path in a system, it is an important task to measure the I/O terminations at least in the device verification phase. The measurement of the I/O terminations usually is referred to in a general way as impedance test, although it is important to note that in an ATE environment this is usually restricted to the measurement of the DC impedance (termination resistances) and not the AC impedance of a DUT that also takes parasitic inductance and capacitance into account. The AC impedance is measured through a return loss measurement that is described in Section 5.9.2.

The measurement of the termination impedances on the ATE is done using the PMU circuitry that is usually included in every ATE digital pin

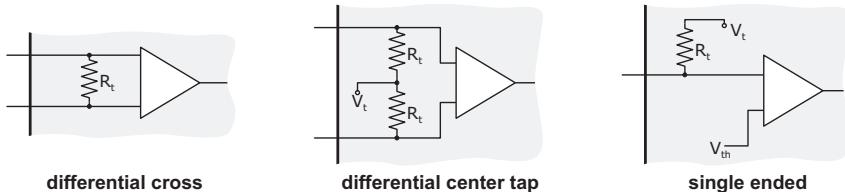
Driver**Receiver**

Figure 5.70 Typical termination schemes used in high-speed I/Os.

electronics card. With the PMU, the test engineer can force a current or voltage to the DUT pin and measure the voltage or current the DUT sets up as a reaction to this forced stimulus. For impedance measurements the PMU is used to determine current/voltage pairs that then are used to calculate the device impedance using Ohm's law. The boundary condition a test engineer has to consider, though, when measuring impedances of a real device is the fact that he does not know the exact values of internal device voltages that might have influence on the test results if the test setup is not selected in a correct way.

The device internal voltages V_t , V_d , V_{dP} , and V_{dN} as used in Figure 5.70, for example, are not known exactly and usually also cannot be measured directly in an accurate way. If these voltages, however, are not considered in impedance measurements that are based on a single voltage-current pair, they cause wrong impedance calculations. The reason for this is that when an impedance is calculated based on a single voltage-current pair, it inherently is assumed that if the force current I_f is equal to zero, then the measured voltage V_m also is equal to zero as shown in the left graph of Figure 5.71. This is not the case, however, and the internal device voltages cause an offset of the voltage current relationship that is the base for our impedance calculation as shown in the right graph of Figure 5.71.

Since the impedance R is represented by the slope of the voltage current relationship, a simple and effective way to avoid this source of inaccuracies is to measure two voltage-current pairs and calculate the impedance as the relationship of the measured voltage difference for a forced current difference and this current difference as shown in Figure 5.72.

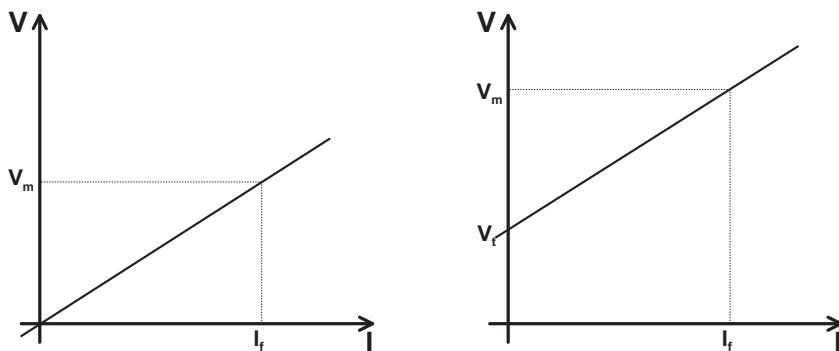


Figure 5.71 Difference of single point impedance measurements without consideration of DUT-internal voltages (left) and with consideration of DUT-internal voltages (right).

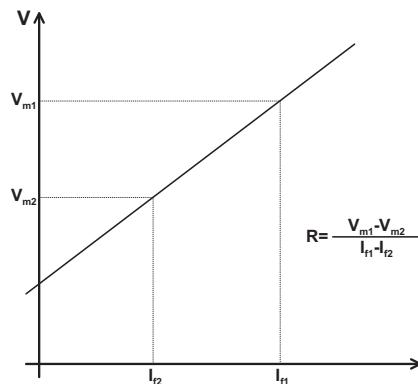


Figure 5.72 Two-point impedance measurement for voltage offset compensation.

All the test engineer has to ensure when performing such a two-point measurement is that the selected force currents I_{f1} and I_{f2} are in an area that has a linear characteristic for the I/O impedance. This is not always the case because I/O impedances often are realized with the help of biased transistors that might have a nonlinear behavior in certain regions.

The measurement setups required to determine the impedance values for single-ended and differential I/Os vary slightly and are described in the following paragraphs. All measurement setups described support the required two-point measurement approach. Of course, as for all other DC measurements, and also for impedance measurements, the pins to be measured have to be configured into a static state.

Single-Ended Signals

For the single-ended impedance measurement, the setup is straightforward. The PMU of the ATE is connected to the input or output pin to be measured and for each of the two force currents selected by the test engineer, the voltage the device pin settles to is measured. The impedance is derived as the relationship between the measured voltage difference and the force current difference (see Figure 5.72).

Differential Signals

Impedance measurements for differential I/Os come in two flavors. The first is the common mode impedance where the impedance between each leg of the differential interface and the common connection point of the two legs inside the DUT is measured. For differential receivers, this, of course, only makes sense if a center tap termination scheme is used.

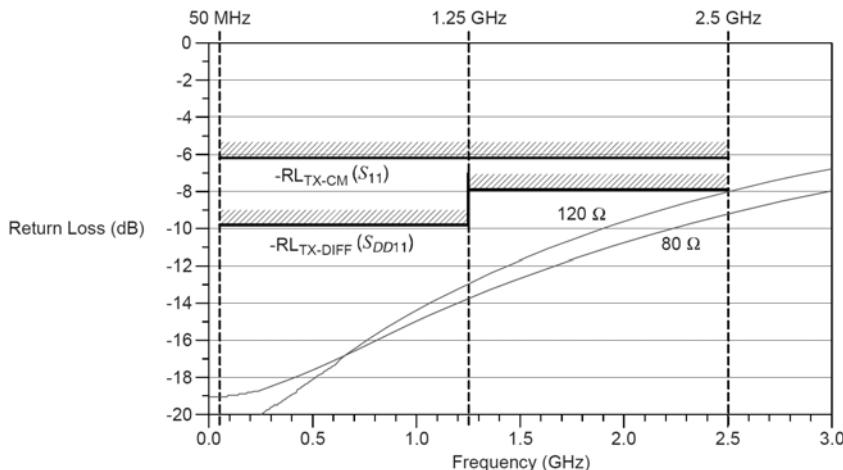
The measurement configuration to do the common mode impedance measurements is the same as for single-ended impedance measurements. The only additional item a test engineer has to take care of is the fact that the differential leg that is not being measured is kept at the same condition for both measurement executions of the two-point measurement on the opposite leg. The easiest way to ensure this is to disconnect the negative leg of a differential signal when the positive leg is measured and vice versa or to use the PMU on the leg that is not being measured and apply a constant voltage while the measurements on the other leg take place.

The other impedance measurement flavor for differential signals is the differential impedance. Here, the impedance between the two legs forming the differential signal is measured. The measurements for the differential impedance are independent of the differential termination scheme (floating cross termination or center tap termination) that is used for a differential signal. In order to measure the differential impedance, each of the two legs forming the differential signal is connected to a separate PMU. One of these PMUs forces the test current while the other one sinks the same test current. This ensures that there is no influence of other device internal current paths that might have an impact on the measurement result. In this PMU configuration, the voltage that settles on the differential legs is measured per leg and the voltage drop over the differential impedance is calculated as the difference of the two voltages that are measured on the two legs.

For the differential impedance measurement that is done in this form, it is not necessarily required to do a two-point measurement because it is ensured that no parasitic currents or voltages influence the measurement due to the selection of the same source and sink current on the opposite legs.

5.9.2 Return Loss

Return loss is an important parameter that is specified in several high-speed digital standards. The return loss specifies the amount of the signal energy that is reflected at the package terminals of either a driver or receiver pin due to impedance mismatches between the test fixture $50\ \Omega$ signal trace and the transition to the package, the package substrate signal traces, and the DUT driver output impedance or the receiver termination. It is easy to see that one would like to have all the signal energy delivered to the receiver internal circuit with no energy reflected. If a significant amount of the input signal energy is reflected before arriving at the internal receiver circuitry, it will have a smaller amount of the input signal to work with, making the probability of an error higher. Figure 5.73 shows the return loss requirements on the receiver from the PCI Express standard [12].



Note: The $80\ \Omega$ and $120\ \Omega$ traces are for informational purposes only and do not reflect any requirements.

Figure 5.73 Return loss receiver requirements from the PCI Express standard (From: [12]. ©2002–2009 PCI-SIG. Reprinted with permission.)

Return loss (RL) is defined by the following equation:

$$RL = 20 \log \left(\frac{Z_L - Z_0}{Z_L + Z_0} \right) \text{ dB} \quad (5.11)$$

where Z_L is the impedance of the receiver input and Z_0 the impedance of the transmission line feeding the input signal to the receiver. Detailed presentation on basic transmission line theory and return loss can be found in [38].

Note that on the requirements presented in Figure 5.73 the common mode (CM) and differential (DIFF) return loss are distinguished. This is needed when defining return loss for a differential interface like PCI Express.

The standard approach to measure return loss is to use a vector network analyzer as shown in Figure 5.74 or a time domain reflectometry (TDR) instrument with the appropriate post-processing software. It is important to note that since the DUT is embedded on a test fixture as shown in Figure 5.74, it is necessary to de-embed the test fixture to obtain accurate measurements. Reference [39] provides a discussion on de-embedding techniques.

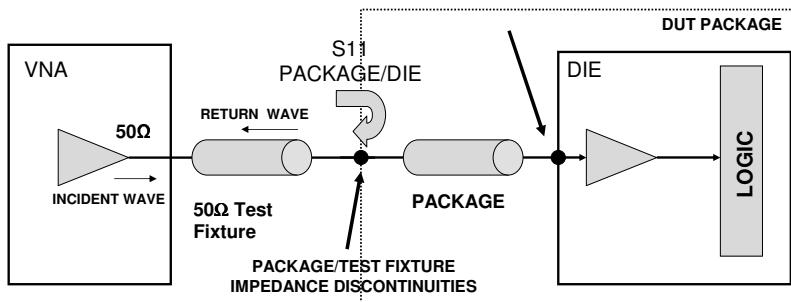


Figure 5.74 Measuring return loss with a network analyzer.

Return loss is usually measured on a bench setup with a vector network analyzer, although some ATE systems provide this capability also.

Figure 5.75 shows an example of a return loss measurement on a unidirectional interface (requiring a return loss measurement on the driver and receiver) and on a bidirectional interface where only one measurement needs to be done.

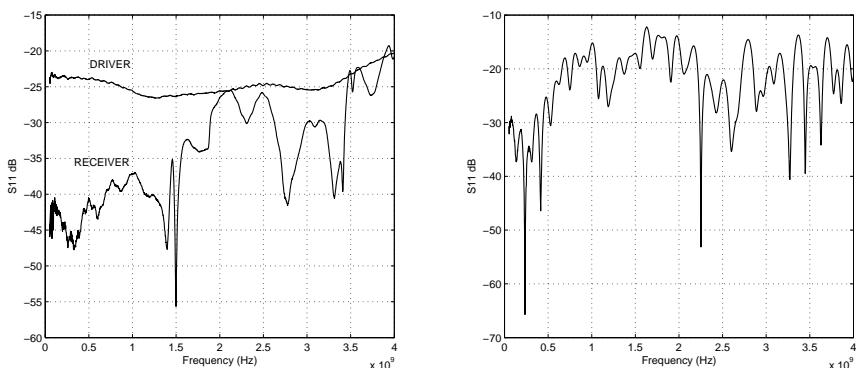


Figure 5.75 Example of the return loss measurements on a bidirectional interface (left) and unidirectional interface (right).

5.10 Measurement Errors

One important question when performing any measurement is what is the associated error to the measured value. Errors can be classified in two categories: systematic and random [40, 41]. There is little a test engineer can do in regard to systematic errors. The measurement instrument specifications should provide guidance for the systematic error one can expect. For example, Figure 5.76 shows one example of the specification for an ATE pin electronics receiver. If the test engineer wants to measure the high level of a digital signal by moving the receiver level threshold up until he gets a fail, the associated error to the measured value will be ± 10 mV based on the specification of Figure 5.76.

DC Performance

Threshold range	-2 to 7 V ¹⁾
Threshold resolution	2.5 mV
Threshold accuracy	± 10 mV ²⁾
Minimum overdrive	50 mV
Input leakage current	± 10 μ A (Characteristics: ± 2 μ A between 0 V to 5 V level)

1. Minimum high-low threshold difference: 100 mV.
2. Valid for threshold voltage range -2 V to 6 V and a DC Update period of 2 weeks.
Characteristics: ± 15 mV within a DC update period of 1 month

Figure 5.76 Specifications of the Verigy V93000 P1000 ATE pin electronics receiver (courtesy of Verigy).

Random errors are another category. In this case, the test engineer can address this type of error by making several measurements of the same variable and averaging the samples to obtain a more accurate result. This is a very important concept since it explains how the number of samples acquired has a direct impact on the accuracy of the measurement.

Mathematically if we intend to measure a variable X which is contaminated by Gaussian random noise (e.g., measuring frequency on a clock with random jitter or the BER of a DUT driver) with a variance σ by taking N samples of X , the obtained average value \bar{X} will have a variance of σ/\sqrt{N} [42, 43]. This means that with an increase in the number of measured samples of the variable X , the average value \bar{X} gets closer to the real value of X .

References

- [1] H. Shiotsuka and N. Masuda, "Measurement Technology for 6.5 Gbps High-Speed Serial Interface Using T2000," *Probo*, no. 25, 2005.
- [2] H. Werkmann, "X-Mode Synchronization to Free Running Bitstreams," *Verigy Technical Note*, 2006.
- [3] P. Patten, "Divide and Conquer Based Fast Shmoo Algorithms," *Test Conference, International*, pp. 197–202, 2004.
- [4] W. Maichen, *Digital Timing Measurements*. Springer, 2006.
- [5] J. R. Andrews, "Removing Jitter from Picosecond Pulse Measurements," *Picosecond Labs Note AN-23*, 2009.
- [6] H. Okawara, "Novel Eye Pattern Test Method by Waveform Sampler," *Verigy Users Conference (VOICE)*, 2007.
- [7] D. Derickson and M. Mueller, *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization*. Prentice-Hall, 2007.
- [8] M. Mueller, R. Stephens, and R. McHugh, "Total Jitter Measurement at Low Probability Level, Using Optimizer BERT Scan Method," *IEC DesignCon*, 2005.
- [9] B. Brannon, "Sampled Systems and the Effects of Clock Phase Noise Jitter," *Analog Devices AN-756 Application Note*.
- [10] K. Feher, *Telecommunications Measurements, Analysis and Instrumentation*. Noble Publishing, 1996.
- [11] B. Laquai and R. Plitschka, "Testing High Speed Serial IO Interfaces Based on Spectral Jitter Decomposition," *IEC DesignCon 2004*, 2004.
- [12] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [13] G. Haensel, K. Stieglbauer, G. Schulze, and J. Moreira, "Implementation of an Economic Jitter Compliance Test for Multi-Gigabit Device on ATE," *IEEE International Test Conference*, 2004.
- [14] M. P. Li, J. Wilstrup, R. Jessen, and D. Petrich, "A New Method for Jitter Decomposition Through Its Distribution Tail Fitting," *IEEE International Test Conference*, 1999.
- [15] T. J. Yamaguchi, K. Ichiyama, H. X. Hou, and M. Ishida, "A Robust Method for Identifying a Deterministic Jitter Model in a Total Jitter Distribution," *IEEE International Test Conference*, 2009.
- [16] F. Nan, Y. Wang, F. Li, W. Yang, and X. Ma, "A Better Method Than Tail-Fitting Algorithm for Jitter Separation Based on Gaussian Mixture Model," *Journal of Electronic Testing*, Sept. 2009.
- [17] J. Stimple and R. Stephens, "Precision Jitter Transmitter," *IEC DesignCon*, 2005.

- [18] B. A. Ward, K. Tan, and M. L. Guenther, "Apparatus and Method for Spectrum Analysis-Based Serial Data Jitter Measurement," *U.S. Patent 6,832,172 B2*, 2004.
- [19] Y. Cai, S. A. Werner, G. J. Zhang, M. J. Olsen, and R. D. Brink, "Jitter Testing for Multi-Gigabits Backplane SerDes – Techniques to Decompose and Combine Various Types of Jitter," *IEEE International Test Conference*, 2002.
- [20] G. Foster, "Why Do Different Instruments Give Different Jitter Answers," *Synthesis Research SR-TN083 Measurement Brief*, 2009.
- [21] P. R. Trischitta and E. L. Varma, *Jitter in Digital Transmission Systems*. Artech House, 1989.
- [22] T. Yamaguchi, M. Soma, M. Ishida, H. Musha, and L. Malarsie, "A New Method for Testing Jitter Tolerance of SerDes Devices Using Sinusoidal Jitter," *IEEE International Test Conference*, 2002.
- [23] J. Cao, M. Green, A. Momtaz, K. Vakilian, D. Chung, K.-C. Jen, M. Caresosa, X. Wang, W.-G. Tan, Y. Cai, I. Fujimori, and A. Hairapetian, "OC-192 Transmitter and Receiver in Standard 0.18- μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 37, Dec. 2002.
- [24] Bellcore, *SONET OC-192 Transport System Generic Criteria*. GR-1377.
- [25] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, and A. Ivanov, "Jitter Model for the Design and Test of Gbps-Speed Serial Interconnects," *IEEE Design and Test of Computers*, July 2004.
- [26] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.
- [27] Y. Cai, B. Laquai, and K. Luehman, "Jitter Testing for Gigabit Serial Communication Transceivers," *IEEE Design and Test of Computers*, 2002.
- [28] F. Zhang and W. Necoechea, "ISI Jitter Injection Filter Designs Using PIN and Varactor Diodes for SerDes Testing on ATE," *IEEE International Test Conference*, 2006.
- [29] K.-T. Cheng and H.-M. Chang, "Test Strategies for Adaptive Equalizers," *IEEE Custom Integrated Circuits Conference*, 2009.
- [30] B. Laquai, "A Model-Based Test Approach for Testing High Speed PLLs and Phase Regulation Circuitry in SOC Devices," *IEEE International Test Conference*, 2004.
- [31] M. Ishida, T. J. Yamaguchi, and M. Soma, "A Method for Testing Jitter Tolerance of SerDes Receivers Using Random Jitter," *IEC DesignCon*, 2007.
- [32] R. C. Dixon, *Spread Spectrum Systems: With Commercial Applications*. John Wiley and Sons, 1994.
- [33] Serial ATA International Organization, *Serial ATA Revision 3.0*, June 2009. Gold Revision.

- [34] H. Okawara, "SSC Applied Serial ATA Signal Generation and Analysis by Analog Tester Resources," *IEEE International Test Conference*, 2009.
- [35] H. Okawara, "Frequency/Phase Movement Analysis by Orthogonal Demodulation," *IEEE International Test Conference*, 2002.
- [36] T. J. Yamaguchi, M. Soma, M. Ishida, T. Watanabe, and T. Ohmi, "Extraction of Instantaneous and RMS Sinusoidal Jitter Using an Analytic Signal Method," *IEEE Transactions on Circuit and Systems II: Analog and Digital Signal Processing*, vol. 50, June 2003.
- [37] J. Liu, Y. Cai, L. Fang, and R. Ratemo, "Spread Spectrum Clocking Measurement—A Digital Approach," *IEEE IC Test Workshop*, 2004.
- [38] D. M. Pozar, *Microwave Engineering*. John Wiley and Sons, 2005.
- [39] Agilent Technologies, "Agilent Signal Integrity Analysis Series, Part 3: The ABCs of De-Embedding," *Application Note 5989-5765EN*, 2007.
- [40] N. Kularatna, *Digital and Analog Instrumentation: Testing and Measurements*. The Institution of Electrical Engineers, 2003.
- [41] *Calibration Philosophy in Practice, 2nd Edition*. Fluke Corporation, 1994.
- [42] J. R. Taylor, *An Introduction to Error Analysis*. University Science Books, 1997.
- [43] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, 2003.

6

Production Testing

This chapter discusses the production focus testing of high-speed digital I/O interfaces. In contrast to the measurement methodologies described previously, the focus of production testing is not the retrieval of measurement values that allow an exact assessment of the individual performance of a device or design implementation. For production focused test implementations, the main goal is to guarantee the performance of a device according to a given device specification or datasheet while striving for minimum cost of test (COT) for a given test environment. The variable a test engineer typically has under control to contribute to the COT is the execution time per device. The performance of a device beyond the specification, datasheet requirements, or a certain speed bin usually is of minor interest for a production test run. As a consequence, not only the measurement approaches used in production test implementations can differ substantially from the ones we described previously, but also the test philosophy deployed for production tests. Figure 6.1 shows a diagram of the different possible approaches available to test a high-speed digital I/O cell. Note that this list is by no means exhaustive and other authors might use a different classification scheme.

Although classical ATE based testing certainly dominates high volume production, high-speed digital I/O interfaces have contributed to the fact that alternative or complementing test approaches like loopback testing have gained traction in recent years. The next sections discuss the different approaches presented in Figure 6.1 in detail. In the scope of this classification, our understanding of at-speed ATE is that the ATE instruments that are used to test the high-speed I/Os provide pattern and full timing generation at the native speed of the DUT's highest data rate signals. Thus, at-speed loopback instruments that rely on DfT circuitry in the DUT and/or low-speed ATE instruments to provide the required at-speed data generated by the DUT to

their inputs are considered as low-speed ATE instruments, although they need to support at-speed data rates on their loopback signal paths. Another important facet of production test implementations is the topic of multisite testing which will be covered at the end of this chapter.

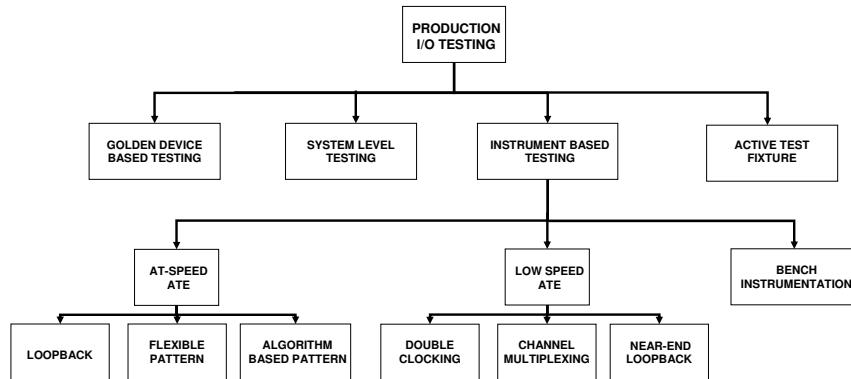


Figure 6.1 Diagram of possible approaches for testing a high-speed digital I/O cell in production testing.

6.1 Golden Device

A quite common alternative for production testing of selected devices is the use of a golden device with a proven good performance mounted on the test fixture. This is a feasible approach for symmetrical applications like a transceiver as shown in Figure 6.2 or when the golden device is the system partner of the DUT.

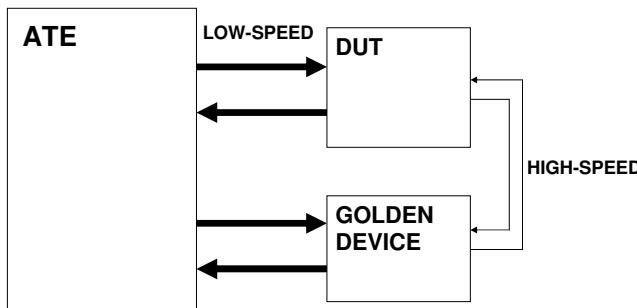


Figure 6.2 Golden device based testing of a transceiver application.

However, this solution has various challenges for the test engineer. Choosing the golden device and verifying its proper function in case of

production issues (e.g., yield drop) might result in a large effort due to the restricted insight into the golden device or the lack of debugging possibilities. Furthermore, variations on multiple golden devices might cause correlation issues between different ATE systems on the production floor.

6.2 System Level Test

Another option is the so-called system level test, where the DUT is tested in its target application (e.g., a microprocessor on a motherboard running special test software). From a cost point of view this approach is very appealing but it can prove difficult to use for testing some physical parameters of the I/O cell. Typically system level tests are used in conjunction with other approaches (e.g., as a second production step after a classical ATE based testing of the DUT). Figure 6.3 shows one example of an automatic system for system level testing. In this case the ATE system tests the DUT using a mission mode environment.



Figure 6.3 Example of an automatic PC based system level tester for GDDR memory applications (courtesy of UniTest).

6.3 Instrument-Based Testing: At-Speed ATE

As shown in Figure 6.1, there are two major options for production testing of I/O cells using an ATE system. One is to use an ATE system that is able to run at the same data rate of the DUT or using an ATE system that runs at a lower speed. The decision of which type of system to use can have a significant impact on the cost of test and also on the failure coverage. This section concentrates on the methodologies that can exclusively be performed by an at-speed ATE system. These methodologies are either the same type of

tests that are done during the characterization phase or substitutes of these with shorter test execution times. A further classification of at-speed ATE systems is possible on the kind of ATE pattern generator, which is either flexible or algorithm based. Finally we distinguish between test solutions with and without far-end loopback. All these aspects are discussed in the next subsections.

6.3.1 Physical Implementation

6.3.1.1 Flexible-Pattern

The flexible-pattern ATE hardware corresponds to the classical at-speed testing approach where digital pin electronics (or sampler for the DUT driver parametric measurements) is used to test the DUT I/O cell by performing measurements using any kind of pattern, within the memory limits of the ATE system as shown in Figure 6.4.

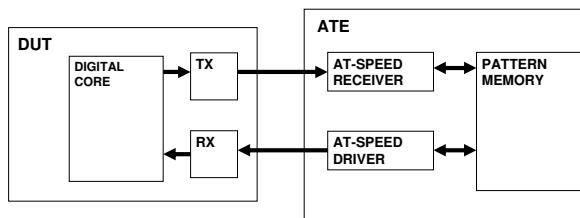


Figure 6.4 Block diagram of an ATE at-speed flexible-pattern approach which relies on the capabilities of the classical at-speed ATE system.

6.3.1.2 Algorithm Based Pattern

The algorithm based pattern approach differs from the flexible-pattern approach in the requirements on the ATE pin electronics. The ability to provide full pattern flexibility at high data rates increases the complexity and cost of the ATE pin electronics. One possible approach to reduce this cost while maintaining the capability to fully test an I/O cell at-speed is to restrict the used pattern to algorithmic patterns that can be generated and compared automatically from hardware (e.g., a PRBS pattern generator/analyzer). If the DUT contains DfT capabilities to create and evaluate the same algorithmic patterns, then it is possible to successfully test the DUT. This approach is shown in Figure 6.5.

In the algorithm based pattern approach, the ATE driver stimulates a data pattern (e.g., a PRBS sequence) to the DUT receiver. A DfT engine contained in the DUT receiver checks the received bit stream for errors. Of course, both

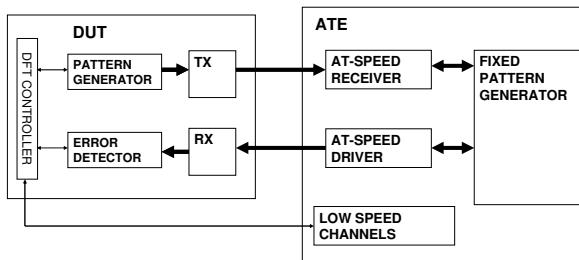


Figure 6.5 Block diagram of an ATE at-speed algorithm based pattern approach.

the sent and received pattern needs to be fully identical bit by bit. On the other side, the DUT driver sends a data sequence generated by the DfT engine to the ATE receiver that in turn checks it for errors. Note that the ATE system must create the compare pattern on-the-fly according to the same algorithms used on the DUT DfT engine.

6.3.1.3 Far-End Loopback

In a far-end loopback testing approach, a loopback between the driver and receiver of the DUT I/O cell is created by DfT paths inside the DUT. Two possible far-end loopback configurations are shown in Figure 6.6. These two configurations are called analog far-end and digital far-end loopback. The important feature of these configurations is that the loopback is generated inside the DUT whereas the high-speed digital stimulus and comparison is performed by the ATE test system. In this case the device's DfT only needs to create the internal loopback paths. The pin electronics of the ATE is responsible for generating the high-speed digital stimulus signal and analyzing the output signal from the DUT driver.

The far-end analog loopback is performed inside or right after the analog I/O cell while the far-end digital loopback is done much further inside the DUT in the core digital logic. The name “analog” is used because the loopback is done inside the high-speed digital I/O cell that contain elements that are considered analog (e.g., PLLs, amplifiers, and so forth) and because of that it can be considered an analog circuit, while the far-digital loopback is performed inside the digital core that is dominated by digital elements and for that reason considered a pure digital circuit. In both cases of the far-end loopback, the device DfT only needs to create the internal loopback paths. The pin electronics of the ATE is responsible for generating the high-speed digital stimulus signal and analyzing the output signal from the DUT driver.

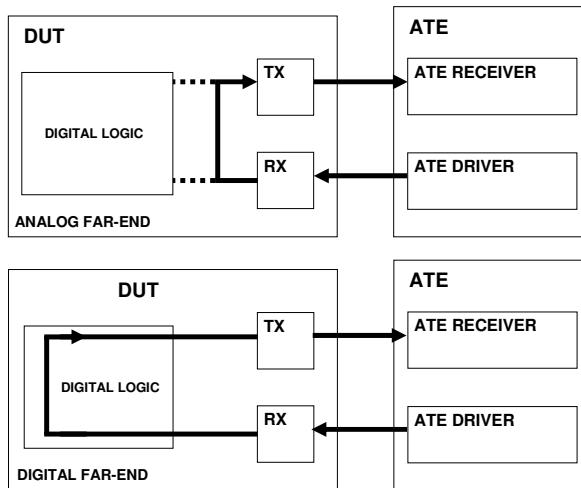


Figure 6.6 Block diagram of the possible far-end loopback approaches: analog far-end (top) and digital far-end loopback (bottom).

6.3.2 Parametric Testing

With an at-speed ATE system all the previously discussed methods for characterization and design verification of an I/O cell can be applied in production with a flexible or algorithm based pattern approach with the DUT configured in a standard mission mode or using a far-end loopback approach. The problem is that in production, test time is critical for the cost of test of each DUT and some of the tests might not be necessary to guarantee the needed failure coverage. Due to this, several test methodologies have been developed specifically for production testing. The next subsections discuss some of these methodologies.

6.3.2.1 Fast Data Eye Mask

The data eye mask test described in Section 5.4.2.5 is an effective way to measure the timing and level performance of a DUT driver and is required by several standards.

In production testing it is common to use a small set of points instead of the full polygon of the data eye mask. This is typically called a fast data eye mask test. The idea is to do a set of functional tests for a selection of the data eye mask points instead of capturing the full data eye and verifying if there are any violations of the data eye mask polygon. This approach is significantly faster than the standard data eye mask test, and with an appropriate choice of the fast data eye mask points, the correlation with the data eye mask test

can be very good. Figure 6.7 compares the fast data eye mask test with the standard data eye mask test. The selection of the number of points and their location is device specific and needs to take into account the expected data eye performance and the critical parts of the data eye to assure the needed failure coverage for the device.

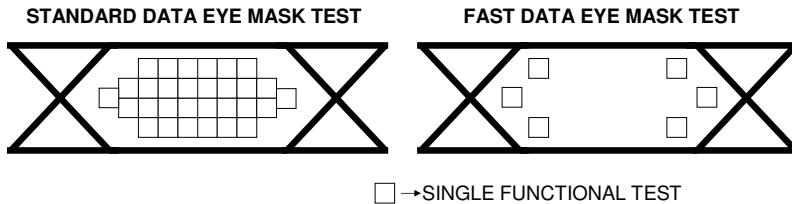


Figure 6.7 Comparison of the standard data eye mask test with the fast data eye mask test.

The drawback of this approach is that it is possible that the data eye mask test is violated at some point but the fast data eye mask test might not detect those violations, since only a reduced number of points are used for the data eye mask test. This risk will depend on the choice of the number of points to be used on the fast eye mask test and their location as shown in Figure 6.8.

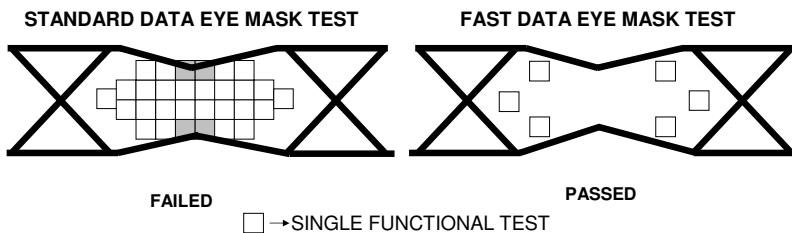


Figure 6.8 Example of how the number and location of the fast data eye mask points affects the fault coverage when compared to a standard data eye mask test.

6.3.2.2 Setup and Hold Time Measurement

Setup and hold time measurement are performed in production for several applications, but unlike the techniques used in Sections 5.4.4 and 5.6.1, in a production test the objective is not to measure the maximum setup and hold values for the DUT but instead ensure that the DUT is able to function with the worst case setup and hold times stated by the DUT specifications. This means that in a production setup and hold test, the timing of the ATE driver or

receiver is set with the worst case setup and hold values and a functional test is performed with these settings.

For high-speed digital interfaces it is important to note that jitter can have a significant impact on setup and hold measurements, meaning that the result of the functional test will depend on the number of compared bits.

6.3.2.3 Transition-Time Measurement

Transition-time measurement as discussed in Section 5.4.1 can be too time consuming in a production test-flow. One approach to reduce the test time when using a digital pin electronics card is to define a priori the transition-time threshold levels VTH (e.g., 20% and 80% of the signal amplitude) and move the compare strobe across the transition region starting first from the right and then from the left and use the pass-to-fail transition-time instants to compute the transition time as shown in Figure 6.9. Like for setup and hold time measurements, jitter can have a significant influence on the result of this measurement especially if all transitions of a pattern are actively compared with the functional tests executed during the timing search. In such a case, it is not guaranteed that the timing results for the two threshold levels originate from the same transition but from two separate transitions within the pattern that are displaced as a whole from the timing grid defined by the signal's data rate. As long as the jitter is small and the device's transition times have enough margin compared to the test limits, this effect can be ignored for a go/no-go test.

In order to keep the jitter that limits the usability of this measurement method to a minimum, a clock pattern should be used for this measurement. This can help to reduce the amount of data dependent jitter (DDJ) that disturbs the measurement significantly. If the jitter of the signal to be measured plus the real transition time of the signal is close to or larger than the test limit for the transition time, it could happen that although all single transitions within the pattern fulfill the transition-time specifications, the test will fail because the jitter spreads the overlayed transitions too far apart from each other. In such a case and in the case that true value measurements are desired, it is required that only a single transition within the pattern is actively compared during the tests executed for the transition-time measurement.

A further test time reduction for go/no-go transition-time tests is achieved by executing the timing search just for one of the two threshold voltage levels. The compare strobe position for the test with the other threshold voltage then is set with an offset of the required test limit plus a potential guardband from the found timing position. With this setting, a single additional functional test after the first search will deliver the required pass/fail

information at the cost of not getting the actual transition-time value but only guarantee that the transition time is below the allowed test limits. If the ATE receiver used to verify the transition time supports nonequidistant compare strobe and dual threshold settings, another go/no-go test implementation is possible. For this implementation the two compare strobes checking for the desired transition are placed with a distance of the transition time that needs to be guaranteed (plus potential guardband). The two threshold levels are set to the high and low levels that are defined for the transition-time measurement (e.g., 20%/80% levels). With such a setting a timing search will yield a result if the required transition time is achieved by the DUT as shown in Figure 6.10.

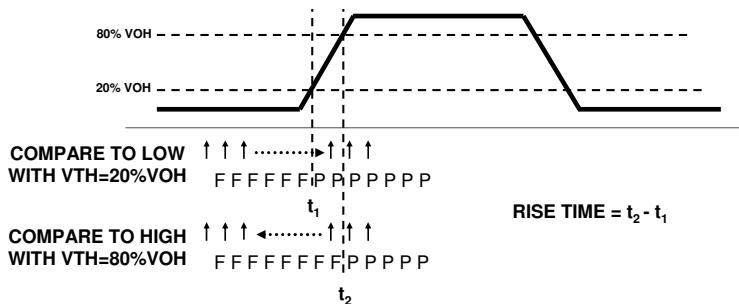


Figure 6.9 Measuring transition time in production with a digital pin electronics card.

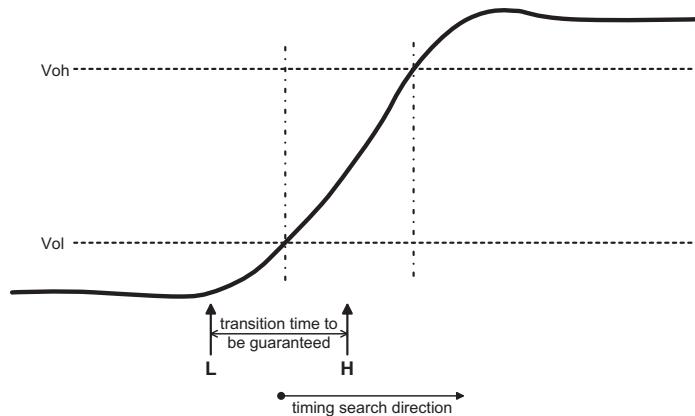


Figure 6.10 Transition-time go/no-go test for dual threshold comparators.

6.4 Instrument-Based Testing: Low-Speed ATE

In this section we will discuss the options available for production testing of an I/O cell using a low-speed ATE system. They are divided into techniques like double data clocking and ATE channel multiplexing that try to perform classical at-speed testing with low-speed ATE system, and near-end loopback testing which uses DfT techniques to enable a self-test of the I/O cell.

6.4.1 Double Data Clocking

Performing a functional at-speed test in production is critical for some high-speed digital applications. This is especially true for devices that are manufactured in very advanced processes with minimum geometries because failure mechanisms in these processes might only exhibit in reduced maximum performance but not in hard functional fails at lower speeds. Double data clocking (DDC) is a methodology to test at full operating speed with reduced I/O data rates on the external signal connections of the DUT. Although DDC predominantly suits DRAM devices, and we focus here on these, it also is usable for other applications. With DDC, lower speed ATE instrumentation can be used to test devices at-speed internally which helps to reduce capital investment for the test solution or extend the lifetime of already available ATE.

Some DRAM devices operate at quad- or even octal data rates (QDR/ODR). For these devices, only the data signals run at the data rate the devices are specified at. The other device signals operate at lower speed at half, a quarter, or an eighth of that data rate. With DDC, instead of writing and reading single bits, the bits on the data signals of the memory device are doubled. Accessing the memory only with bit pairs of the same logical level reduces the effective I/O data rate on the high-speed memory signals by a factor of two. The nondata signals can be kept running at their native speed in a DDC setup. In order to force transitions on any bit boundary, a dual pass test is executed for DDC setups where transitions are present for even bits in a first test run and these transitions then are moved to the odd bits in a second run as shown in Figure 6.11. The result overlay of both of these test executions then corresponds to single bit results.

It is important to note that with DDC only the high-speed data pins of the device are running at reduced speed. This means that all the device internal circuitry like multiplexers, PLLs, and so on operate at-speed. Since adjacent bits on an external data connection are usually not written to memory cells that are geometrically located next to each other, the two bits forming a lower data rate pulse on the external signal are separated in the device and travel on different signal paths inside the device as single bit pulses. Thus, from a

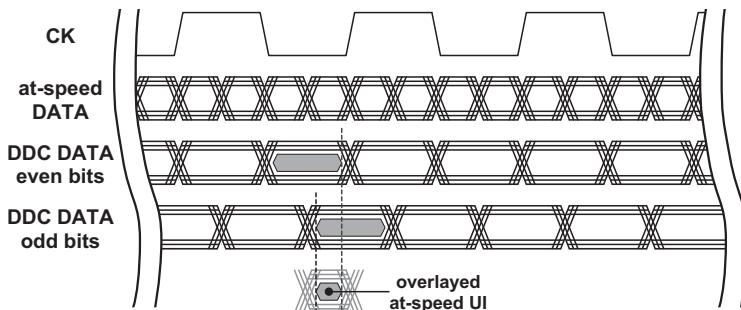


Figure 6.11 The double data clocking approach.

device operation point of view most of the at-speed failures that are detected with an entire at-speed test are also detectable using the DDC approach. Figure 6.12 shows an example of a 4.8-Gbps XDR application comparing the device operation frequency dependent on the power supply voltage for an at-speed capable ATE system and a lower-speed ATE system using DDC. It has to be noted that the ATE systems used for this comparison have significant differences in their timing accuracy. As one can see, the difference between the two shmoo results is visible but not very significant. The device test operation margins are slightly reduced with DDC due to the parametric differences between the lower-speed ATE used for the DDC measurements and the at-speed ATE. However, if the device would not already be tested in a marginal condition on an at-speed ATE, there usually is still enough margin to test it on a lower-speed ATE in DDC mode instead.

One exception from the described DDC concept in the memory area is represented by any kind of DDR device where not only the data signals are operated at the specified data rate but also the clock signal(s). In order to keep the device operation at-speed internally for these devices, the clock signals need to be running at-speed while the data signals still can be operated in DDC manner. Using DDC for this kind of device can still be attractive for production testing because the number of required high-speed ATE channels can be significantly reduced compared to at-speed coverage of all device signals.

When using DDC setups, one of the most critical parameters is the bandwidth of the ATE instruments used to stimulate and check the device. Since these are designed for lower data rates than the ones the DUT natively runs at, they obviously do not provide enough bandwidth for the DDC data rate to be pushed to its extreme (meaning double the data rate the ATE hardware is designed and specified for). The weakest point of a DDC setup

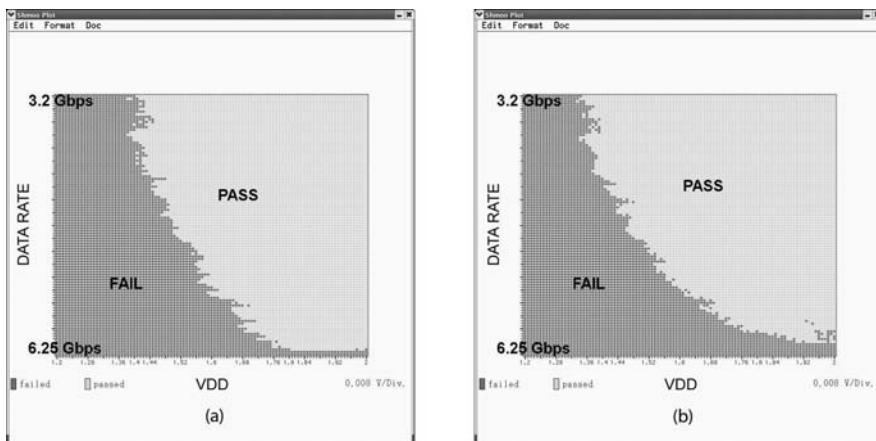


Figure 6.12 Comparison of shmoo results (a) using at-speed functional testing on a high-speed ATE system with (b) a double data clocking approach on a lower-speed ATE-system. (From: [1]. ©2008 Verigy. Reprinted with permission.)

in this regard is the fact that the ATE receiver reliably has to recognize a single at-speed data bit coming from the device. Such a single bit can occur if one of the double bits written to the memory flips its state due to a defect in the memory or parametric deficiencies of the device internal data path. The worst case scenario would be that in a long row of double clocked zero bit pairs one of these zero bits flipped to one (or in a long row of ones a single bit is flipped to zero). The ATE receiver has to be able to recognize this single bit that is sent by the DUT not with the reduced DDC data rate but in nominal speed. Since an ATE receiver used in DDC setups certainly will not be designed to handle that data rate, the signal the receiver sees will only reach a reduced signal level due to the increased transition times caused by the receiver bandwidth limitation. If the signal levels seen by the receiver become marginal with regard to the threshold voltage set for this receiver, a reliable detection of such a single bit failure is no longer possible. Typically, ATE receivers that operate in the required speed ranges are implemented either as differential receivers with an inherent single threshold at the crossing of the positive and negative leg voltages or support a single threshold voltage only. An adaption of this threshold voltage to regain former margins is not possible because a threshold adaption to recognize single bit failures on a sequence of expected ones and also on a sequence of expected zeros is mutually exclusive as shown in Figure 6.13.

Thus, the maximum device data rate that can be achieved with a DDC setup is usually not just double the data rate that is supported by the

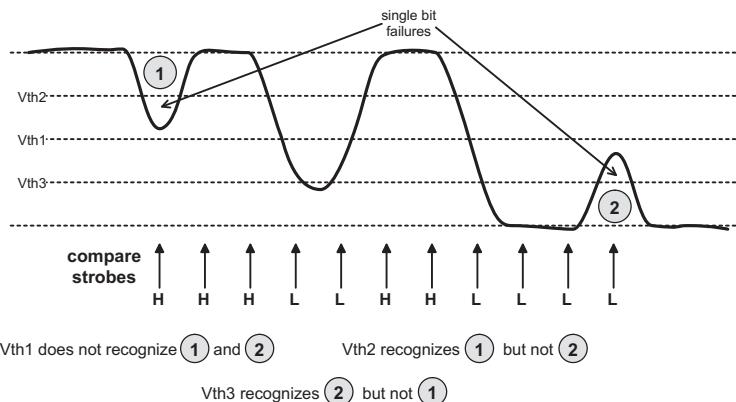


Figure 6.13 Mutual exclusive threshold for DDC single bit failures on bandwidth limited receivers.

ATE instrument used, but strongly depends on the actual bandwidth of that instrument.

6.4.2 Channel Multiplexing

ATE channel multiplexing tries to overcome the fact that the ATE system is not able to test the DUT I/O cell at-speed by multiplexing multiple channels to achieve the needed data rate [2]. This multiplexing can be done on the test fixture using, for example, a XOR logic gate, but this approach can have significant performance challenges for high-speed digital application and requires the use of a large number of channels. Section 9.2 discusses this technique in more detail.

6.4.3 Near-End Loopback Testing

Near-end loopback testing is the preferred approach to implement built-in self-test (BIST) for high-speed digital I/O cells [3, 4]. The idea is to avoid the added cost of using ATE instrumentation to test the I/O cells by connecting the I/O driver to the receiver through an internal or external loop and using the loop to test if the DUT I/O cell is working.

Although the idea is simple, generating an implementation that has the needed fault coverage is not. In fact, there are several varieties of loopback testing with different types of complexity as shown in Figure 6.14. The classification/naming presented on the figure is not standard and other authors might use a different classification/naming for the same loopback methodology. It is important to note that loopback testing might have some

limitations; for example, defect mechanisms that affect both the transmitter and receiver could get masked. Also in some loopback approaches the I/O cell transmitter and receiver work in a synchronous mode while in normal operation they work asynchronously and hence the clock and data recovery logic does not get exercised [5].

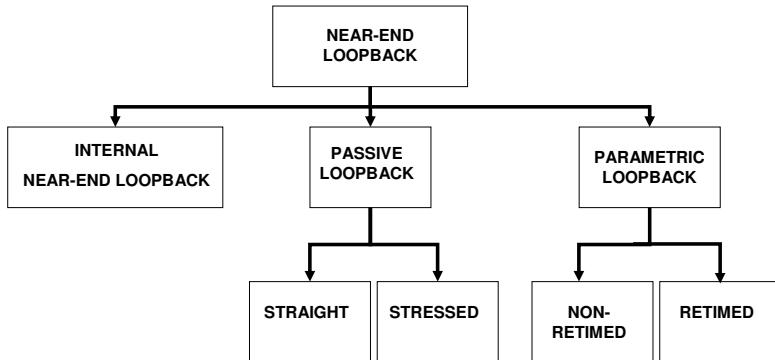


Figure 6.14 Loopback testing approaches with a low-speed ATE system.

Although some engineers see loopback testing as only intended for high-volume production, loopback testing can also provide important characterization data with the appropriate measurement methodology as will be discussed later in this chapter [6].

Loopback testing might also not be appropriate for all types of high-speed digital applications. The ability to create a loopback configuration implies a symmetry between the number of drivers and receivers on the DUT. A high-speed multiplexer cannot be tested using a loopback methodology since it lacks a receiver. One strategy to address this challenge is to add the missing drivers and receivers on a nonsymmetric interface to allow for a loopback testing approach even if they are of no use in the end application (they are only added for DfT purposes) [5]. Although this approach implies adding additional circuitry on the DUT and maybe even more pins on the package for the missing drivers or receivers, in some applications the cost of test advantages from being able to use a loopback approach surpasses the additional costs of implementing these extra drivers or receivers.

DfT Requirements for Loopback Testing

Loopback testing requires that the DUT contains some DfT capabilities implemented in the silicon. This is due to the fact that loopback testing requires the device to operate in a mode that is different from its normal

operation mode (i.e., the device must generate the stimulus signal and also be able to receive this stimulus signal and compare it to the expected value as shown in Figure 6.15).

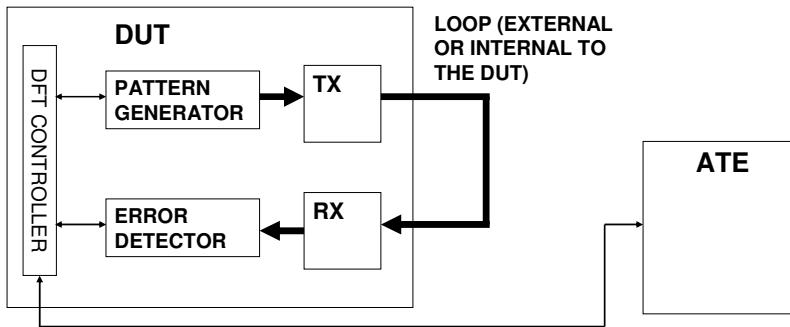


Figure 6.15 Block diagram of basic DfT requirements for loopback testing.

This DfT engine can be very simple or complicated depending on the application and how much test coverage one wants to achieve with the loopback test. Figure 6.16 shows an example of a loopback DfT on a DUT for a wired communications application [7]. Notice that in this example, the DUT is able to use different patterns on the loopback test. A loopback testing approach is not restricted to embedded clock type serial interfaces. Source-synchronous interfaces can also benefit from loopback testing techniques by using the appropriate DfT [8].

More complex DfT techniques for testing high-speed interfaces are available. For example, it is possible to include in the DUT receiver the capability to measure the received eye diagram and even more (see, for example, [9–11]) but this topic goes beyond the scope of this book.

Loop Delay

One important point when thinking about a near-end loopback approach is the loop delay between the transmitter and receiver. The DfT engine in the DUT generates the transmitted pattern on the driver and checks for errors in the received pattern by the receiver. There are some boundaries on the maximum delay the DfT engine can tolerate between the driver and receiver. The DfT engine is usually designed to tolerate a certain amount of maximum delay assuming a straight loopback between the driver and receiver. When one is using a long wired loopback signal path or a parametric loopback approach that might include several active components, it is important to check that the expected loop delay is still acceptable for the I/O cell DfT engine.

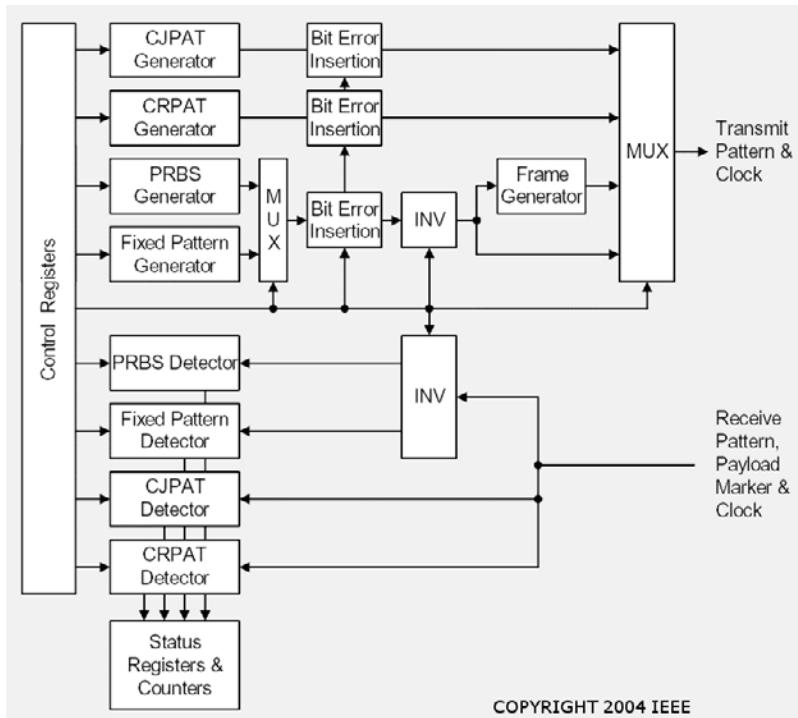


Figure 6.16 Example of a DfT implementation for loopback testing on a DUT (reprinted with permission from [7]).

6.4.3.1 Internal Near-End Loopback

Internal near-end loopback is implemented internally inside the silicon without going out of the DUT die. Typically most DfT implementations to support loopback testing provide different types of internal loopback configurations together with external loopback support. Internal loopback has the advantage that it requires no support of the test fixture in terms of loopback wiring. In near-end loopback two configurations are possible: near-end analog loopback and near-end digital loopback. These configurations are shown in Figure 6.17.

The near-end digital loopback is realized at the digital logic level before the I/O cell, while the near-end analog loopback is done at the I/O cell level. The reasons for the analog/digital naming are the same as explained in Section 6.3.1.3. As shown in Figure 6.17, a DfT controller in the DUT is implemented to start and evaluate the results of the loopback test. It is even possible for some ICs to evaluate the results of the loopback test by functional testing for other I/O pins (typically at lower speed) as shown in Figure 6.18.

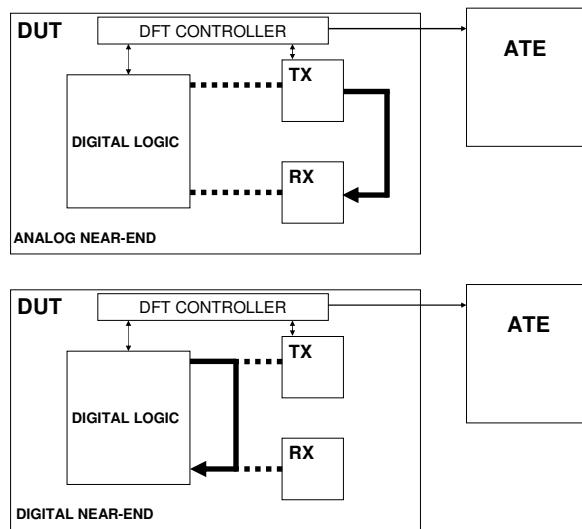


Figure 6.17 Block diagram of possible near-end loopback approaches: analog near-end (top) and digital near-end (bottom).

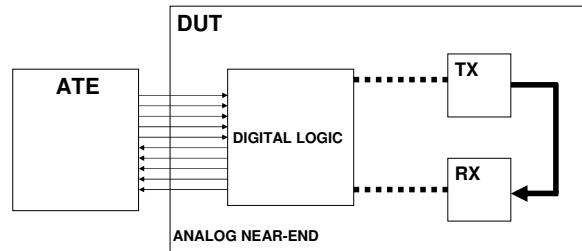


Figure 6.18 Block diagram of an analog near-end loopback approach without a DfT controller to evaluate the loopback results.

6.4.3.2 Passive Straight Loopback

Passive straight loopback (also referred to as wire loopback) is the most simple form of an external near-end loopback testing. As the name indicates, it is simply a wired connection on the test fixture of the DUT driver to its receiver. Figure 6.19 shows two examples of implementing this loop. Typically little attention is given to the loop performance (this is different for a passive stressed loopback as will be discussed later), and usually it is implemented as short as possible. The main idea is that if either the driver or the receiver are not working correctly, the functional test through the loop will show some type of problems (e.g., a nonzero BER). However, modern I/O cells and standards

require compliance to more complex parametric measurements (e.g., jitter). This means that although a wire loopback can catch severe failures in the I/O cell, it might be unable to catch more complex parametric failures. It is important to note that this also depends on the complexity of the used DfT implementation, since it might support more parametric measurements with a loopback methodology (e.g., an integrated level shifter).

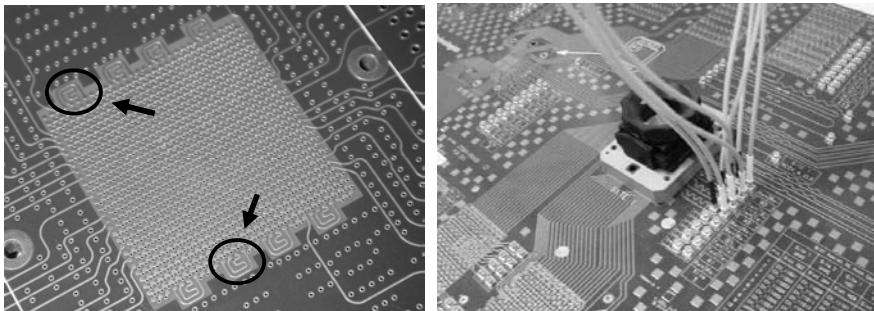


Figure 6.19 Implementation of a wired loopback on the test interface board (left: using a short microstrip loop; right: using a coaxial cable connected to a surface mounted connector).

DC and Scan Testing on Wire Loopback

Although a wire loopback testing approach does not require ATE channels to be connected to the DUT I/O cell for at-speed testing, there is typically the need to perform DC tests on the I/O cell pins (Rx and Tx) and even sometimes use these pins for scan chain access. There are several options to address this need. The most straightforward is to use relays as shown in Figure 6.20. For high pin count devices, however, it might be very difficult to have so many relays on the test fixture.

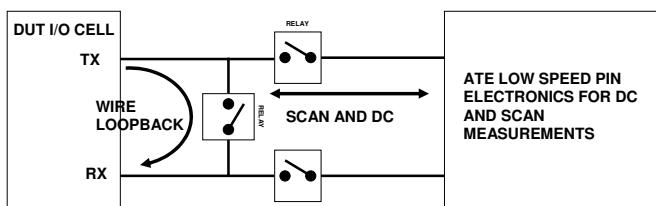


Figure 6.20 Using relays for DC and SCAN measurements on a wire loopback application.

Another option that does not require relays is to use a combination of coils and a blocking capacitor as shown in Figure 6.21. The idea is to create a

path from the pin electronics to the DUT Tx ad Rx pins at low frequencies where the coils behave like a short circuit and the capacitor as an open circuit. At high frequencies the capacitor becomes a short circuit creating the loopback path while the coils behave like an open circuit isolating the loopback path from the pin electronics. Figure 6.22 shows the insertion loss of the TX to RX loopback path through the capacitor and the insertion loss between the RX or TX pin to the DC measurement pin through the coil. A value of 1 pF was used for the capacitor and of 1 nH for the coils.

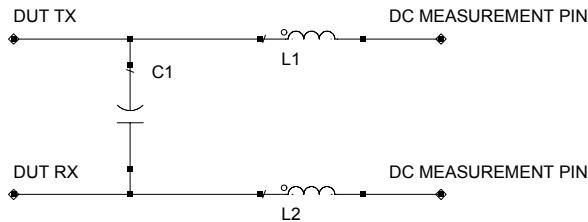


Figure 6.21 Using a coil and a blocking capacitor to provide DC measurement capabilities on a wired loopback application without using relays.

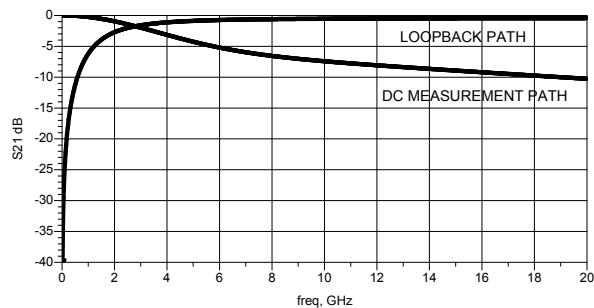


Figure 6.22 Frequency domain response of the TX to RX loopback path and the TX/RX pin to the DC measurement pin for C=1 pF and L=1 nH.

From the results in Figure 6.22, the loss at low frequencies in the RX/TX path to the DC measurement pin is very low due to the LC circuit, which is the precondition for accurate DC measurements and access for scan testing. The TX to RX loopback path insertion loss in this example shows a very lossy performance at low frequencies due to the LC circuit. This does not imply that this approach is not applicable because a stressed loopback that adds a frequency dependent loss might help to improve the test coverage as discussed in the next section. Also, typical patterns for high-speed digital applications have no content at very low frequencies since a minimum transition density is required on the pattern, although there are exceptions (e.g., TMDS). Of

course, these results are based on ideal circuit elements and do not take into account the transmission line behavior of the test fixture traces. This can have a significant impact especially at very high data rates.

6.4.3.3 Passive Stressed Loopback

Passive stressed loopback tries to overcome some of the drawbacks of the passive straight loopback approach by trying to increase the fault coverage without a significant adder on the cost of test. The basic principle is to tune the loopback signal path on the test fixture toward parameters creating more stress on the receiver. The objective of this stress is twofold. On the one hand the margins of the receiver are checked. On the other hand the loopback generated stress is added to the driver characteristics and can cause excess stress to the receiver due to marginal driver performance.

One possible approach is to make the loopback signal path correspond to the worst case signal path the application must support (e.g., some LAN devices' test fixtures are connected to a large roll with several meters of LAN cable for testing the device in loopback). Another option is to add a filter to the loopback signal path (e.g., in [12] a filter that generates data dependent jitter is used on the loopback path). Another option is to implement the loopback signal path using a lossy signal trace on the test fixture. Figure 6.23 shows an example of this approach.

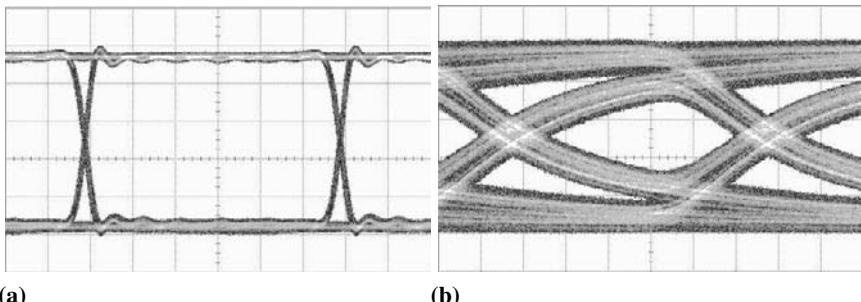


Figure 6.23 Data eye (a) at the DUT driver output and (b) at the DUT receiver after a lossy signal path with a 50.8 cm (20 in) length and a trace width of 127 μm (5 mil) on an FR4 type dielectric.

Since the passive stressed loopback always tests a driver and receiver combination, it is very challenging to design a passive stressed loopback characteristic that addresses the full fault coverage for both driver and receiver on the DUT. The retimed parametric loopback approach that will be discussed in the following section tries to address this challenge.

6.4.3.4 Parametric Loopback

Parametric loopback (also sometimes called active loopback) tries to extend the fault coverage capability of the loopback approach by integrating active components on the loopback signal path [13–15]. The objective is to modify the waveform from the DUT driver in a controllable manner so that it increases the stress for the DUT receiver.

Figure 6.24 shows a block diagram of a simple parametric loopback implementation. In a parametric loopback the DUT output signal is fed into the loop-receiver with a programmable level threshold. Besides some issues, which will be discussed later, this offers the ability to measure the data eye height of the waveform under test. The receiver output is then fed into a jitter injection unit which adds a variable amount of jitter to the test signal. Finally a driver unit with configurable values for amplitude and common mode voltages transfers the modulated waveform to the DUT receiver.

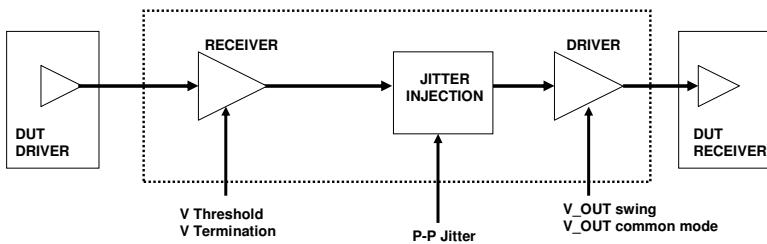


Figure 6.24 Block diagram of a parametric loopback.

This approach has significant advantages compared to the straight or stressed loopback. It is possible with this setup to change the shape of the waveform passing through the parametric loopback in a continuous way. This allows controlled sweeping of parameters to obtain parametric information to assess the DUT I/O performance. Figure 6.25 is a shmoo plot with parametric information for jitter tolerance created with a parametric loopback card of an ATE system. The vertical axis shows the peak-to-peak amplitude of the injected jitter and the horizontal axis is the sinusoidal input jitter frequency. For every pair of these two values the BER measured by the DUT is plotted. As a precondition the DUT (a 10-Gbps FPGA) contains DfT circuitry which can measure the BER of a functional test pattern [16].

The figure clearly demonstrates the capability of a parametric loopback to provide important characterization information on a DUT. This advantage is accompanied by the additional effort and cost for the integration of the loop

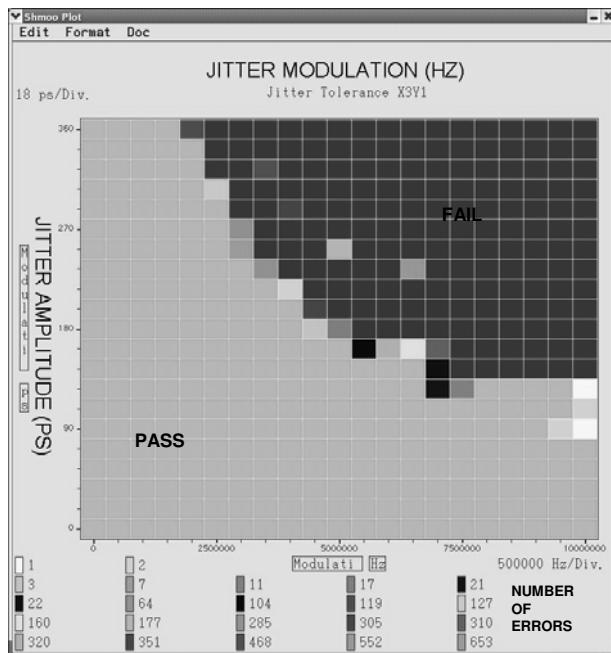


Figure 6.25 Jitter tolerance measurement example with a parametric loopback (courtesy of Verigty).

components on the test fixture or the dedicated ATE module with appropriate control software.

Another challenge with a parametric loopback approach is that some interfaces/standards use out-of-band (OOB) signaling in the interface initialization (e.g., an electrical idle where the positive and negative legs have the same voltage level). A wire loopback approach would allow OOB signaling since the DUT Rx and Tx can understand this type of signaling while a parametric loopback might not be able to handle it properly. The reason is that circuitry to recognize and propagate this special protocol state would be required in the parametric loopback implementation. One solution is to include the capability on the parametric loopback to interpret the out-of-band type signals of the DUT Tx and send the corresponding signals to the DUT Rx. Another possible approach is to include the possibility of a direct wire loopback path on the parametric loopback loop (e.g., through the use of relays) to provide a way to handle this type of signaling during the initialization phases of the interface.

It is possible to increase the capability of the parametric loopback implementation of Figure 6.24. One possibility is to add a measurement unit

to allow for additional parametric measurements of the DUT driver as shown in Figure 6.26.

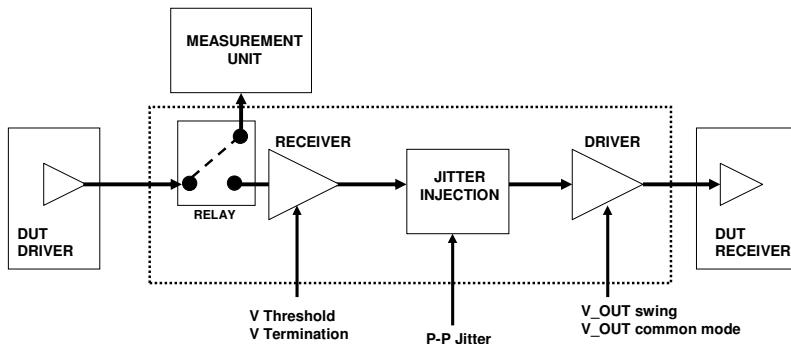


Figure 6.26 Block diagram of a parametric loopback with a measurement unit (e.g., TIA).

Again, this increases the cost of the solution but adds additional capabilities for increasing the test coverage. One possibility for the measurement unit would be a TIA that would allow parametric timing measurements like, for example, rise/fall time or jitter. For reducing the cost of this approach, the measurement unit could be shared between different loops. Further improvement can be achieved by adding a DC measurement unit on both sides of the parametric loop or the integration of a retiming stage as shown in Figure 6.27.

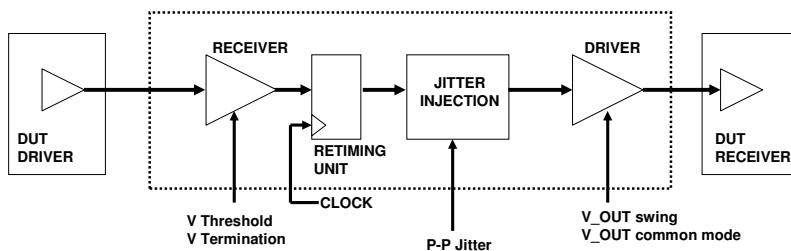


Figure 6.27 Block diagram of a parametric loopback with retiming.

The advantage of a retiming unit in the loop is the ability to precisely control the timing of the waveform fed into the DUT receiver. This is achieved because the retiming unit removes the jitter from the DUT output signal and thus the jitter of the waveform going into the DUT receiver will only be determined by the jitter of the retiming and the jitter injection unit as shown in Figure 6.28.

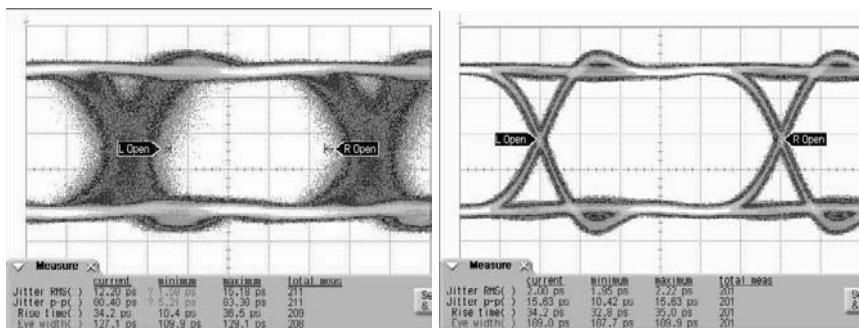


Figure 6.28 Comparison of the DUT Tx data eye after the ATE parametric loopback card but before the input of the DUT Rx (left: no retiming is used; right: retiming is used).

The essential feature of the shown configuration is the independency of the waveform into the DUT receiver from the one originating from the DUT driver. This completely excludes the influence of the DUT driver performance during the receiver test, and therefore provides more accurate results as shown in the two shmoos in Figure 6.29 (i.e., it gets closer to the real performance of the DUT).

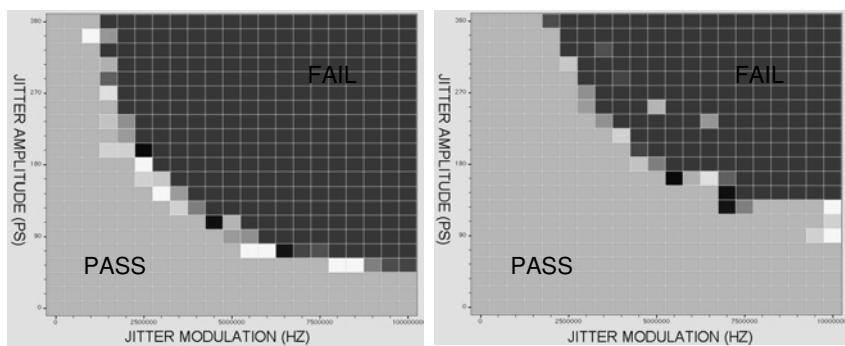


Figure 6.29 Comparison of a receiver jitter tolerance test at 10 Gbps with an ATE parametric loopback card (left: no retiming is used; right: retiming is used).

In Figure 6.29, the jitter tolerance results obtained with a retimed parametric loopback test are more accurate than with a nonretimed loopback test for the exact same I/O cell. The difference corresponds approximately to the jitter from the DUT driver (which might change from cell to cell) that is not removed in the nonretimed loopback approach.

Another advantage of a retimed loop is its higher measurement accuracy for the DUT driver data eye height. Figure 6.30 shows timing diagrams during such a data eye height measurement, where the parametric loopback receiver threshold is varied over a predefined voltage range. When the programmed threshold reaches the driver voltage rails, the nonretimed loop on the left clearly shows a smaller duty cycle than the one using retiming on the right. This makes the task of the receiver under test more difficult than in normal operation mode since it is stimulated by a data stream with significant duty-cycle distortion compared with a retiming approach.

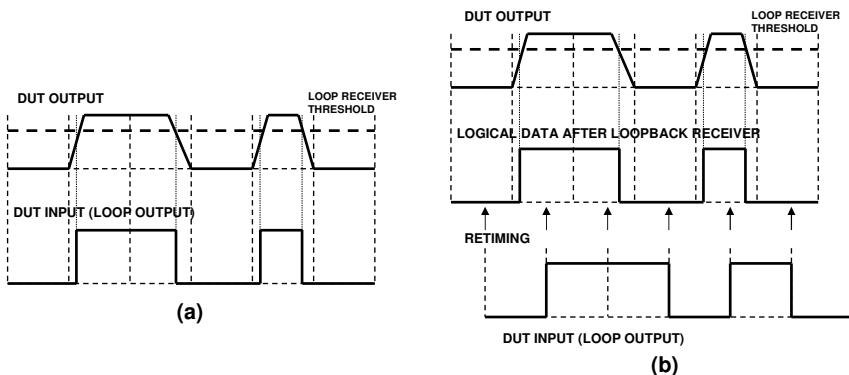


Figure 6.30 The duty-cycle issue (a) with nonretimed loopback versus (b) retimed loopback.

6.5 Instrument-Based Testing: Bench Instrumentation

Another option for production testing of a high-speed digital I/O interface is to use bench type measurement instrumentation on a rack and stack configuration or even integrated with an ATE system as discussed in Section 7.9 [17]. Although such a test setup has significant drawbacks compared to a fully integrated ATE system, it might be an option for devices requiring very high measurement accuracy (typically in low volume applications).

6.6 Active Test Fixture

Another option for production testing is to add instrumentation on the test fixture to test at-speed the high-speed I/O interfaces while using a low cost ATE system. Since this implies adding active components to the test fixture, this approach is sometimes referred to as using an active test fixture or active loadboard. In [18] an example is shown of using this type of approach to

test a 10-Gbps device. In [19] a compact module for testing high-speed I/O interfaces that can be integrated on the test fixture is proposed. ATE channel multiplexing on the test fixture as described in [14] and also in Section 9.2 can also be considered as an active test fixture methodology. Figure 6.31 shows an example of an active test fixture for high-speed digital applications consisting of multiplexing modules using low-speed ATE channels to obtain a high-speed data rate.

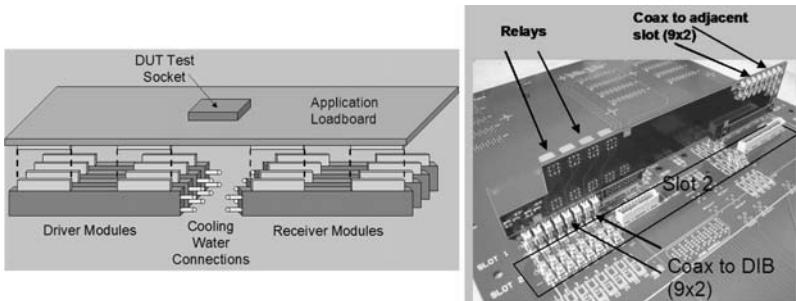


Figure 6.31 Example of an active test fixture approach for high-speed digital testing (reprinted with permission from [20]).

Another possible approach is to leverage the current technology available in modern FPGAs and integrate such an FPGA on the test fixture to test the high-speed I/O interface [21].

Challenges associated with the active test fixture approach are the calibration of the measurement components on the test fixture, and correlation between different test fixtures used for production testing of the same device. The engineering effort needed to develop this type of solution especially in the high-speed digital area can be significant.

6.7 Multisite Testing

One popular concept that helps to reduce the cost of production testing is multisite testing [22]. In a multisite setup multiple devices are tested simultaneously on a single ATE system. The number of sites ranges from dual-site configurations for high pin count devices like a microprocessor up to 256 sites and higher, quite common in memory testing. One key metric to describe such configurations is the multisite efficiency, described by the following equation [23]:

$$\text{Multisite Efficiency} = \left(1 - \frac{T_N - T_1}{(N - 1) \times T_1} \right) \times 100\% \quad (6.1)$$

where T_N is the test-time of an N -site configuration and T_1 is the test time of a single site configuration. When performing a production test using N sites instead of a single site we would ideally expect that the test time of the N site configuration is the same as the test time of a single site resulting in a value of 100% for the multisite efficiency in (6.1). If the test time of an N -site configuration is N times the test time of a single site then the multisite efficiency would be 0%. In reality, multisite efficiency will be below 100%. One reason is that some tasks, like downloading data to the workstation for post-processing, will take more time with an increased number of sites. Also, in high-speed digital applications some tests might require the use of shared resources between sites like a high-speed measurement instrument.

6.7.1 Driver Sharing for Multisite Applications

For production testing especially in multisite applications like high-speed memory, cost of test (COT) is very critical. One way to reduce COT is to have a large number of sites. In some situations it is advantageous to use a single channel to stimulate multiple DUTs and reduce the number of required ATE channels [24].

This approach is called driver sharing and is usually applied to the unidirectional input pins of the DUT (e.g., the address or control lines on a DDR application). There are several different topologies that can be used to implement a driver sharing methodology. The most common approaches in this area are:

- Y-sharing;
- Daisy-chain sharing;
- Active sharing.

Figure 6.32 presents a block diagram of a dual-site Y-sharing, a quad-site daisy-chain sharing, and a hybrid sharing approach. The daisy-chain configuration is very similar to a tapped bus topology that is used in memory applications (e.g., DDR). Figure 6.33 shows an example of an active driver sharing architecture where extra drivers are used on the test fixture to support the sharing of the ATE driver between different DUTs.

It is important to note that several variations are possible in the driver sharing architectures presented in Figures 6.32 and 6.33 [25]. The challenge with driver sharing is to achieve the required signal performance at each DUT. Therefore the signal has to be equal at each of the served DUTs even when one or multiple DUTs are deactivated from testing. To better understand this challenge, Figure 6.34 presents the results of a daisy-chain driver sharing architecture simulation with four DUTs. Note that each DUT input is modeled

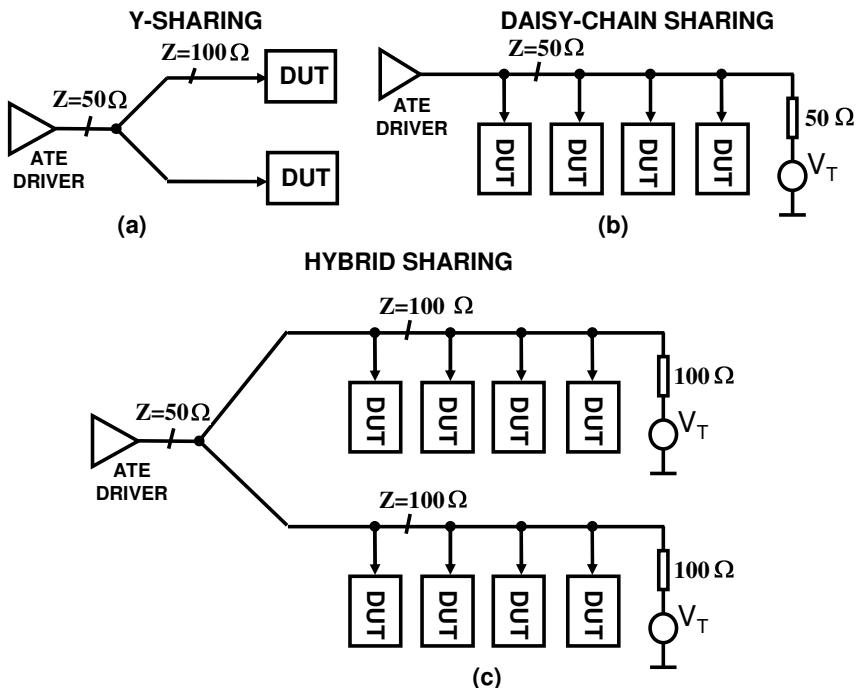


Figure 6.32 Block diagram of (a) two-DUT Y-sharing topology, (b) a four-DUT daisy-chain topology, and (c) a hybrid sharing topology mixing Y-sharing with daisy-chain sharing to obtain an eight-DUT sharing topology.

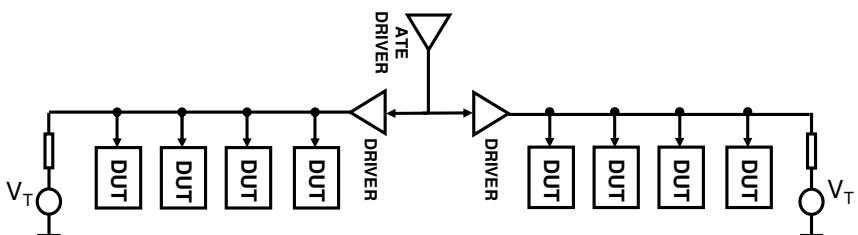


Figure 6.33 Block diagram of an active driver sharing topology.

as a high impedance input with a parasitic capacitance. The timing delay caused by unequal run times to each DUT can be observed, but more critical is that the signal degradation at each DUT is different due to the presence of the multiple taps of the daisy-chain configuration that create reflections that are dependent of the specific DUT position on the daisy chain.

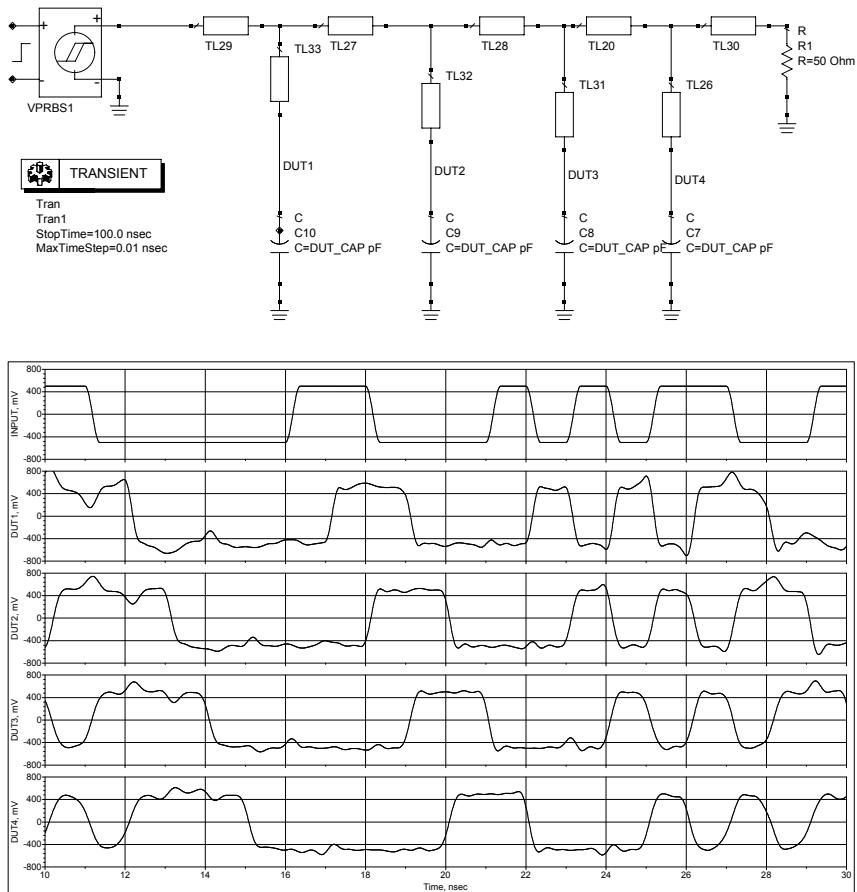


Figure 6.34 Simulation of a passive daisy-chain architecture for driver sharing with four DUTs.

For high-speed digital applications, driver sharing presents significant signal integrity challenges that require a very careful design of the sharing circuitry and signal traces.

References

- [1] K. Kato and T. Sonoda, "Mass Production Test Solution for XDR 4.8Gbps on the Leading Edge of DRAM," *Verigy Users Group Conference (VOICE)*, 2008.
- [2] D. C. Keezer, D. Minier, and M.-C. Caron, "Multiplexing ATE Channels for Production Testing at 2.5 Gbps," *IEEE Design and Test of Computers*, 2004.
- [3] A. Meixner, A. Kakizawa, B. Provost, and S. Bedwani, "External Loopback Testing Experiences with High Speed Serial Interfaces," *IEEE International Test Conference*, 2008.
- [4] T. Pham, B. Koehler, D. Young, and L. Bushard, "Test Structure and Testing of the Microsoft XBOX360 Processor High Speed Front Side Bus," *IEEE International Test Conference*, 2006.
- [5] I. Robertson, G. Hetherington, T. Leslie, I. Parulkar, and R. Lesnikoski, "Testing High-Speed, Large Scale Implementation of SerDes I/Os on Chips Used in Throughput Computing Systems," *IEEE International Test Conference*, 2005.
- [6] J. Moreira, J.-W. Mohr, and R. Nettles, "At-Speed Loopback Testing: Strategies and Techniques," *2nd IEEE International Workshop on Automated Test Equipment (ATE VISION 2020)*, 2008.
- [7] G. Haensel, K. Stieglbauer, G. Schulze, and J. Moreira, "Implementation of an Economic Jitter Compliance Test for Multi-Gigabit Device on ATE," *IEEE International Test Conference*, 2004.
- [8] T. M. Mak, M. Tripp, and A. Meixner, "Testing Gbps Interfaces without a Gigahertz Tester," *IEEE Design & Test of Computers*, vol. 21, no. 4, 2004.
- [9] K. Krishna, "A Built-In Mixed Signal Test Architecture for a PCIe Phy in 90nm and 130nm CMOS," *PCI-SIG 2006*, 2006.
- [10] B. Casper, A. Marti, J. E. Jaussi, J. Kennedy, and R. Mooney, "An 8-Gb/s Simultaneous Bidirectional Link with On-Die Waveform Capture," *IEEE Journal of Solid-State Circuits*, vol. 38, Dec. 2003.
- [11] S. Sunter and A. Roy, "Structural Tests for Jitter Tolerance in SerDes Receivers," *IEEE International Test Conference*, 2005.
- [12] Y. Cai, B. Laquai, and K. Luehman, "Jitter Testing for Gigabit Serial Communication Transceivers," *IEEE Design and Test of Computers*, 2002.
- [13] B. Laquai and U. Schoettmer, "A Low-Cost Vectorless ATE-Channel Architecture for Testing High-Speed IO Signal Integrity in High Volume Manufacturing," *IEEE ITC Future of ATE Workshop*, 2003.
- [14] D. Keezer, D. Minier, P. Ducharme, and A. Majid, "An Electronic Module for 12.8 Gbps Multiplexing and Loopback Test," *IEEE International Test Conference*, 2008.

-
- [15] W. A. Fritzsche and A. E. Haque, “Low Cost Testing of Multi-GBit Device Pins with ATE Assisted Loopback Instrument,” *IEEE International Test Conference*, 2008.
 - [16] J. Moreira, B. Sadler, and D. Ferguson, “Parametric Testing of a 10Gbs I/O Cell in Production Through a Parametric Loopback Approach with Retiming,” *IEEE International Design and Test Workshop*, 2006.
 - [17] J. Moreira, G. Haensel, and F. Koban, “Addressing the Challenges of Implementing an At-Speed Production Test-Cell for 10Gb/s Wafer Probing,” *DesignCon*, 2005.
 - [18] S. C. Ong, “Enabling Test-Time Optimized Pseudorandom Bit Stream (PRBS) 2E31 BER Testing on Automated Test Equipment for 10Gbps Device,” *3rd IEEE International Workshop on Electronic Design, Test and Applications*, 2006.
 - [19] M. Hafed, D. Watkins, C. Tam, and B. Pishdad, “Massively Parallel Validation of High-Speed Serial Interfaces Using Compact Instrument Modules,” *IEEE International Test Conference*, 2006.
 - [20] D. Keezer, D. Minier, P. Ducharme, D. Viens, G. Flynn, and J. McKillop, “Multi-GHz Loopback Testing Using MEMS Switches and SiGe Logic,” *IEEE International Test Conference*, 2007.
 - [21] S. Sunter and A. Roy, “A Self-Testing BOST for High-Frequency PLLs, DLLs, and SerDes,” *IEEE International Test Conference*, 2007.
 - [22] J. Rivoir, “Parallel Test Reduces Cost of Test More Effectively Than Just a Cheap Tester,” *IEEE/SEMI International Manufacturing Technology Symposium*, 2004.
 - [23] J. Kelly, “Multi-Site Efficiency and Throughput,” *Verigy Technical Note*, 2008.
 - [24] S. West, “Redefining Memory Test,” *Evaluation Engineering*, Jan. 2009.
 - [25] Y. Jeon, S.-I. Kwon, C. Mack, and G. Westendorf, “Parallel Logic Test Interface Solutions,” *Burn-In and Test Socket Workshop*, 2009.

7

Support Instrumentation

This chapter presents a discussion on bench measurement instrumentation and its relation to high-speed digital measurements with ATE. References [1–4] provide more information on the topic of measurement instrumentation. Also, the Web sites of the different measurement instruments manufacturers are good sources of information.

The measurement instruments presented in this chapter represent a set of tools that ideally every engineer working on high-speed digital testing should have access to. It is important to understand the capabilities of each instrument and how they can be used in a high-speed digital application.

7.1 Oscilloscopes

7.1.1 Real-Time Oscilloscopes

The real-time oscilloscope is probably the bench instrument most engineers will feel comfortable with, since it resembles the analog oscilloscopes most electrical engineers encounter as their first bench instrument. Figure 7.1 shows a picture of a real-time oscilloscope and Figure 7.2 a high-level block diagram.

The challenge for the real-time oscilloscope is the acquisition of both time and level information in real-time using a sampling frequency that is higher than the data rate of the digital signal under measurement. It is obvious that for high-speed digital signals this is not a trivial task, and is one of the reasons for the high cost of state-of-the-art real-time oscilloscopes.

Figure 7.3 shows the data gathering process. Modern real-time oscilloscopes are able to collect and store a very large number of samples in their memory. This allows the user to acquire very long nonrepetitive waveforms and transfer this data to a post-processing software package (e.g., MATLAB)



Figure 7.1 Picture of a real-time oscilloscope. (Copyright Tektronix. Printed with permission. All rights reserved.)

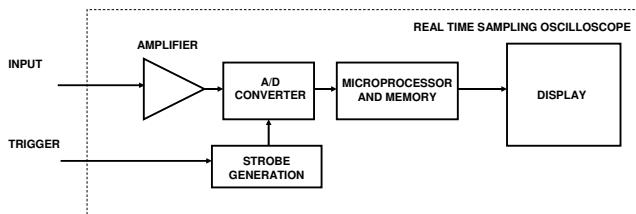


Figure 7.2 High-level block diagram of a real-time oscilloscope.

where the user can process the data using any imaginable algorithm. This feature makes the real-time oscilloscope a very versatile tool. The main challenge with the real-time oscilloscope is that in terms of signal bandwidth, it is typically inferior to other instruments like an equivalent-time sampling oscilloscope. This means that for very accurate parametric measurements of a high-speed digital signal, the real-time oscilloscope might not provide the most accurate results possible. For example, at the time of writing this book, the best real-time oscilloscopes have a specified bandwidth of 20 GHz while equivalent-time sampling oscilloscopes have specified bandwidths of 80 GHz. Clearly, for versatility the real-time oscilloscope is without rival but one must always be aware of its limitations.

7.1.2 Equivalent-Time Sampling Oscilloscopes

The equivalent-time sampling oscilloscope (sometimes also referred to as a digital communication analyzer or DCA) is intended to measure the parametric performance of a digital signal (e.g., jitter, rise time, period) with a very high timing accuracy and bandwidth by using an undersampling

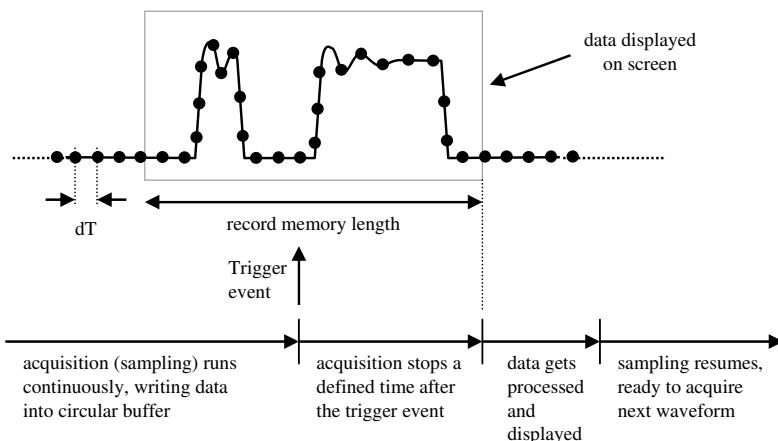


Figure 7.3 Data gathering process of a real-time oscilloscope. (From: [4]. ©2006 Springer. Reprinted with permission.)

approach. Figure 7.4 shows a photograph of a commercially available equivalent-time oscilloscope.

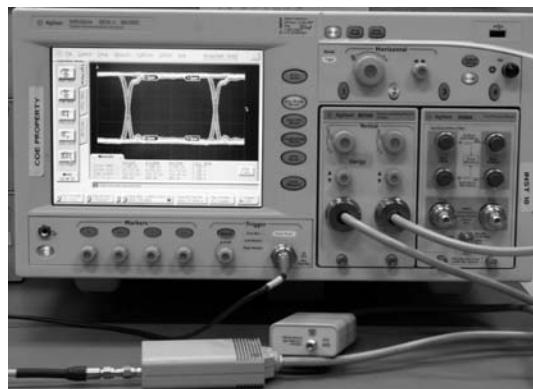


Figure 7.4 Picture of an equivalent-time oscilloscope.

Reference [2] provides an excellent introduction to equivalent-time oscilloscopes and Figure 7.5 shows the typical block diagram of an equivalent-time oscilloscope. The main idea is to use a trigger signal coupled with a high-precision delay line to sample the signal. This means that the signal to be measured needs to be repetitive (i.e., a digital pattern must consist of a repetitive pattern). However, if a data eye diagram is the only measurement to be done, the pattern does not need to be repetitive since we are not interested in acquiring the entire pattern waveform but only measure the data eye diagram.

Figure 7.6 shows a diagram explaining how the equivalent-time sampling approach generates the measured waveform.

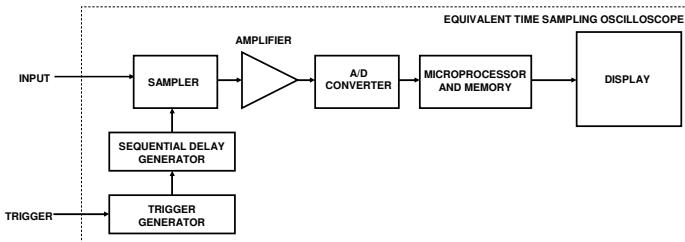


Figure 7.5 Typical block diagram of an equivalent-time sampling oscilloscope.

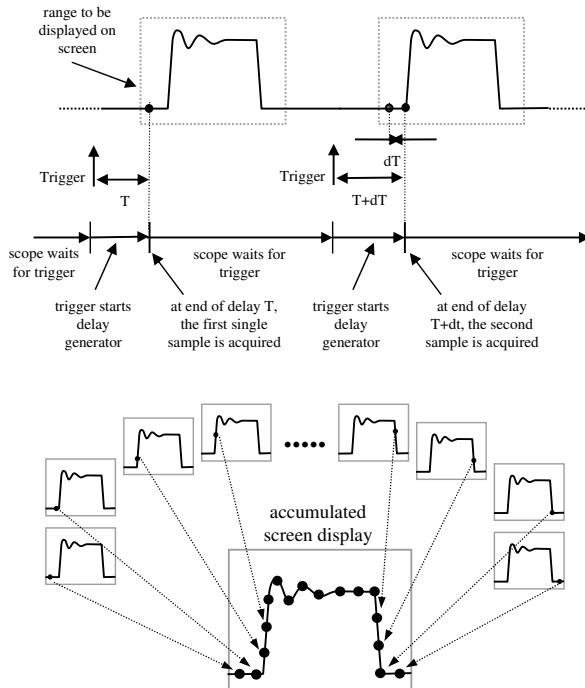


Figure 7.6 Data gathering process of an equivalent-time sampling oscilloscope.
(From: [4]. ©2006 Springer. Reprinted with permission.)

An equivalent-time oscilloscope will display the full waveform of a repetitive digital signal as long as the trigger corresponds to the signal data rate divided by the pattern length in UI. Figure 7.7 shows an example for this type of measurement. Of course, the waveform will correspond to an overlay of several samples of the measured pattern at different points on the

endless measurement loop similar to the waveform reconstruction discussed in Section 4.3.

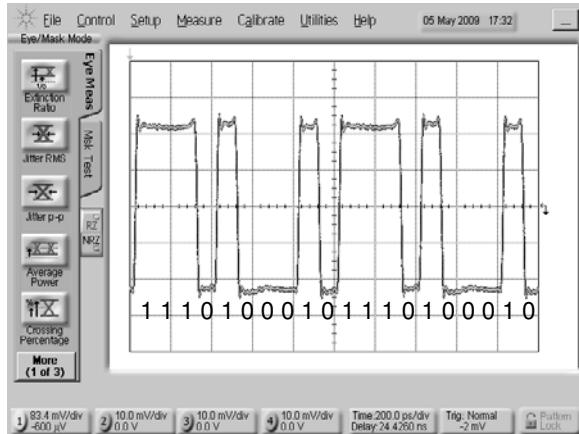


Figure 7.7 Ten-bit waveform (1011101000 repeated endlessly) at 10 Gbps measured by an equivalent-time oscilloscope using a trigger signal with 1 GHz frequency.

The best known feature of the equivalent-time oscilloscope is the data eye diagram as shown in Figure 7.8. On a data eye all the bits in the pattern are merged into an overlay view (see Section 2.1.2). In this case there is no need for a repetitive pattern, only for a trigger that is synchronous to the bits in the pattern. Such a trigger signal could even be obtained from a clock and data recovery (CDR) module. Of course, the data eye diagram does not provide any information on the measured pattern bit sequence but does provide information on the parametric performance of the signal under measurement like the average rise/fall time, the average signal amplitude, and the timing and level noise (jitter).

The equivalent-time sampling oscilloscope is an excellent tool for analyzing high-speed digital signals due to its accuracy and price (compared to a real-time oscilloscope). However, there are some areas where it cannot substitute for a real-time oscilloscope, especially when debugging nonrepetitive problems or issues that require the waveform to be sampled in real-time. Ideally one would like to have both types of oscilloscopes, real-time and equivalent-time, available.

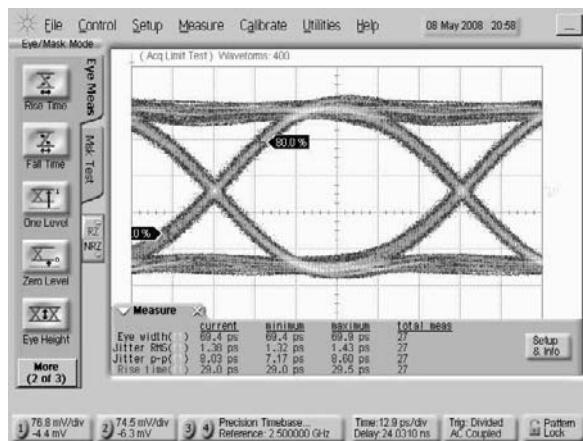


Figure 7.8 Data eye measurement with an equivalent-time oscilloscope for a 12.8-Gbps data signal.

7.2 Bit Error Rate Tester

The bit error rate tester (BERT) is the bench measurement instrument that is most closely related to a standard ATE digital pin electronics. It is able to stimulate and receive digital signals and compare the received signal to an expected pattern to identify bit failures. Current state-of-the-art BERTs are able to achieve data rates above 100 Gbps. Figure 7.9 shows a picture of a BERT instrument.

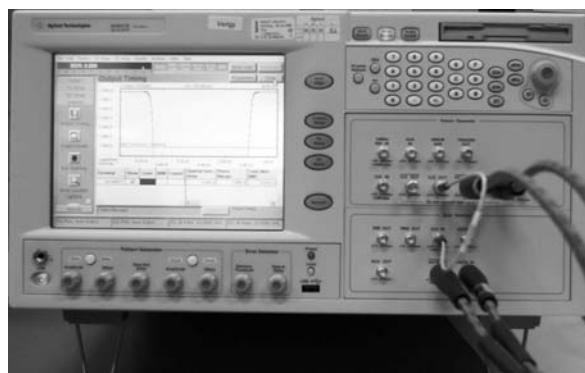


Figure 7.9 Picture of a bit error rate tester (BERT).

Traditionally the BERT has been used to measure the BER of a DUT by running it for several hours. Fortunately, modern BERT instruments have significantly more capabilities [2]. A BERT is able to do data eye

measurements, and jitter measurements including jitter tolerance. The main drawback of the BERT is that it is not as intuitive to use as an oscilloscope for some engineers. However, one should not forget that the BERT makes a pass/fail comparison at each bit generated by the DUT output and normally it is able to store the results of that comparison in memory. This allows the BERT to identify nonrepetitive or sporadic failures. This fact, coupled with the ability to change the compare strobe with a resolution that is a fraction of the DUT output period, makes the BERT a formidable tool.

7.3 Time Interval Analyzer

The time interval analyzer (TIA) is an instrument that is able to measure the time intervals between consecutive events. These events are defined by the signal being measured crossing a certain level threshold. TIAs are not only available as bench instruments but have also been integrated into ATE pin electronics. Figure 7.10 shows a picture of a TIA instrument.



Figure 7.10 Picture of a time interval analyzer (TIA).

TIA measurement instruments can vary significantly in their capabilities, from very simple and low cost implementations intended to measure simple parametric values like frequency to highly sophisticated implementations that are able to characterize a high-speed digital DUT. Figure 7.11 shows a typical high-level block diagram of a TIA.

As stated before, a TIA measures time intervals between events defined as the signal under measurement crossing a certain voltage threshold. One possible differentiator between TIAs is how many thresholds one is able to use at the same time (e.g., having two thresholds allows the measurement of a signal transition time with a single measurement by setting the thresholds to the 20 and 80 percent points of the expected signal amplitude and measuring the time difference between the two events).

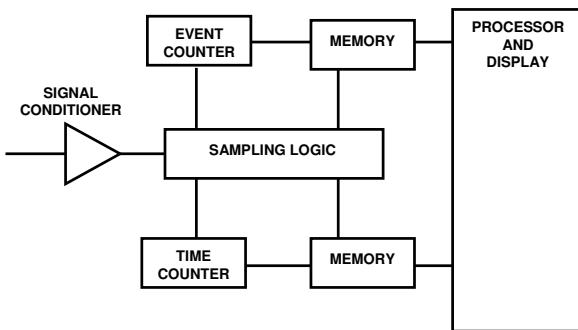


Figure 7.11 Simplified block diagram of a TIA.

Another important point is the number of consecutive events a TIA is able to acquire before it has to prepare for another measurement run (rearming time). This rearming time is typically very large compared to the period of the signal under measurement.

Depending on the capabilities of the TIA instrument it is possible to measure several of the typical parameters of an I/O interface. This is achieved by using some options available in advanced TIAs (voltage threshold, random rearming time, pattern marker, multiple events) and with the appropriate data processing software.

7.4 Spectrum Analyzer

A spectrum analyzer is an instrument that allows the measurement of the spectral (i.e., frequency domain) composition of an electrical signal. Spectrum analyzers are typically divided in two large families (analog and digital) regarding the way the spectral information is measured, although some instruments also use a hybrid approach. Apart from the Web sites of spectrum analyzers manufacturers, [5, 6] provide a good introduction to this instrument. Figure 7.12 shows a picture of a spectrum analyzer.

It is important to note that although most real-time oscilloscopes provide a way to measure the spectral components of a signal by applying an FFT to the acquired signal, their measurement accuracy is inferior to a dedicated spectrum analyzer instrument.

For the test engineer working on the characterization and testing of high-speed digital interfaces, a spectrum analyzer can be used for the analysis of clock signals and the associated jitter. Section I.1 describes the calibration of sinusoidal jitter injection using a spectrum analyzer.



Figure 7.12 Picture of a spectrum analyzer (courtesy of Rohde & Schwarz).

7.5 Vector Network Analyzer

A vector network analyzer (VNA) measures the properties of an electrical network, especially properties associated with the reflection and transmission of electrical signals. A network analyzer uses a controlled sinusoidal generator and measures the amplitude and phase of the transmitted and reflected waveforms at each frequency. In this way it characterizes an electrical network completely. The measured data is typically represented in the form of scattering parameters or S-parameters which are further discussed in Appendix F. Current network analyzers can achieve very high measurement frequencies (e.g., 110 GHz).

Apart from the Web sites of VNA manufacturers, [5, 7] provide a good introduction to this type of instrument. One important characteristics of a VNA instrument apart from its frequency and dynamic range is the number of ports it can measure. The ability to measure multiple ports (e.g., four) allows the user to more easily characterize multiport structures for items like crosstalk. Figure 7.13 shows a picture of a two-port VNA.

In the area of high-speed digital interface characterization, the main applications of a VNA are on the return loss measurement of an I/O cell (Section 5.9.2) and the characterization of the test fixture elements. The fact that a VNA is able to completely characterize an electrical network through the measurement of its S-parameters make it an ideal instrument to obtain measurement based models for simulation (Appendix H).

7.6 Arbitrary Waveform and Function Generators

Arbitrary waveform generators (AWGs) and function generators (AFGs) are instruments that can be used to generate arbitrary types of signals other than the standard digital NRZ type that is the standard in pattern generators and bit error rate testers.

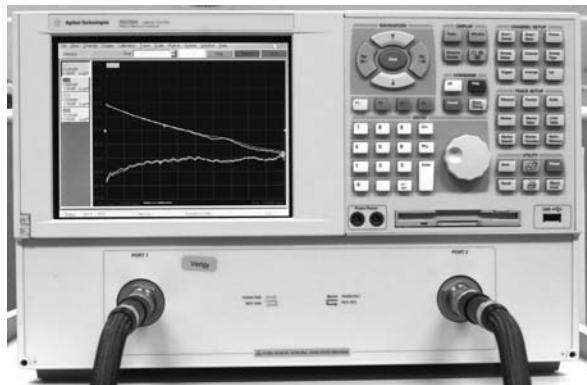


Figure 7.13 Picture of a two-port vector network analyzer (VNA).

An arbitrary function generator provides different waveforms from a predefined set (e.g., sine wave, square wave). The user cannot define a completely generic waveform beyond this predefined set. On the other hand, an AWG is capable of generating a waveform based on a series of voltage points at a given sampling rate. This means that the user can specify any type of waveform. Note that although an AFG seems more limited, for its specific set of waveforms it might have a higher performance than an AWG. Figure 7.14 shows a picture of an AWG.



Figure 7.14 Picture of an arbitrary waveform generator (AWG). (Copyright Tektronix. Printed with permission. All rights reserved.)

Clearly an AWG is a very powerful instrument in the sense that any arbitrary waveform can be generated within the limits of the instrument sampling rate. For digital applications this type of instrument is especially interesting not only for generating modulation signals (e.g., for jitter injection) but also to generate digital waveforms at very high data rates including multi-level signal like PAM-4. References [1, 3] are a good starting point on this topic.

7.7 Noise Generators

Noise generators are instruments designed to generate voltage noise with a Gaussian distribution. Figure 7.15 shows a picture of a commercial noise generator instrument. Typically this type of instrument uses a reverse biased diode to generate the noise distribution with different selectable attenuation values available to set the correct noise level.



Figure 7.15 Picture of a voltage noise source instrument (courtesy of Noisecom).

Characterizing the quality of a noise source is important [8], since in theory a true Gaussian distribution should have an infinite tail on the output voltage levels which would imply the capability to achieve infinite voltage levels (although with a very low probability, i.e., millions of years).

When looking at noise generators, one of the most important parameters is its crest factor. The crest factor of a Gaussian noise voltage source is defined as the ratio of the peak voltage to the RMS voltage value (standard deviation) of the noise generator as shown in Figure 7.16. Sometimes the crest factor is defined as the ratio of the peak-to-peak voltage value to the RMS voltage value. One needs to verify which definition is being used when analyzing the performance of a noise source.

$$\text{CREST FACTOR} = \frac{V_P}{V_{RMS}} \quad (7.1)$$

Some arbitrary waveform and function generators provide the capability to generate a random waveform also. It is important to understand that there can be a significant difference on the generated Gaussian distribution when compared to a noise generator. Figure 7.17 shows a measurement on the time and frequency domain of a pattern generator output that had jitter injected from an AWG type source set for random noise and the same measurements using a dedicated noise source instrument.

The two top graphs on the figure show a comparison of the jitter histogram taken with an equivalent-time oscilloscope (the pattern used was a bit clock pattern). Both histograms appear to have a Gaussian shape but when analyzing the histograms with a post-processing software [9] that compares

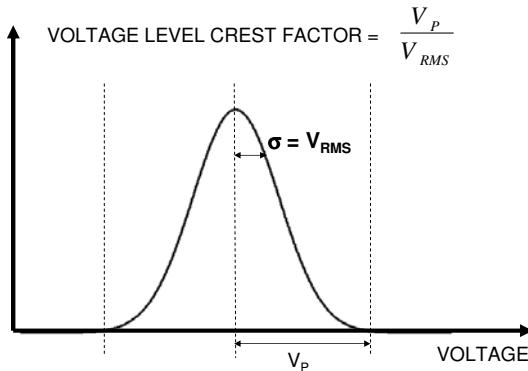


Figure 7.16 Crest factor definition for a voltage noise source.

the measured histogram to the expected Gaussian fitted distribution in a logarithmic scale, one observes a significant difference between the AWG source and the noise source. Another area where this difference is visible is on the frequency domain. In the bottom two graphs of the figure, the measured frequency spectrum is shown for both sources. Ideally one would expect an infinite white noise spectrum, but on the measurements one observes that the AWG source only has spectrum bandwidth to approximately 80 MHz (this was the maximum rated frequency of the used AWG instrument) while the noise generator is able to achieve a noise bandwidth of approximately 2 GHz.

Complementary to the noise generation bench instruments, there are also noise generation modules that provide a fixed noise level as shown in Figure 7.18. These modules, due to their size, are better suited for integration into a test fixture. They also provide a more cost effective way of combining several noise sources and low pass filters to generate more complicated random jitter tolerance profiles that are required by some standards (see Section I.2).

7.8 Sinusoidal Clock Sources

Sinusoidal clock sources, or signal generators, are an important part of any measurement setup since they generate a high-precision, continuous sinusoidal waveform that can be used as a high-precision time reference for triggering and allow the synchronization of several instruments. Figure 7.19 shows a photograph of a commercially available signal generator.

Note that a sinusoidal clock source will not only provide a “sinusoidal clock signal” with a given frequency and low jitter, it might also provide the ability to modulate the output signal (important for low frequency jitter injection) depending on the instrument capabilities. Also of importance is the

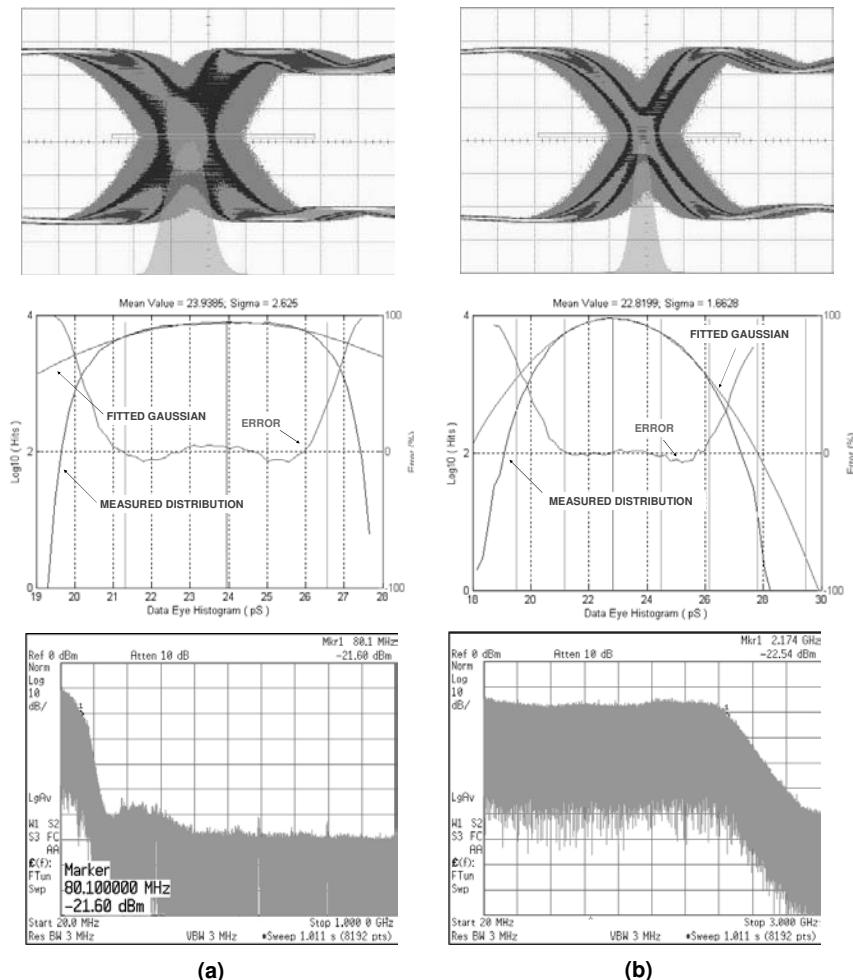


Figure 7.17 Comparison of (a) using the random function of an arbitrary waveform generator for random jitter injection with (b) using a dedicated noise source.



Figure 7.18 Picture of a voltage noise source (courtesy of Noisewave).

10-MHz synchronization input/output. The 10-MHz output is locked to the source PLL and is typically very stable (stability is defined by the parts per million that the 10-MHz signal might drift in time) and can be used to force other instruments to have their internal PLLs synchronized to this 10-MHz output. Section 7.9.2 discusses the synchronization topic in more detail.



Figure 7.19 Picture of a signal generator that can be used as a sinusoidal clock source.

7.9 Connecting Bench Instrumentation to an ATE System

Very often there is the need to connect external instruments to an ATE system. This can be as simple as connecting a voltmeter to an ATE DUT power supply to measure its voltage. Sometimes, however, it means to connect multiple external instruments like a parallel bit error rate tester to help testing a multigigabit I/O DUT on a low-speed ATE system. Figure 7.20 shows two examples of integrating external instruments with an ATE system. The problems that arise from connecting external instrument to an ATE system can be divided in two categories: signal integrity and synchronization. Software integration is also a significant challenge that might require a large effort but it is usually not a bottleneck for integrating external instruments for high-speed digital applications.

7.9.1 Signal Integrity

The first challenge when connecting external instruments to an ATE system, especially when dealing with high-speed digital signals, is to provide a connection between the external instruments and the test fixture (i.e., connectors and cables) with the required performance or signal integrity. The

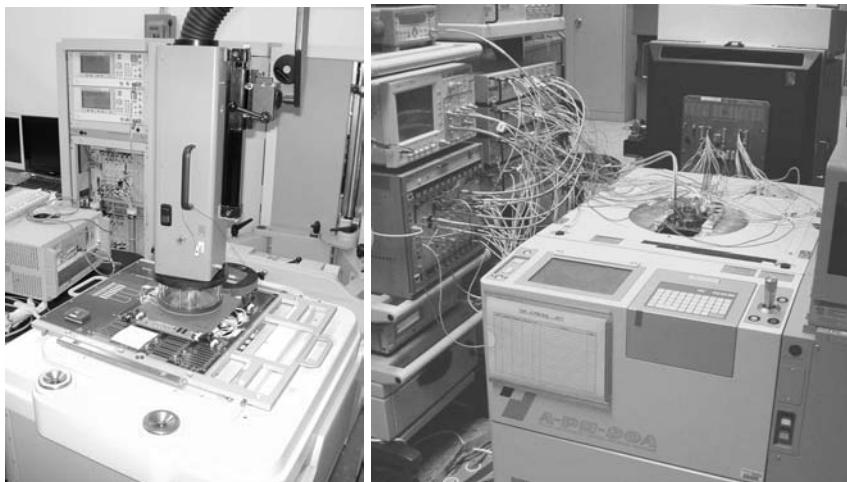


Figure 7.20 Examples of integrating bench instrumentation with ATE systems [10, 11].

challenge is easier if one knows up front that there is a need to connect external instrumentation to the ATE system when developing the test fixture. For example, in Figure 7.21 there is an example of how knowledge of the need to use external instrumentation was taken into account on the test fixture design.

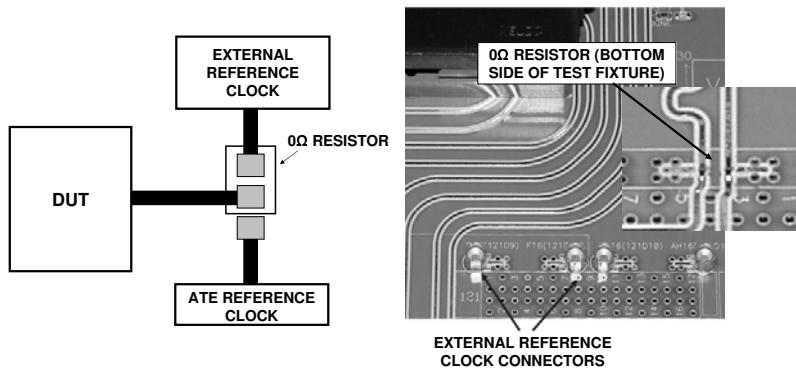


Figure 7.21 Test fixture layout allowing the DUT reference clock to be connected to an ATE pin channel or an external low jitter reference clock by moving a $0\ \Omega$ resistor.

In this application the initial problem was whether the ATE pin electronics could provide a reference clock with low jitter. The question was not on the amount of the pin electronics jitter (its value is known from the

ATE specification), but if a lower jitter clock could improve the production yield even further. Having this knowledge, the test fixture was designed with the possibility to use the ATE pin electronics or an external clock source (connected through an SMP connector) as the reference clock for the DUT. This was accomplished by simply moving a $0\ \Omega$ resistor on the board. In this specific application, the discontinuity generated by the $0\ \Omega$ resistor was negligible.

On other occasions, the need for connecting external equipment arises from the need to debug problems on an application or test fixture that were not foreseen. In this case, one needs to connect the external instrument to the test fixture or DUT socket using an appropriate probe. Appendix H discusses the topic of probing a test fixture in more detail. Note that probing a test fixture is not a trivial task. Not only should the appropriate probing techniques and equipment be used, but also appropriate electrostatic discharge (ESD) precautions should be taken, so as not to damage the ATE system and measurement equipment. Note also when connecting bench instrumentation to an ATE system it is important to guarantee that no ground loops are created and that any extra required shielding is implemented [12, 13].

7.9.2 Synchronization

Another separate challenge is the synchronization of external instruments to an ATE system. Typically when measuring signals from an ATE system with external instrumentation (e.g., an oscilloscope), the user only needs a trigger signal from the ATE system to be provided to the external measurement instrument. This can easily be obtained from an unused ATE channel as shown on Figure 7.22.

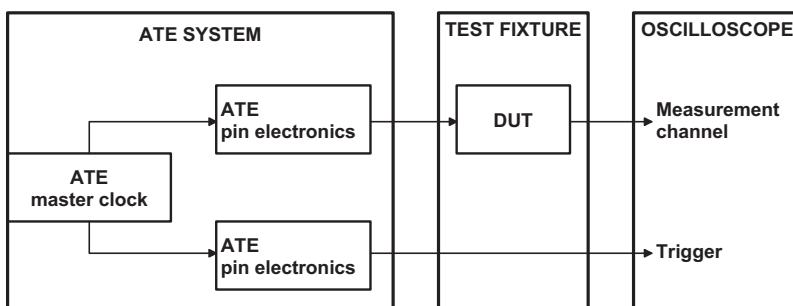


Figure 7.22 Measuring an ATE channel driver signal with a scope using an extra ATE driver channel as trigger.

In this setup the ATE driver channel is used as the reference for all timing measurements. There are some situations where using an extra ATE channel

as the trigger is not good enough, especially when one wants to measure timing performance of the ATE or DUT against a high accuracy external timing reference. Figure 7.23 shows two typical setups for this measurement approach.

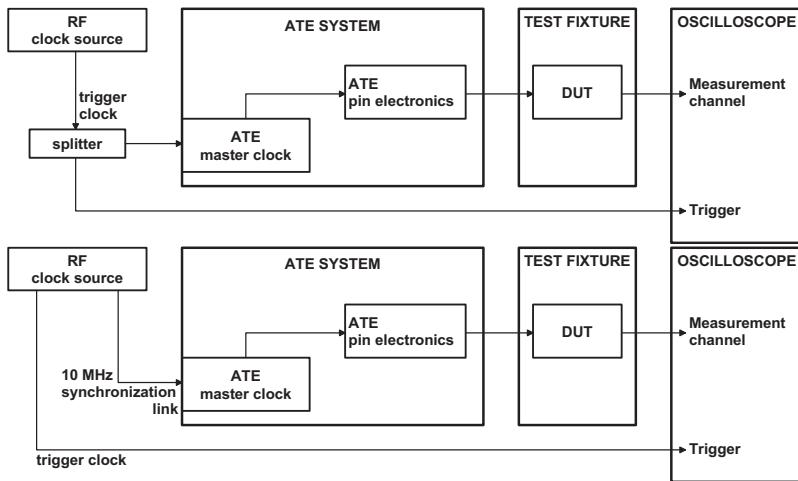


Figure 7.23 Synchronizing an external RF clock source to serve as a master timing reference.

The main challenge is to ensure frequency synchronization between the reference timing source and the ATE system, guaranteeing that there are no frequency drifts or frequency offsets between the two systems. This can be accomplished by using the external timing reference as the master clock for the ATE system or locking the ATE master clock to the external timing reference with a synchronization signal (typically a 10-MHz frequency signal). Notice that these setups depend on the specifics of the ATE hardware architecture and in some ATE architectures this might not even be possible. It is clear that the second setup in Figure 7.23 is the most demanding, since all the timing uncertainties of the ATE system including the master clock are measured against an external time reference.

This type of setup can also be used for another type of synchronization challenge: using an external pattern generator as a signal source. Figure 7.24 shows an example of this type of setup for a demultiplexer DUT. In this case the external pattern generator provides a high-speed digital stimulus signal (e.g., 10 Gbps) to the DUT which in turn demultiplexes this signal into four 2.5-Gbps data lines measured by the ATE pin electronics (assuming the ATE pin electronics maximum data rate in this example is 2.5 Gbps).

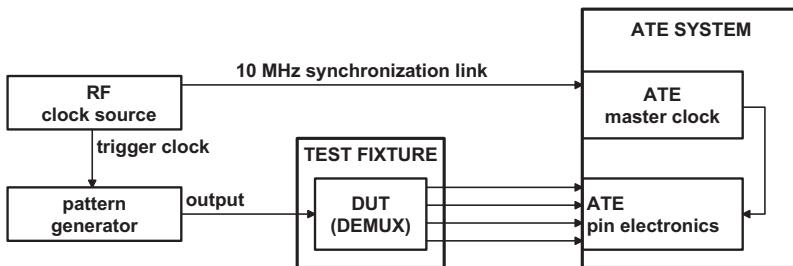


Figure 7.24 Connecting an external pattern generator to an ATE system.

As previously explained, with this setup it is guaranteed that there is no frequency offset or frequency drift between the ATE system and the external pattern generator. However, the fixed phase relationship between external instrument and ATE does not solve another problem, which is to know and to control the exact timing or phase relationship between the ATE system and the external pattern generator and thus the data received by the DUT [14]. The challenges with this pattern synchronization between ATE and external instruments is identical to the bit and pattern alignment discussed in Section 5.1. Thus, also the solutions are the same as presented in that section. The advantage for the synchronization to data provided by external instruments or DUT data derived from external instrument stimuli is the fact that the compare data on the ATE is well decoupled from the DUT stimulus data that is transformed to the DUT data stream to be compared. Thus, solution mechanisms that are based on dynamic pattern offsets between the incoming data stream and the compare data, like the match-loop approach described in Section 5.1.2.1, are easier to implement with external instruments than with DUTs that are completely stimulated by ATE drivers. Another benefit of external instruments stimulating the DUT is the fact that these run on their own clock domain and keep generating data to the DUT even if the ATE sequencers are stopped. For devices that potentially change the phase and/or latency of the data they generate if their stimulus data is stopped and restarted, this simplifies search-based synchronization methods that repeatedly stop and restart the ATE sequencers.

7.9.3 External Reference Clock Impact on Jitter Measurements

It is important to note that using external instruments like a clock source for oscilloscope triggering or to provide the reference clock to the DUT can have an impact on jitter measurement results. When discussing jitter measurements, the key question arises as to which reference the jitter value is measured. Not only has the intrinsic jitter of the reference created a baseline for the

achievable measurement accuracy, even the measurement setup itself might have a significant influence. Figure 7.25 shows two different measurement setups for the jitter performance of an ATE channel. The first setup uses an external reference clock for the oscilloscope triggering that also provides a 10-MHz reference signal to the ATE master clock. The second setup uses an additional ATE channel for triggering the oscilloscope.

If we assume only Gaussian random jitter being present on the setup, the amount of measured random jitter will depend not only on the random jitter present on the signal to be measured but also on the trigger signal.¹ Assuming also that the external reference clock has an intrinsic random jitter that is lower than the random jitter of the ATE channels, then the setup with the external reference clock for triggering provides a more accurate measurement of the ATE channel random jitter. In the presence of other jitter components on the ATE system (e.g., a periodic jitter component on the ATE master clock), when using an ATE channel for oscilloscope triggering these jitter components might not be measured since it is present on both the signal to be measured and the trigger. If the external clock source is used instead, then this periodic jitter component will be measured because it is not present on the trigger signal from the external reference clock source.

7.10 Coaxial Cables and Connectors

7.10.1 Coaxial Cables

Understanding and correctly using coaxial cables is very important for high-speed digital applications. Spending a significant amount of time optimizing the test fixture and then measuring its performance with the first “good looking cable” in the laboratory is not advisable. It is important not only to understand the limitations of the applied coaxial cables, but also how to correctly select one in a given application.

Figure 7.26 presents a diagram of a typical coaxial cable and the equation for its characteristic impedance. Like on a PCB trace, the loss of a coaxial cable is mainly determined by the dielectric type (dielectric loss) and the inner core diameter (skin effect loss). Good quality coaxial cables use low loss dielectrics like Teflon. This means that for typical high quality coaxial cables, the loss is dominated by the skin effect. To reduce the skin effect loss it is necessary to increase the diameter of the cable core which implies an increase on the shield diameter to keep the characteristic

¹Assuming that only Gaussian random jitter is present on the signal to be measured and on the trigger, the measured random jitter is given by: $\sqrt{Jitter_{signal}^2 + Jitter_{trigger}^2}$.

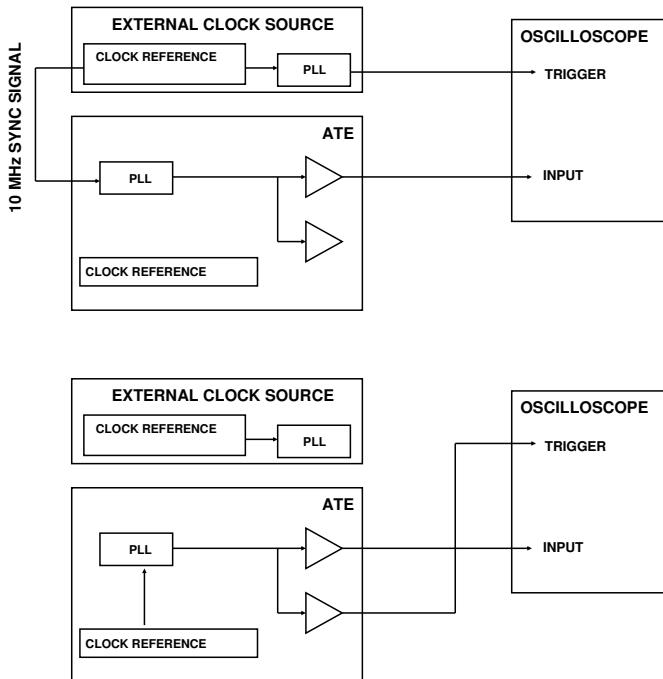


Figure 7.25 Comparison of two different jitter measurement setups.

impedance constant. It is a good rule of thumb to assume that thicker cables provide lower loss for the same length. Figure 7.27 shows a comparison of a data eye diagram measured with two high-quality cables with different lengths and different diameters (i.e., core diameters) and also a comparison of the measured insertion loss. Figure 7.28 shows a picture of the two cables.

Although one would expect the longer cable (80 cm) to have a worse performance than the shorter cable (61 cm), the larger diameter (10.6 mm compared to 6.5 mm) makes it a less lossy cable as shown by the improved rise time (36 ps versus 38 ps) and the measured insertion loss.

One point to note is that there is a drawback of using coaxial cables with large diameters. There is a relationship between the diameter of the cable and the frequency at which the cable starts to exhibit unwanted behaviors in the form of resonances. Figure 7.29 shows a comparison of the insertion loss of two cables with different diameters. One can observe that the larger diameter cable, although having a lower loss, does exhibit a resonant behavior after approximately 19 GHz.

This effect can also be observed in the time domain by using a source with a very fast rise time. Figure 7.30 shows the output of a data source with

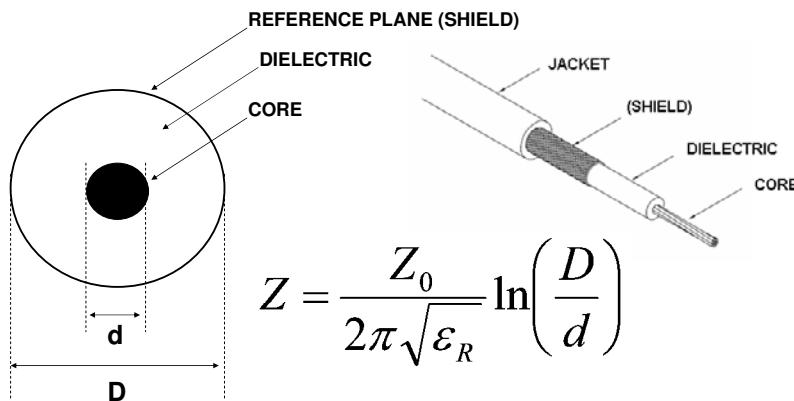


Figure 7.26 Geometry of a typical coaxial cable and its characteristic impedance equation.

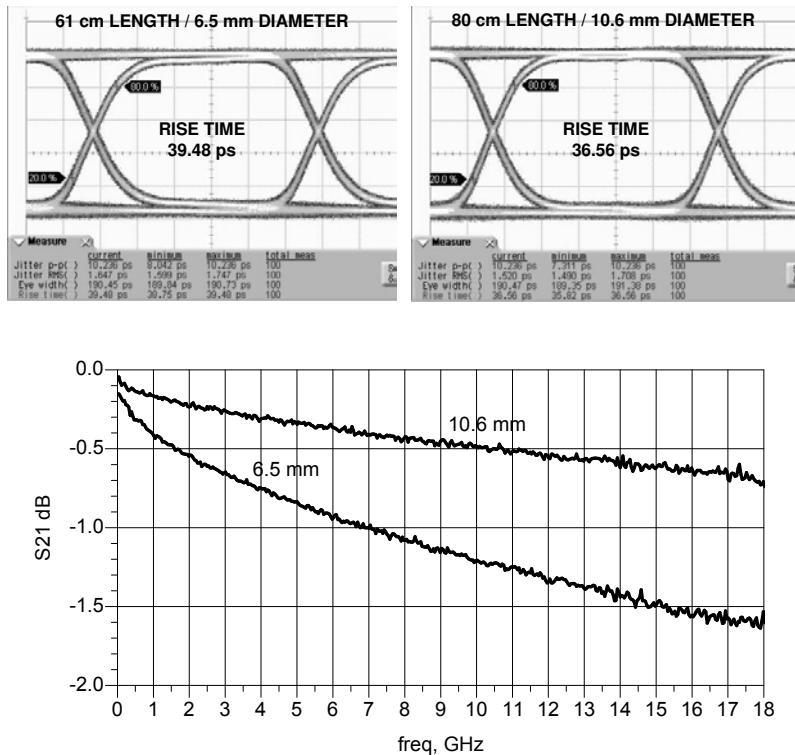


Figure 7.27 Data eye diagram comparison of two coaxial cables (top left: 61 cm length with 6.5 mm diameter; top right: 80 cm length with 10.6 mm diameter) at 5 Gbps with a PRBS $2^{31} - 1$ data pattern (top). Comparison of the insertion loss of the two cables (bottom).

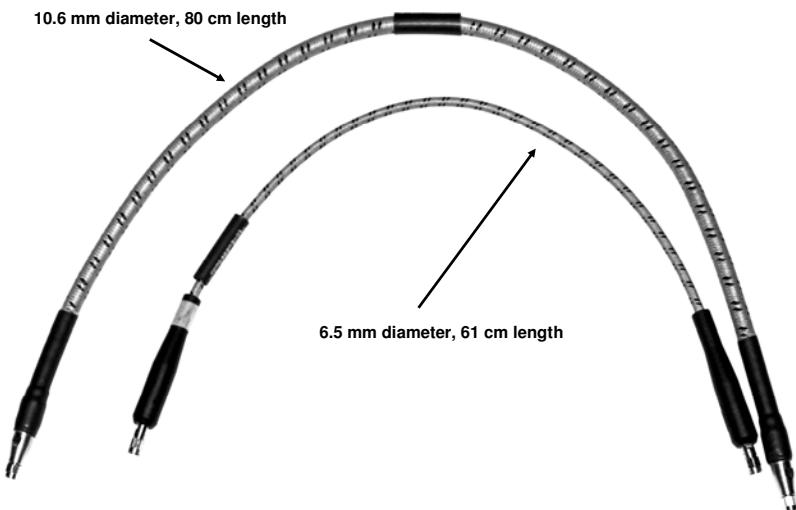


Figure 7.28 Picture of the two coaxial cables with different length and diameter.

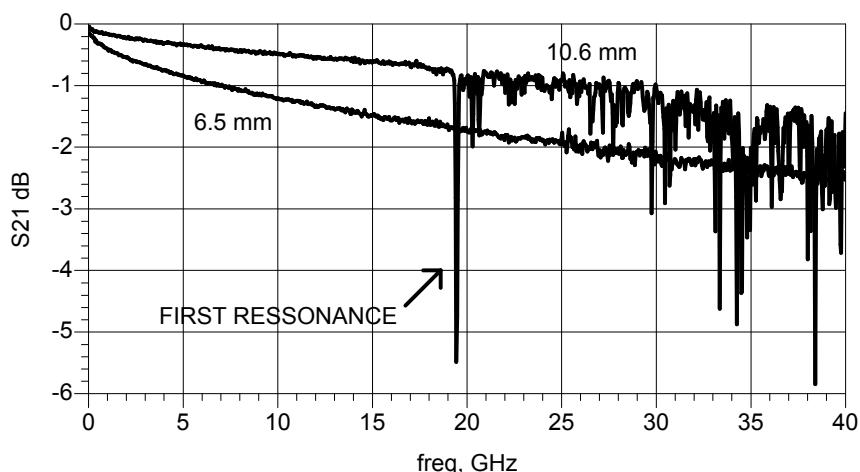


Figure 7.29 Insertion loss comparison of two cables with different core diameters showing the resonant behavior of the large diameter cable beyond 19 GHz.

8-ps rise time and how that waveform looks like after the 80-cm 10.6-mm diameter coaxial cable in Figure 7.28.

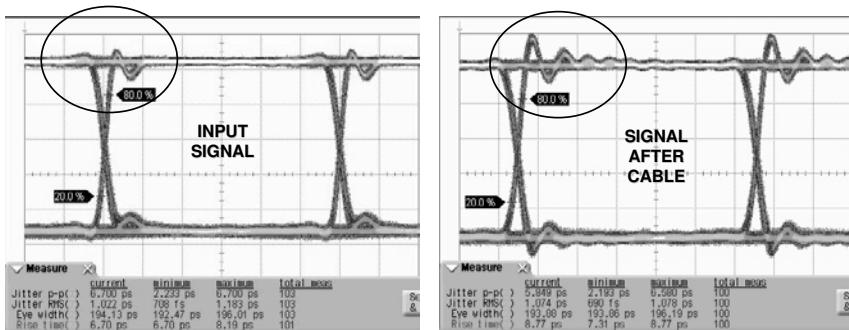


Figure 7.30 5-Gbps digital pattern input signal with 6.7-ps (20/80) rise time (left) and data eye after a 80-cm length, 10.5-mm diameter coaxial cable (right).

The figure shows that the odd behavior of the cable at very high frequencies due to its diameter coupled with a pattern with a very fast rise time results in a waveform with a more pronounced oscillation as shown in Figure 7.30 (right).

It is also important to understand that even with a thin coaxial cable we can always have a lower loss than a PCB signal trace. Figure 7.31 shows this difference by comparing a microstrip in several geometries and different dielectrics to a coaxial cable with different diameters.

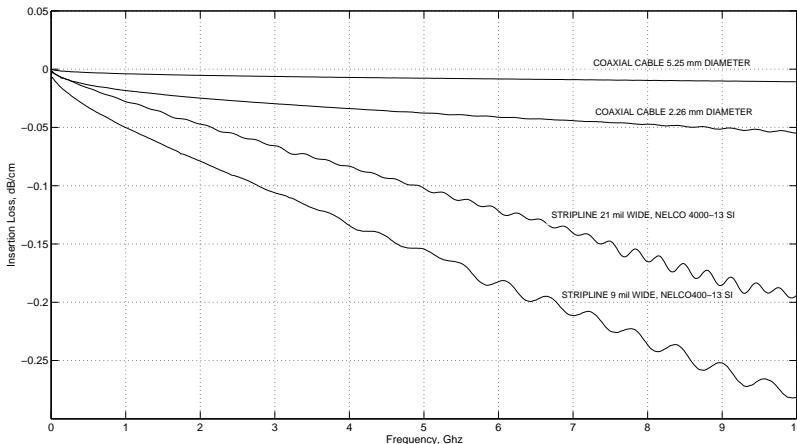


Figure 7.31 Comparison of the insertion loss between two coaxial cables with different diameters with two PCB striplines with different trace widths.

MECHANICAL CHARACTERISTICS						
UTIFLEX TYPE	UFB142C	UFB142A	UFB197C	UFB205A	UFB293C	UFB311A
Outer Diameter inch (mm)	0.142 (3.61)	0.142 (3.61)	0.197 (5.00)	0.205 (5.21)	0.293 (7.44)	0.311 (7.90)
Center Conductor Type	Stranded	Solid	Stranded	Solid	Stranded	Solid
Weight grams/ft (meter)	9 (29.5)	10 (32.8)	18 (59.1)	20 (65.6)	37 (121.4)	42 (137.8)
Minimum Bend Radius inch (mm)	0.38 (9.65)	0.38 (9.65)	0.50 (12.70)	0.50 (12.70)	0.75 (19.05)	1.25 (31.75)
Flexures	35,000	5,000	35,000	5,000	35,000	5,000
ELECTRICAL CHARACTERISTICS						
Impedance (ohms)	50	50	50	50	50	50
Frequency Range (GHz)	DC-18	DC-40	DC-26.5	DC-26.5	DC-18	DC-18
Velocity of Propagation	83%	83%	83%	83%	83%	83%
Capacitance pF/ft (meter)	24.5 (80.4)	24.5 (80.4)	24.5 (80.4)	24.5 (80.4)	24.5 (80.4)	24.5 (80.4)
Shielding Effectiveness (dB @ 1 GHz)	> 100	> 100	> 100	> 100	> 100	> 100
Maximum Insertion Loss dB/ft (meter)	See figure on next page					
1 GHz	0.12 (0.39)	0.11 (0.36)	0.09 (0.30)	0.08 (0.26)	0.06 (0.20)	0.05 (0.16)
10 GHz	0.38 (1.25)	0.33 (1.08)	0.30 (0.98)	0.23 (0.76)	0.19 (0.62)	0.15 (0.49)
18 GHz	0.51 (1.67)	0.45 (1.48)	0.40 (1.31)	0.32 (1.05)	0.28 (0.85)	0.21 (0.69)
26.5 GHz	—	0.55 (1.81)	0.49 (1.61)	0.39 (1.28)	—	—
40 GHz	—	0.68 (2.23)	—	—	—	—
Phase Stability vs Flexure*	10 GHz	1°	2°	1°	1°	3°**
	18 GHz	1°	3°	1°	2°	5°**
						5°**

Figure 7.32 Example of a low loss coaxial cable family spec sheet (courtesy of Micro-Coax).

Figure 7.32 shows an example of the specifications for a low loss coaxial cable family. For high-speed digital applications it is important to be able to map the application requirements into the coaxial cable specifications. The first important point to consider is if there are any mechanical restrictions that must be obeyed, for example:

Cable Flexibility: When a cable is intended to be flexed multiple times, then one should consider the use of a stranded core in the cable (a stranded core will have more loss than a solid one) and also make sure that the maximum bending diameter is acceptable.

Diameter: Is there a maximum diameter for the cable? How bulky can the cable be?

After the mechanical requirements are evaluated, one should now address the cable bandwidth requirements. Note that the maximum frequency described on the cable specification refers to the maximum frequency for which it is guaranteed that the propagation through the coaxial cable is without any unwanted behaviors like resonances. For computing the cable bandwidth, one needs to take the loss at the frequencies specified and the cable length that one intends to use. Some cable manufacturers specify the cable loss through an equation that approximates the real loss, allowing the cable loss for a specific length to be easily evaluated.

7.10.2 Coaxial Connectors

Coaxial connectors are typically used on ATE applications in two ways: either as part of cable assemblies or as connectors soldered on the test fixture. It is also possible that for certain applications and ATE interface architectures, coaxial connectors are used on the interface between the test fixture and the ATE pin electronics.

Confronted with the huge variety of coaxial connectors, an inexperienced engineer can easily get lost on all the different names. In this subsection we will concentrate on the type of connectors that are most relevant for high-speed digital applications.

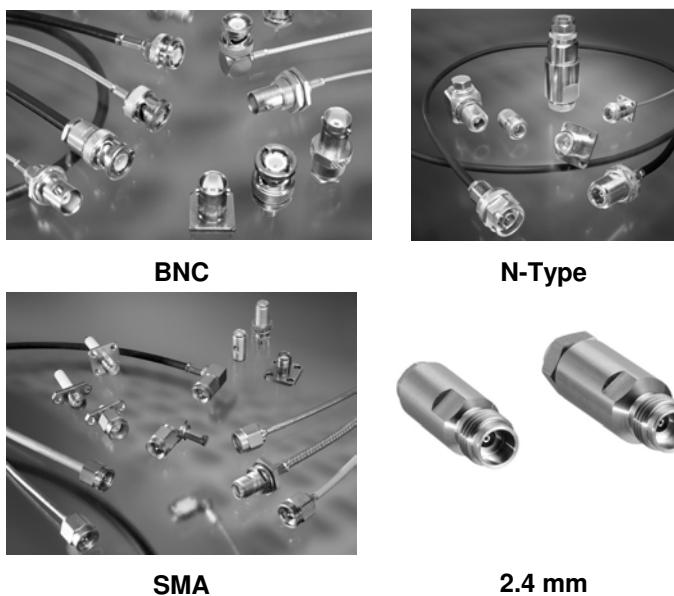


Figure 7.33 Example of coaxial connectors, top right: N-Type; top left: BNC; bottom right: 2.4 mm; bottom left: SMA (courtesy of Rosenberger).

Figure 7.33 shows several examples of coaxial connectors for cable assemblies. When choosing a connector, the following items should be taken into consideration:

Do I have to mate the connector to a specific instrument?

This might define which connector you must use.

What is the minimum bandwidth that I need for the connector?

40-GHz bandwidth connectors are available if needed, but if the application only requires 10-GHz bandwidth, there is no need to go

through the extra cost and effort. Other items like wear out cycling and mechanical robustness might also be important depending on the application. Make sure you understand your application needs.

Table 7.1 shows some of the most important coaxial connector families. Note that some connectors are sometimes further divided into categories like standard and precision, which typically imply different performances. One should always inspect the connector manufacturer specifications. Note also that although two different types of connectors might be mechanically compatible, it might not be appropriate to mate those different types since it can damage the higher precision connector type due to manufacturing tolerances.

Table 7.1
Typical Coaxial Connector Families for High-Speed Digital Applications

Type	BW	Compatible Mating Types	Notes
BNC	2 GHz	BNC	
Type N	18 GHz	Type N	
SMA	18.0 GHz 26.5 GHz	SMA; 3.5 mm; K-type	Typically Teflon dielectric core
3.5 mm	26.5 GHz	3.5 mm; SMA; K-type	Air core with some rated at 34 GHz
K-type	40 GHz	K-type; SMA; 3.5 mm	2.92-mm connector
2.4 mm	50 GHz	2.4 mm; 1.85 mm	
APC-7	18 GHz	APC-7	Sexless connector typically used in metrology
1.85 mm	67 GHz	1.85 mm; 2.4 mm	
SMP	40 GHz	SMP	
BMA	18 GHz	BMA	Blind mate connection

It is always possible to move from one type of coaxial connector to another through the use of a coaxial connector adapter. Figure 7.34 shows several different adapters. The drawback of using a coaxial adapter is that it will add cost, they take space, and, of course, it will degrade the signal, although the degradation can be minimal if an appropriate connector adapter is chosen.



Figure 7.34 Example of several types of coaxial adapters.

Another important point regarding coaxial connectors on cable assemblies for high-speed digital applications is the importance of using an appropriate torque wrench as shown in Figure 7.35 (left) for mating the connectors. Note that each connector family might require a different torque value. Table 7.2 shows the parameters for the SMA, 3.5-mm and 2.4-mm connector types. Note that SMA and 3.5-mm connectors require different torque values.



Figure 7.35 Torque wrench for correctly connecting a connector (left) and connector measurement gauge (right).

Dirty connectors should be cleaned using a cotton swab and isopropyl alcohol (IPA). In connectors with air dielectric, one should not clean the connector pin since it is easy to damage, and, in fact, any dirt on the inside of the connector will go to the back where you cannot reach it any more. To

Table 7.2

Toque Wrench Requirements for Typical Connector Families Used on High-Speed Digital Applications

Connector Type	Wrench Size (in)	Preset Torque (lb/in)	Preset Torque (N/cm)
3.5 mm, 2.4 mm	0.312 HEX	8	90
SMA	0.312 HEX	5	56

avoid getting your connectors dirty, you should keep the plastic plugs on them when not used (that is why an expensive 3.5-mm adapter comes with plastic plugs for protection).

It is important to note that one damaged connector might damage any connectors that you mate it with. Meaning that if you have a damaged connector, you should simply throw it away. A more difficult connector damage to assess is if the height of the connector center pin is still within specifications. Changes due to damage might have performance impact on high-precision measurements or calibrations, and it is not easy to realize that a given connector has a problem. To address this issue, a connector gauge like the one shown in Figure 7.35 (right) should be used. With this instrument one can measure the height of the connector center pin and check if it is within the allowed range. As a final note on connector protection, remember that not all connectors are manufactured in the same way. There is a reason for the price difference between an SMA connector and 3.5-mm connector. The 3.5-mm connector is a precise machined mechanical part and if you mate it with a cheap SMA connector you might damage the expensive 3.5-mm connector. This is especially important with expensive bench equipment or calibration kits. In this case avoid connecting different connector types directly to the instruments or calibration standards and instead include an appropriate adapter to protect the most delicate connector.

The other type of coaxial connectors used on ATE applications are the ones that can be assembled into the test fixture. The number of designs and types of connectors for printed circuit board (PCB) assembly is even larger than for coaxial cable assemblies. PCB connectors for high-speed digital applications can be divided in the following groups:

- Edge mounted;
- Surface mounted;
- Through hole mounted;
- Hybrid through hole/surface mounted.

Figure 7.36 shows several type of connectors for PCB assembly. It is important to understand that the performance of a coaxial connector on a PCB assembly does not only depend on the performance of the connector itself, but also on the footprint used on the test fixture PCB for mounting the connector, and especially on how the transition is made from the signal trace to the connector. Although some manufacturers provide default footprints for their connectors, for high-speed digital test and measurement applications it is sometimes necessary to develop a custom footprint and transition strategy for a specific connector. Figure 7.37 shows an example of a custom footprint and transition design developed using a 3D EM field solver (Appendix G) for a specific ATE test fixture.



Figure 7.36 Example of coaxial connectors for PCB assembly.

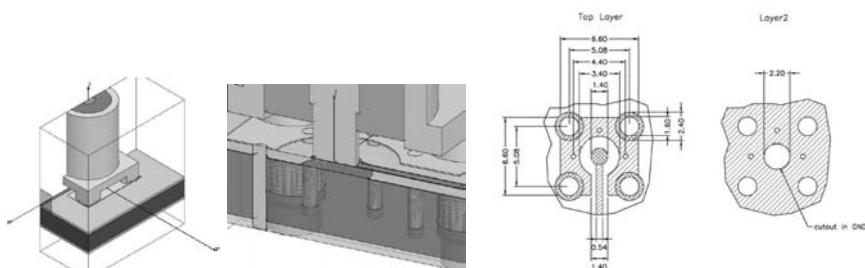


Figure 7.37 Example of an optimized footprint for a microstrip transition to a surface mounted SMA connector (courtesy of Rosenberger).

7.11 Accessories

When connecting or integrating bench instrumentation to an ATE system, there are several accessories that might be needed. This section presents a brief description of some important accessories.

7.11.1 Power Splitters and Power Dividers/Combiners

Power splitters and power dividers/combiners are two types of accessories that are frequently used for high-speed digital measurements. There is sometimes confusion regarding the difference between a power splitter and a power divider/combiner. Figure 7.38 shows a circuit diagram of a power splitter and a power divider/combiner. Both the power splitter and the power divider allow the splitting of a signal in two identical signals, each one with half of the original power. The difference is that while on the power combiner all ports are backwards matched (i.e., any reflections that arrive on any port of the power combiner are absorbed), on the power splitter only port 1 is backward matched, meaning that any reflections that arrive on ports 2 and 3 of the power splitter will not be absorbed [15].

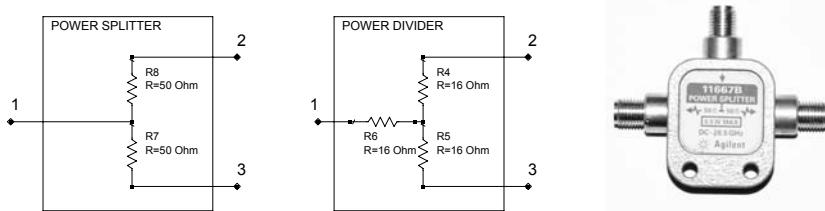


Figure 7.38 Block diagram of a resistive power splitter and a power divider/combiner (left) and a picture of a power splitter (right).

Skew Calibration of Two Oscilloscope Channels

In some applications it might be necessary to measure the skew between two signals, but usually one finds that measurement instruments already have an internal skew between the channels that is not fully calibrated. Assuming that we have a power splitter in which the skew between both paths is negligible for the application, the procedure described in Figure 7.39 can be used to deskew the measurement setup.

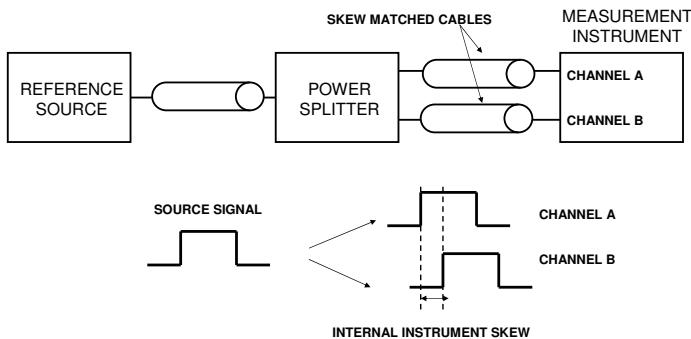


Figure 7.39 Calibrating the internal skew between two channels of a measurement instrument like an oscilloscope.

Providing a Termination Voltage when Using an Instrument with a Fixed Termination to Ground

Some measurement instruments only provide a fixed $50\ \Omega$ to ground termination that for some DUT outputs might be inappropriate, since they require a specific termination voltage. In these cases, one needs to add a small circuit to provide the $50\ \Omega$ to a specific termination voltage using power splitters and combiners before the measurement instrument input as shown in Figure 7.40 [4].

7.11.2 Attenuators, Blocking Capacitors, and Terminations

Attenuators, blocking capacitors (DC blocks), and coaxial terminations are three types of accessories that are very helpful when debugging an ATE application with bench instrumentation. Figure 7.41 shows a picture of these accessories.

The main application of blocking capacitors is to remove the DC component of a signal. In fact, it removes all the signal energy from DC to a certain frequency. This, in conjunction with the bandwidth of the blocking capacitor, defines its performance. Typical use cases are connecting a device output to an instrument with a fixed $50\ \Omega$ termination to ground or for protection of the measurement instrument (some measurement instruments inputs can be destroyed by a DC voltage above a certain threshold). When choosing a blocking capacitor, it is not only important to choose the blocking capacitor's maximum frequency according to the rise time of the signal to be measured but also to choose the minimum frequency according to the pattern to be applied [16]. Table 7.3 shows an example of the lowest frequency component for a 10-Gbps signal for three different PRBS patterns.

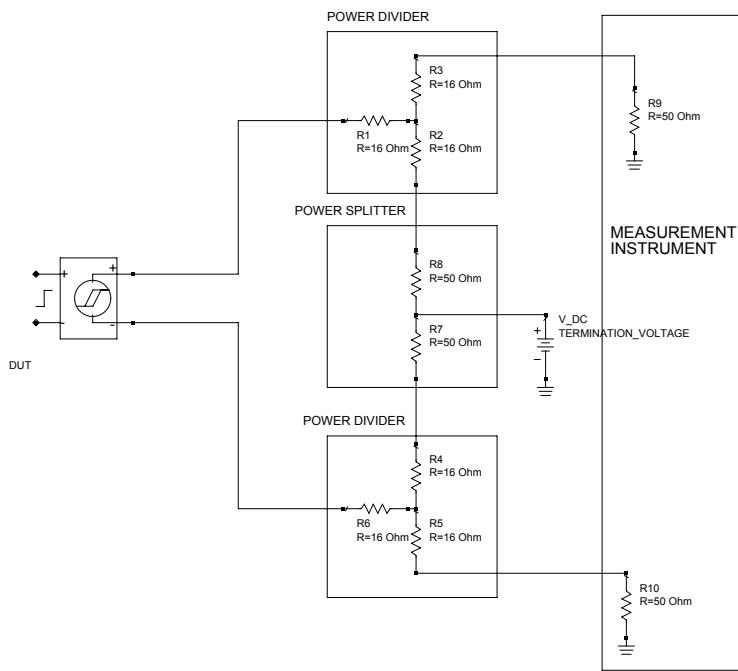


Figure 7.40 Possible circuit to provide a $50\text{ }\Omega$ termination to a specific voltage before a measurement instrument with a fixed $50\text{ }\Omega$ termination to ground.

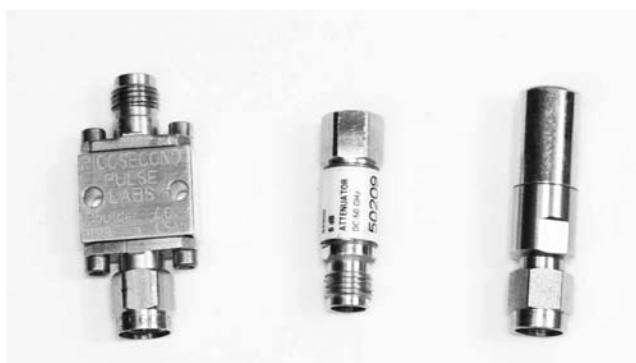


Figure 7.41 Blocking capacitor or DC block (left), a 6-dB attenuator (center), and a $50\text{ }\Omega$ coaxial termination (right).

Table 7.3

Lowest Frequency Component for a Digital Signal Running at 10 Gbps for Different PRBS Patterns

Pattern	Lowest Frequency Component
PRBS $2^7 - 1$	78.7 MHz
PRBS $2^{23} - 1$	1.2 kHz
PRBS $2^{31} - 1$	5 Hz

Attenuators as the name indicates are able to attenuate a signal. Attenuators are normally specified in dB; Table 7.4 provides a relation between dB and the attenuation factor. Typical applications are the attenuation of a signal before a measurement instrument input. The reason is that some measurement instruments have a maximum range for the signal amplitude they can measure and sometimes this range is smaller than the amplitude of the signal that is to be measured. Another application is to generate a low amplitude signal from a digital source by attenuation.

Table 7.4

Voltage Attenuation in dB to Attenuation Factor Conversion Table

Attenuation (dB)	Attenuation Factor
3 dB	0.71
6 dB	0.5
10 dB	0.32
20 dB	0.1

Coaxial terminations can be used to terminate nonused ports to prevent reflections. For example, when performing a measurement of a differential signal using a single-ended instrument, in most situations it is necessary to properly terminate the nonused leg of the differential pair with a $50\ \Omega$ termination. Another important application is the use of the termination as a $50\ \Omega$ reference to calibrate a measurement (e.g., a TDR). It is important to note that coaxial terminations can vary significantly in performance and price and it is important to choose the correct one depending on the measurement needs.

7.11.3 Pick-Off T

The pick-off T can be a helpful accessory to debug a high-speed digital application on an ATE system. It allows the user to probe the signal on a given signal path without a significant disturbance of the signal. The drawback

is that the probed signal has a significantly reduced amplitude and a worse performance than the original signal. The advantage is that the original signal is practically not disturbed and we are able to use this accessory at speeds even above 10 Gbps. Figure 7.42 shows an example of a pick-off T and a diagram of its usage.

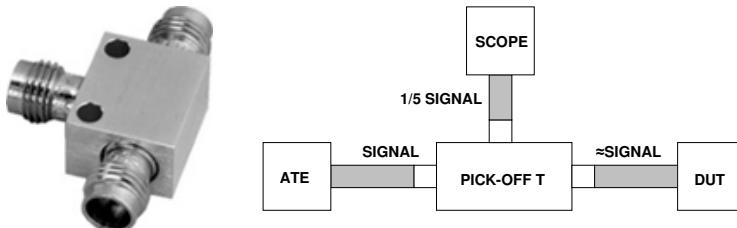


Figure 7.42 Picture of a pick-off T and a block diagram of its usage.

As the diagram shows, one possible use of a pick-off T is to monitor the signals between the ATE and DUT. Of course, this accessory for ATE applications requires some preconditions (i.e., the availability of appropriate connection points). Figure 7.43 shows an example result with a 10 Gbps data signal. Although the data eye probed by the pick-off T shows a significant degradation with regard to the through signal (which shows minimal degradation), it allows, for example, one to easily check if the through data pattern is correct.

7.11.4 Delay Lines

A delay line is another accessory that might be helpful in some situations. The type of delay lines we are interested in are of the passive type, also known as trombone delay lines. One example for this type of delay line is shown in Figure 7.44 where the delay can be adjusted mechanically from a minimum of 238 ps to a maximum of 293 ps.

Typical usage for this accessory is during the validation of a measurement setup with external instrumentation. In some situations it is necessary to skew a defined signal relative to others and this type of accessory provides a simple solution. It also provides the high bandwidth that is needed for high-speed digital signals.

7.11.5 Probes

Measurement probes are probably the most common instrumentation accessory among test engineers. There is a large amount of documentation

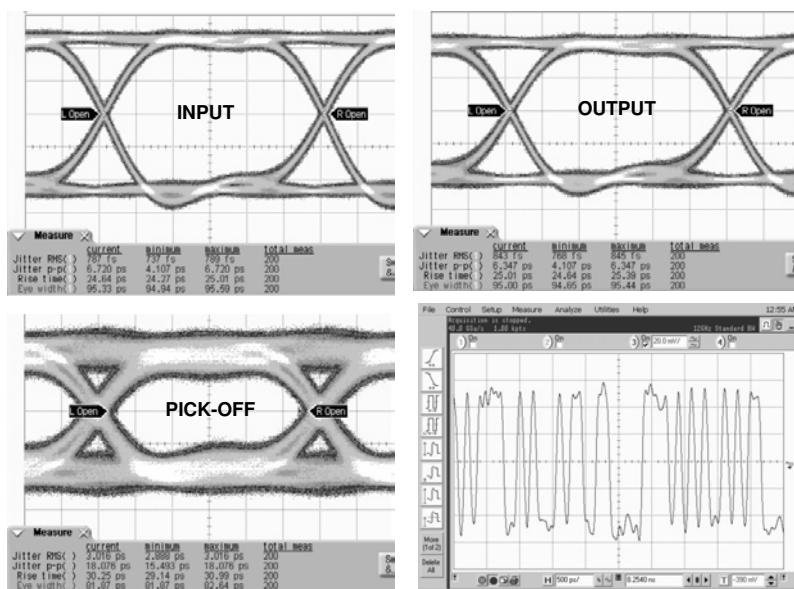


Figure 7.43 Example of the waveforms when using the pick-off T (Picosecond Labs model 5361) with 10-Gbps waveform and a PRBS $2^{31} - 1$ data pattern (top left: original signal; top right: signal after the pick-off T; bottom left: probed signal measured with an equivalent-time oscilloscope; bottom right: probed signal measured with a real-time oscilloscope).



Figure 7.44 Passive delay line example. The delay line is adjusted by mechanically changing its length.

material available on bench instrumentation probes. References [3, 4, 17] are good starting points. The Web sites of probe and bench instrumentation manufacturers are also good sources for information. In this section we will only provide a high-level overview on measurement probes for high-speed digital signals. Measurement probes can be divided in two major families: passive probes and active probes.

7.11.5.1 Passive Probes

Figure 7.45 shows a picture of different commercially available passive probes. The name “passive probe” indicates that there are no active elements inside (like transistors). For high-speed digital signals we are interested in a probe that provides a connection to the measurement point with minimum amount of distortion. This means for the typical high-speed application where $50\ \Omega$ PCB signal traces are used, the probe design should create the minimum possible discontinuity at the probing point. The large passive probe on Figure 7.45 (left) connects to a coaxial cable and provides a signal and a ground tip to attach to the ground reference with a variable pitch. This type of probe provides significant flexibility but at the cost of signal performance. The critical geometry factor in a probe is the loop size between the signal and ground pins, and in this probe the flexibility to address any signal to ground pitches (e.g., the distance between the signal BGA pad and the closest ground BGA pad) creates a large loop reducing the probe bandwidth. On the other hand, the probe in Figure 7.45 (center) is also passive, can be hand-used, and provides a much higher frequency at the expense of a smaller pitch and less flexibility. Finally the coplanar microcoaxial probe in Figure 7.45 (right) is also passive but the signal to ground pitch is fixed to a value of 400 micrometers in this example. This allows the highest bandwidth of the three probe examples but the least amount of flexibility.



Figure 7.45 Passive probe example. Left: $50\ \Omega$ low-frequency single-ended passive probe; center: high-frequency differential passive probe; right: microcoaxial $50\ \Omega$ passive coplanar probe (picture courtesy of GGB Industries).

It is also possible to construct a passive $50\ \Omega$ probe from a coaxial cable as shown in Figure 7.46. This type of self-made probe can be very useful and provides a high bandwidth if the signal ground leads are kept very short.



Figure 7.46 Examples of hand-made $50\ \Omega$ passive probes from a coaxial cable.

7.11.5.2 Active Probes

Figure 7.47 shows a picture of different active probes. An active probe tries to solve the challenge of how to probe a signal in situations where one does not want to disturb the probed signal (e.g., when probing a test point on a microstrip between the ATE and the DUT). Because of this, active probes are more sophisticated and contain electronic elements like a transistor. For active probes, the same concerns regarding the loop size between the signal and ground pins apply as with passive probes. With differential signaling becoming more prevalent, there are also differential active probes available on the market.

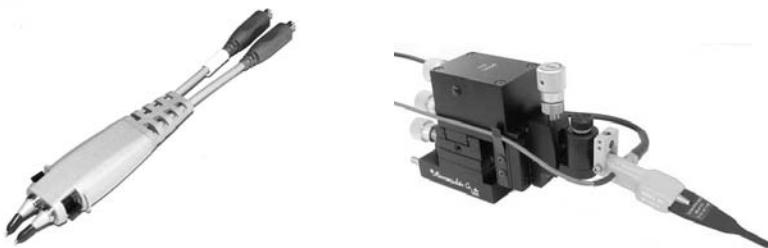


Figure 7.47 Examples of active probes. Left: differential active probe; right: single-ended active probe (courtesy of GGB Industries).

Although an active probe is usually applied in the context of ATE applications to monitor signals between the ATE pin electronics and the DUT, in some situations an active probe is needed to measure the performance of

the ATE system at the DUT socket. One example is to access the performance of the ATE system for an application where the DUT receiver input is a high impedance type input (e.g., a DDR address input). One approach to perform this measurement is to measure the performance at the DUT socket using an interposer and an active high impedance probe (interposers are discussed in detail in Appendix H). Figure 7.48 shows an example of this measurement. In this case a capacitor was also soldered between the DUT receiver input and ground pads to simulate the input capacitance of the DUT receiver since this can have a significant influence on the waveform shape.

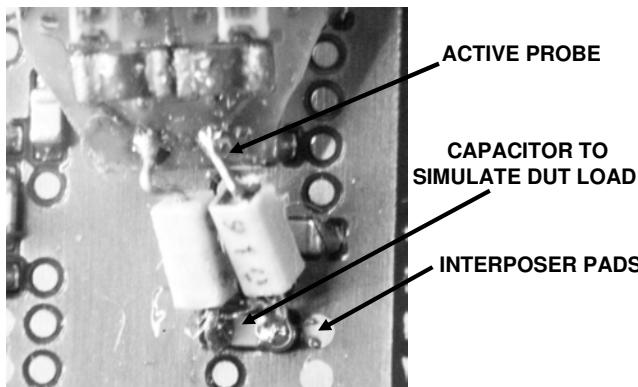


Figure 7.48 Probing a single-ended signal with a high impedance active probe (note that a capacitor was soldered to simulate the input capacitance at the DUT).

7.11.5.3 Probe Positioners

Although it is possible to use a probe manually to measure a signal, in some situations the test engineer needs the probe to be continuously connected to the probing point or the probe does not allow a simple manual positioning approach. In these cases a probe positioner like the ones shown in Figure 7.49 can be used. Another possible option is to use solderable probes, although this might not be possible in some cases because it damages the measurement point (e.g., the socket pads).

7.11.6 Balun

Sometimes test engineers face the challenge of measuring a differential signal with single-ended measurement instrumentation or providing a differential stimulus signal with a single-ended source. A typical approach is to terminate

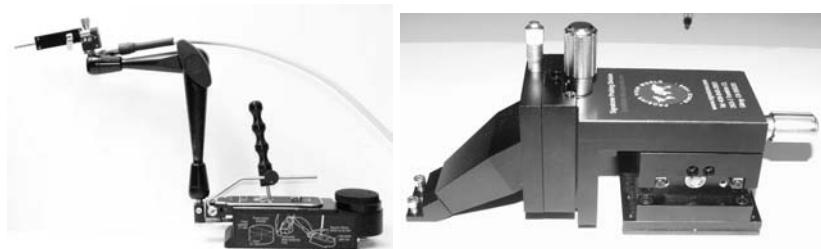


Figure 7.49 Examples of probe positioners (left: adjustable probe arm; right: micro-probe positioner with 3D adjustment).

one side of the differential pair and measure the other pair or use one single-ended instrument for each leg of the differential pair. Another option is to use a balun (BALanced UNbalanced transformer). Figure 7.50 shows a typical high-level schematic of a balun. Reference [18] describes a balun for high-speed digital applications. In Figure 7.50 the turns ratio of the transformer is one since it is assumed that both sides of the transformer have the same impedance. The input is the unbalanced (single-ended) input with one terminal of the transformer connected to ground. The output (secondary winding) is not connected to ground and in this way it is “floating” with respect to ground. This is a very simple example of balun construction and more complex designs are available for high-speed digital applications. Figure 7.51 (left) shows a photograph of a ultra-wideband balun. Another interesting application of a balun in the context of high-speed applications is to perform differential measurements with a standard two-port VNA (e.g., differential return loss) by using a pair of baluns as described in [19].

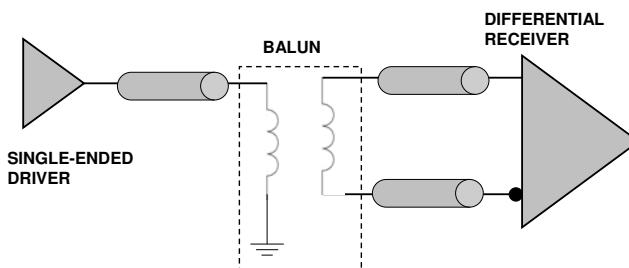


Figure 7.50 Typical high-level schematic of a balun.

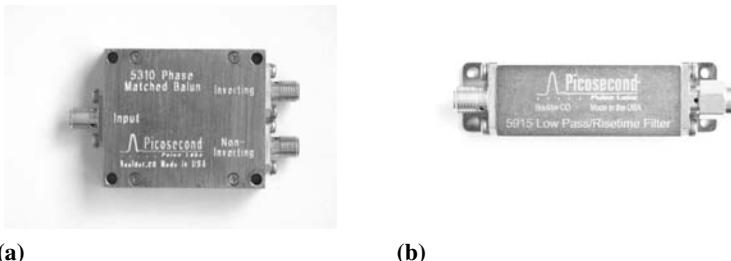


Figure 7.51 Picture of (a) a wideband balun and (b) a rise time converter.

7.11.7 Rise Time Converters

Typically the transition time of a stimulus source like a pattern generator is fixed to the best value the instrument manufacturer is able to achieve since this is considered an important specification (i.e., the smaller the rise time the better the driver in the instrument). In some situations the fast rise time of an instrument might present a problem to test a certain application. In this case the user needs to slow down the transition time of the instrument. This can be achieved by using a rise time converter (also called transition-time converters or TTCS) as shown in Figure 7.51 (right).

A rise time converter is designed to provide a specific rise time, assuming that the stimulus source has a faster rise time. The user needs to choose the correct rise time converter for the application. Figure 7.52 shows an example obtained with a rise time converter.

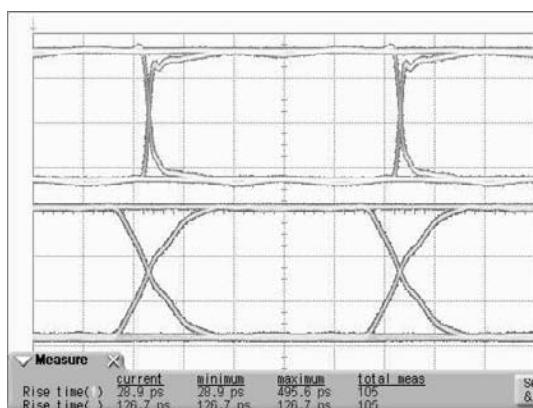


Figure 7.52 Example of using a rise time converter: stimulus source with 28-ps (20/80) rise time running at 2 Gbps (top) and after a 150-ps rise time converter (bottom).

Please also note that rise time filters are typically specified with 10%–90% rise times. To obtain the 20%–80% rise time, the following formula is used assuming a linear rise time in the 10%–90% region:

$$T_R(10\% - 90\%) \times 0.75 = T_R(20\% - 80\%) \quad (7.2)$$

References

- [1] N. Kularatna, *Digital and Analog Instrumentation: Testing and Measurements*. The Institution of Electrical Engineers, 2003.
- [2] D. Derickson and M. Mueller, *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization*. Prentice-Hall, 2007.
- [3] D. I. Geoff Lawday and G. Edlund, *A Signal Integrity Engineer's Companion*. Prentice-Hall, 2008.
- [4] W. Maichen, *Digital Timing Measurements*. Springer, 2006.
- [5] R. A. White, *Spectrum & Network Measurements*. Noble Publishing Corporation, 1993.
- [6] C. Rauscher, *Fundamentals of Spectrum Analysis*. Rhode & Schwarz, 2004.
- [7] M. Hiebel, *Fundamentals of Vector Network Analysis*. Rhode & Schwarz, 2007.
- [8] R. Stephens and R. Muro, “Characterization of Gaussian Noise Sources,” *IEC DesignCon*, 2008.
- [9] Tektronix, “Controlled Jitter Generation for Jitter Tolerance and Jitter Transfer Testing,” *Application Note*, 2005.
- [10] J. Moreira, B. Sadler, and D. Ferguson, “Parametric Testing of a 10Gbs I/O Cell in Production Through a Parametric Loopback Approach with Retiming,” *IEEE International Design and Test Workshop*, 2006.
- [11] J. Moreira, G. Haensel, and F. Koban, “Addressing the Challenges of Implementing an At-Speed Production Test-Cell for 10Gb/s Wafer Probing,” *DesignCon*, 2005.
- [12] H. W. Ott, *Electromagnetic Compatibility Engineering*. John Wiley and Sons, 2009.
- [13] E. B. Joffe and K.-S. Lock, *Grounds for Grounding: A Circuit-to-System Handbook*. John Wiley and Sons, 2010.
- [14] G. Haensel and K. Stieglbauer, “Optimized Method for Pattern Matching of High Speed Devices on ATE,” *IEEE International Test Conference*, 2005.
- [15] R. A. Johnson, “Understanding Microwave Power Splitters,” *Microwave Journal*, Dec. 1975.

- [16] G. Foster, “D.C. Blocks, a Trap for the Unwary When Using Long Patterns,” *Synthesis Research SR-TN088 Measurement Brief*, 2009.
- [17] A. Heltborg, I. Pollock, and R. V. Epps, “High-Speed Probe InterConnect Techniques,” *IEC DesignCon*, 2008.
- [18] J. R. Andrews, “Ultra-wideband Differential Measurements Using PSPL Baluns,” *Picosecond Pulse Labs Application Note AN-8*, 1999.
- [19] J. R. Andrews, “Differential VNA Measurements Using Single-Ended, Two-Port Instruments and BALUNs,” *Picosecond Labs Application Note AN-21*, 2008.

8

Test Fixture Design¹

The test fixture interface between the ATE system and the DUT (also known as DUT board, loadboard, or device interface board (DIB)) is a critical element in the performance of high-speed I/O characterization and production testing. In the past, books dealing with ATE testing (e.g., [1]) spent a significant amount of time on transmission line termination issues and techniques. This is no longer the major topic of concern since modern I/O interfaces and ATE pin electronics incorporate transmission line terminations on the DUT and the ATE pin electronics. The main challenge with multigigabit data rates is the transmission line losses across a test fixture and the effects of discontinuities like vias and relays. Several excellent books exist on the PCB design for high-speed digital applications (e.g., [2–7]). For those interested in the PCB manufacturing process itself, [8, 9] are a good starting point.

The ATE engineer must keep in mind that most books on high-speed signal integrity are written for a generic audience which includes engineers working on high-volume consumer applications where minimizing PCB layer counts and routing distances are critical for reducing manufacturing costs. A different set of trade-offs on the ATE test fixture can result in a completely different physical topology from that of the DUT application board. These differences can be seen in Figure 8.1 where a high-volume desktop computer motherboard is compared with an ATE test fixture. The cost sensitive consumer motherboard application has a high density of components and short routing distances as compared with the performance driven ATE test fixture with only a single IC (the DUT) and long routing distances to the ATE pin electronics.

Rules of thumb or design guidelines are always made for a specific application and must be reviewed for accuracy when leveraged to a new

¹In collaboration with Heidi Barnes.

design. When discussing high-speed PCB design guidelines with a colleague from a different area it is important not to assume that his requirements and constraints are the same as the ones for an ATE test fixture. Remember that a test engineer's objective is to measure as accurately and cost effectively as possible the performance of the DUT while conforming to the ATE interface. On the other hand, designers of a multicomponent PCB for an end user system only need to guarantee functionality with the lowest cost and smallest form factor.

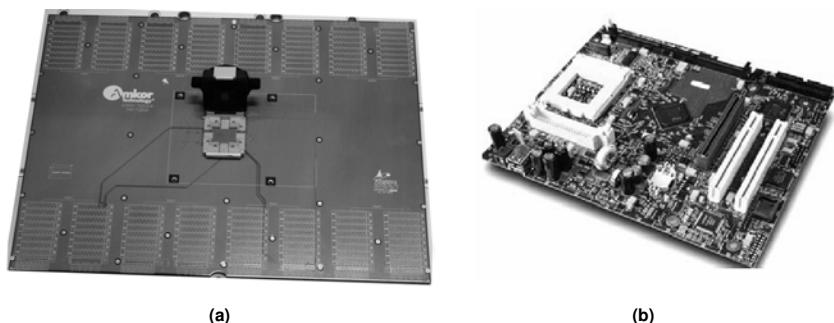


Figure 8.1 Comparison of a (a) high-speed digital test fixture PCB design compared with (b) "traditional" PCB design.

The signal path on an ATE system is composed of a chain of four basic parts as shown in Figure 8.2. Starting with the ATE pin electronics the signal travels through the ATE pogo pin assembly to the PCB test fixture and ends at the DUT socket. A simple DC continuity check to validate the netlist of the ATE to DUT connections along this signal path is no longer sufficient. The signal integrity of each of these parts is important and any one of them can break the high-speed digital connection.



Figure 8.2 The four parts of the ATE signal integrity chain (the ATE pin electronics card, the pogo pin assembly, the test fixture, and the DUT socket).

The ATE pin electronics card and the pogo pin assembly (or any other approach to connect the ATE pin electronics to the test fixture) are the responsibility of the ATE manufacturer. The situation is different for the test

fixture and DUT socket performance where the ATE user must define the requirements for interfacing with their product. The ATE user will typically design this interface with the expectation that the design will work the first time and that no “prototyping” is needed. This requires close collaboration with the ATE manufacturer, the test fixture layout and fabricators, and socket manufacturers to leverage past experience and guarantee that the test fixture and socket have the required performance. In the next sections of this chapter we will present some of the important topics associated with the design of test fixtures for high-speed digital applications.

8.1 Test Fixtures

Test fixtures can be significantly different depending on the ATE system model, manufacturer, and the intended application (RF, high-speed digital, wafer test, package test). Figure 8.3 shows an example of two high-speed digital application test fixtures from two different ATE vendors. One of the test fixtures demonstrates a single socket interface which is typical for higher performance characterization and design verification while the other test fixture utilizes four sockets to increase the throughput and lower the production test cost of a large digital device such as a microprocessor. Each of these design examples depends upon the ability of a multilayer PCB to route the interconnections between the ATE system and the DUT. The larger size and high layer count of this type of ATE test fixture is not common within the PCB industry and often requires highly specialized PCB fabricators as well as specialized design techniques which will be addressed throughout this chapter.

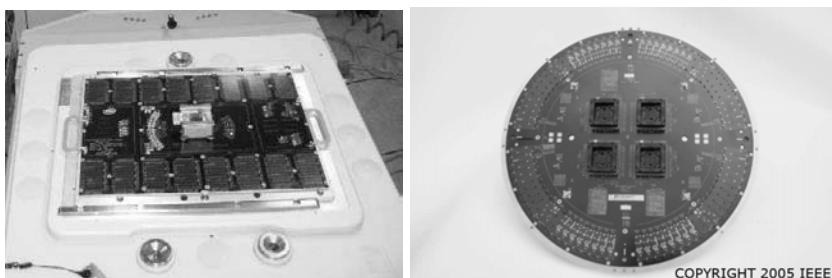


Figure 8.3 Examples of ATE test fixtures for high-speed digital applications (left: Verigy V93000; right: Advantest T2000 [10]).

Another style of ATE test fixtures can be found in the production testing of high-speed memory devices where multiple sites are required on a single ATE test fixture. In this case a combination of high density cabling to an array

of small DUT socket board PCB assemblies is utilized. This type of test fixture is also sometimes referred to as a “Hifix” since this was Advantest’s brand name for this type of test fixture. Figure 8.4 shows an example of a test fixture for a 256-site high-speed memory application. Note that the PCB portion of this test fixture is significantly different from that of a single site test fixture as shown on Figure 8.3. The use of multiple small PCB socket board assemblies with coaxial cabling to the ATE interface presents additional challenges in the design of the connecting cable transitions and the limited space for layout and support circuitry. Figure 8.5 shows a test fixture where the cover was removed to show the cable connections and a picture of a socket board.



Figure 8.4 A 256-site test fixture for high-speed memory applications (courtesy of Verigy).

The ATE test fixture also includes a mechanical metal frame to simplify the installation and removal (also known as docking and undocking) of a test fixture from an ATE system. This mechanical frame is often called a stiffener since it ensures that the PCB is flat and mechanically supported to handle the pressure from hundreds or even thousands of connecting pins on the ATE side and the insertion force of a DUT by an automated handler on the other side. Figure 8.6 shows an example of one of these mechanical frames. The mechanical frame creates constraints on the test fixture PCB design in terms of mounting holes, outer layer clearances, and board thicknesses and will vary according to the ATE manufacturer and system model.

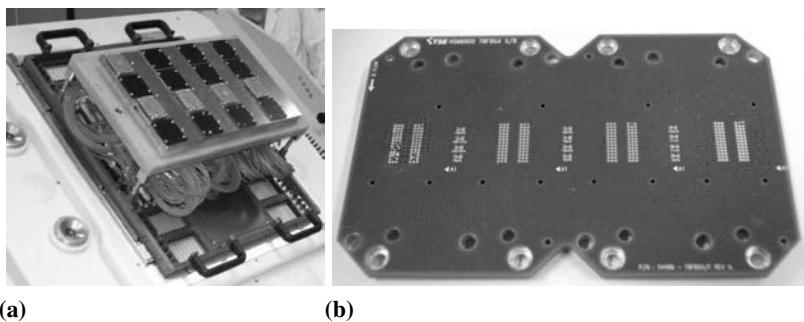


Figure 8.5 Example of (a) a multisite high-speed memory testing test fixture without the cover showing the cabling to the socket boards and (b) a socket board for four DUT sockets (courtesy of Verigy).



Figure 8.6 Picture of a mechanical frame (also known as stiffener) for a test fixture PCB (courtesy of Verigy).

8.2 High-Speed Design Effects

One of the difficulties in high-speed design is in understanding the multiple effects causing the degradation of a signal as the data rate increases and the rise time decreases. Signal degradation can be caused by impedance mismatches, material losses, conductor losses, radiation, and proximity effects as shown in Figure 8.7. One of the most confusing aspects of high-speed design is that the design rule trade-offs for minimizing these sources of signal degradation can be different depending upon the application and the PCB fabrication details. Since these effects are frequency dependent, it is important to look at each one and understand when they start to degrade a high-speed I/O signal.

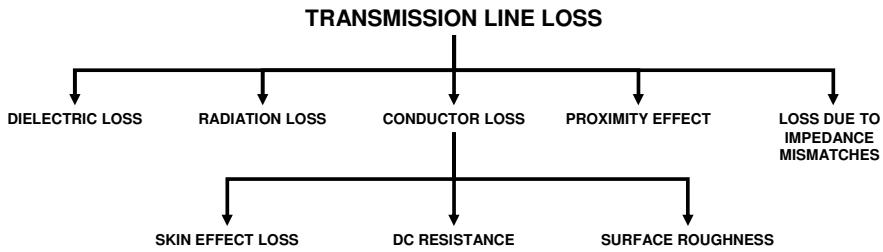


Figure 8.7 Loss factors for a printed circuit board signal trace.

8.2.1 Reflections Due to Impedance Mismatches

The problem of reflections due to impedance mismatches was first addressed by Oliver Heaviside in the late 1800s to describe the propagation of telegraph communication signals over long distances. He discovered that by breaking up the signal path into discrete sections with the incremental values of series inductance and parallel capacitance to ground and low resistive and conductive losses one could describe the transmission line as having a characteristic impedance with the following unique properties:

$$\begin{aligned}
 \text{Propagation Velocity: } \nu &= \frac{1}{\sqrt{LC}} \\
 \text{Characteristic Impedance: } Z_o &= \sqrt{\frac{L}{C}} \\
 \text{Reflection Coefficient: } \Gamma &= \frac{Z_L - Z_S}{Z_L + Z_S}
 \end{aligned} \tag{8.1}$$

where Z_L is the impedance of the load and Z_S is the impedance of the source.

The important discovery was the relationship between the size of the reflected signal and a change of the impedance along a transmission line. If there are impedance mismatches in the transmission line then some portion of the signal will be reflected and the transmitted signal at the end of the transmission line will be reduced in amplitude. A good analogy is to imagine a very long transmission line with constant impedance where a voltage step will result in a certain current flow determined by the transmission line characteristic impedance. Assuming the transmission line is eventually terminated in the same impedance, then the rate of current flow never changes and no voltage reflections occur. If the line is left open at the end ($Z_L = \infty$), then the current traveling down the line must go somewhere and is reflected back resulting in a positive voltage amplitude reflecting back along the transmission line. If the end of the line is a short ($Z_L = 0$), then the voltage

must go to zero and a negative voltage amplitude is reflected back along the transmission line.

The next consideration is the finite rise time or finite slew rate of a transmitted voltage step. Assume a transmission line which is much shorter than the distance for the propagation of either a rising or falling edge of a voltage step. Before such a slope reaches its full voltage some of the reflected signal from a nonideal load is already returning to the source and in this way supports the maintenance of the same voltage at the source and load. The assumption for a lumped circuit design is that the voltage at one end of a connecting trace is the same as at the other end, and this approximation is valid when the rising edge of a signal transition is on the order of 6 to 10 times the length of the connecting trace.

The rise time of the step edges for a high-speed digital signal are typically given in terms of the time to go from 10% to 90% or 20% to 80% of the voltage amplitude. This can be converted to physical distance by rewriting the Telegrapher Equation for the propagation velocity of a signal in terms of the dielectric constant:

$$\text{Propagation Velocity: } \nu = \frac{c}{\sqrt{\epsilon_r}} \quad (8.2)$$

where c is the speed of light in vacuum and ϵ_r is the dielectric constant of the material surrounding the signal trace. For FR4 as the PCB dielectric material with $\epsilon_r = 4.4$, the velocity on an inner layer stripline is approximately 15 centimeters per nanosecond.² This shows that for a multigigabit signal with a subnanosecond rise time the signal will transition from a low to a high in less than 15 cm of distance. If an ATE signal path is on the order of 15 cm then the signal at one end of the test fixture is not the same as at the other end and it can no longer be considered lumped element design. Designers often want the distance required for the rising edge to be 6 to 10 times the signal path for a true lumped approximation and in this case it clearly requires a distributed transmission line theory approach with matched impedances for optimum signal integrity.

When multiple impedance discontinuities happen in the middle of the signal path then one has to consider the double reflection that occurs when the reflection at one impedance discontinuity travels back towards the source and encounters another discontinuity so that the resulting double reflection is now traveling in the same direction as the original signal but is delayed and reduced in amplitude. The forward traveling reflection will then add and subtract from the original signal creating amplitude variations or ripple as shown in Figure 8.8. Again, if the impedance discontinuities are closer together than the 6 to 10

²6 inch per nanosecond.

times the length of the rising edge then the voltage differences between them are small and the amplitude of the ripple can be neglected. A 10-Gbps signal typically has a rise time below 50 ps which is approximately equivalent to a length of 7.6 mm (300 mil³) for a signal path on FR4 PCB material. A via on a high layer count ATE test fixture is close to this same length and can easily result in an impedance discontinuity at each end causing signal degradation due to this type of double reflection.

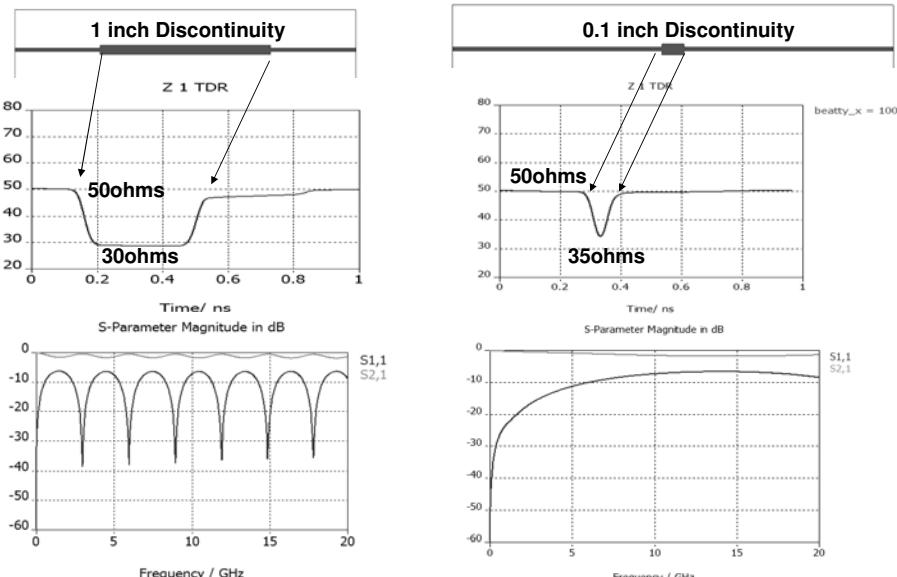


Figure 8.8 Comparison of the signal degradation in the time and frequency domain versus the length of an impedance discontinuity for a given signal with a 30-ps rise time.

As multiple impedance discontinuities occur along a signal path, one quickly finds that the interactions are data rate and rise time dependent and require simulations in order to predict the signal path losses. Certain combinations of closely spaced discontinuities can act like a lowpass filter and quickly attenuate the higher frequencies. The challenge for the ATE test fixture designer is to ensure that each section of a transmission line including the transition structures are matched in impedance.

³mil is the standard unit used for the width of a PCB signal trace and corresponds to 1/1,000 of an inch or 0.0254 mm.

8.2.2 Conductor Losses

The simple resistivity of a conducting material can describe the conductor losses at DC, but is no longer valid for higher frequencies. The conductor losses shown in Figure 8.7 include the frequency dependent skin effect and surface roughness losses in addition to the DC resistance. The skin effect loss, as the name indicates, is based on the fact that for increasing frequencies, the magnetic fields set up by the rapidly changing currents flowing in the conductor will force the electrical current to flow on the edge or “skin” of the conductor where the inductance is lowest. Figure 8.9 shows how this current distribution in a conductor changes with frequency and highlights that there is only a very small depth into a conductor where the high frequency currents are actually flowing [11].

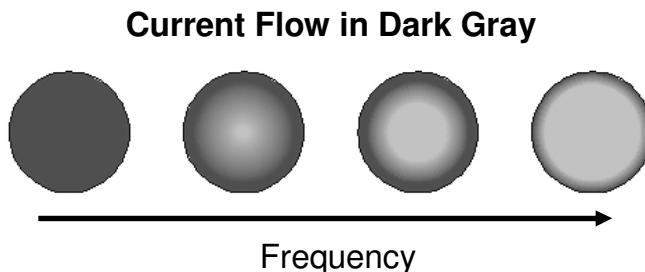


Figure 8.9 Skin effect on the center conductor of a coaxial cable.

The skin depth can be computed by (8.3) with ρ being the bulk resistivity of the core material (in $\Omega\text{-m}$, e.g., for copper it is $1.673 \times 10^{-4} \Omega\text{-m}$), f the frequency, μ_0 the permeability constant ($4\pi \times 10^{-7} N/A^2$), and μ_R is the relative permeability (usually 1 in most application cases).

$$\text{Skin Depth}(\delta_S) = \sqrt{\frac{\rho}{\pi f \mu_0 \mu_r}} \quad (8.3)$$

The density of current flow is falling off exponentially with distance into the conductor, and the skin depth represents the thickness of the outer skin that is carrying $1/e$ or 37% of the total current flow. Conductor losses that include skin effect will depend on the trace geometry and a simple approximation for a low loss transmission line with trace thickness much larger than the skin depth is [7]:

$$\begin{aligned} \text{Loss due to skin effect} &= 4.34 \frac{R_L}{Z_o} \text{ dB/m} \\ \text{Series resistance per length of stripline, } R_L &\approx \frac{\rho}{2w\delta_S} \end{aligned} \quad (8.4)$$

where w is the trace width in meters.

The skin depth losses are proportional to $1/\delta_S$ and thus increase with the \sqrt{f} making it difficult to predict the significance for a given digital signal. The skin depth can be increased and losses reduced by using a low resistivity conductor and avoiding magnetic materials. An ATE test fixture typically uses copper which is one of the metals with the lowest resistivity so little improvement can be made with the material selection; however, (8.4) shows that a shorter length conductor with a wider trace width will have lower losses.

The skin depth for copper at 5 GHz is only 0.9 μm which can easily be less than the height of surface roughness features on the conductor. This thin surface layer of current must follow the longer and higher resistance signal path of the conductor amorphous surface contours. Standard PCB fabrication processes often include surface treatment steps to roughen up the surface of the copper foil layers for improved mechanical adhesion [12] and at multi-gigabit data rates it becomes important to quantify the loss properties for a given lamination process. If a design is moved from one PCB fabricator to another then a reevaluation of the transmission line losses and electrical length are required to ensure that the electrical performance is still within the desired limits.

8.2.3 Dielectric Losses

The dielectric loss occurs when electromagnetic fields lose energy to the molecular dipoles found in the dielectric material between the signal and reference conductors in a transmission line. The amount of dielectric loss depends on the type of material and the frequency or rate at which the magnetic fields are changing. Figure 8.10 shows this effect. Dielectric loss, unlike skin effect loss, is independent of the trace or cable geometry being used and only depends on the dielectric material properties. Equation (8.5) shows a closed formula approximation for the dielectric loss. The $\tan \delta$ is the loss tangent or dissipation factor of the dielectric material which is a frequency dependent property that is specified along with the dielectric constant to determine the amount of attenuation or loss per meter for a given material.

$$\text{Dielectric Loss} = 92.0216 f \sqrt{\epsilon_r} \tan \delta \text{ dB/m} \quad (8.5)$$

Test engineers looking for the lowest dielectric and skin effect losses on a transmission line topology have found that coaxial cables with their lower loss dielectrics and larger signal conductors are superior to the geometries of a PCB trace. Two different implementations of coaxial cable routing on an ATE test fixture are presented in Figure 8.11. This type of approach is common in test fixtures for device characterization with low I/O pin counts, but becomes

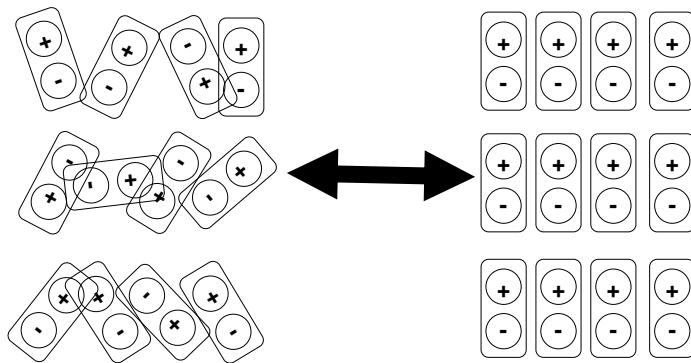


Figure 8.10 The polarization effect of the dielectric material molecular dipoles that generates the dielectric loss effect.

less efficient as the pin count increases and connections must be spread further from the DUT.

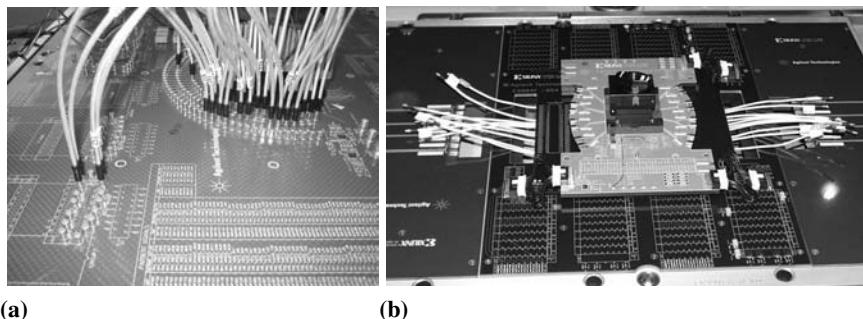


Figure 8.11 Test fixtures for the Verigy V93000 ATE system using coaxial cables to address the PCB signal trace loss: (a) a single board approach and (b) a mother/daughter card approach.

Although a coaxial cable has a lower loss than a PCB trace (as shown in Section 7.10), this technique requires the addition of connectors to transition from the PCB to the low loss coaxial cable. Even though the connector PCB transitions are small relative to the total length of a cable, they can still have significant high frequency losses if they are not impedance matched. The signal degradation from a poorly designed transition can quickly degrade any benefit of the low loss coaxial cable topology and one finds that coaxial cables often increase the complexity of an ATE test fixture with no measurable benefit.

Optimizing the material selection and PCB trace geometries on an ATE test fixture can significantly reduce the high frequency losses and make it quite attractive for at-speed high density I/O testing. A PCB test fixture needs to have some type of dielectric material to separate the different copper layers that contain the signal traces and their respective return current path or ground. Ideally this would be air with the lowest possible dielectric loss (with the exception of a vacuum); however, this is not something that can be manufactured and one must search for the best balance of mechanical properties versus high frequency electrical performance. The selection of the dielectric material and the separation it provides between layers will also define the impedance of a given trace geometry.

The trade-offs between mechanical performance, electrical performance, and ease of fabrication have led to a wide selection of available PCB materials. The PCB dielectric materials can be divided in two groups: the ones that use a fiber glass mesh filled with resin and the ones that use no fiber glass. Table 8.1 provides a list of common dielectric materials used for high-speed digital test fixtures [13, 14].

Table 8.1
Some Dielectric Materials Typically Used for High-Speed Digital Test Fixtures PCB
(FR4 is Included for Comparison)

Material	ϵ_r	$\tan\delta$ (1 GHz)	$\tan\delta$ (10 GHz)	Relative Cost
FR4	4.4	0.018	N/A	1
NELCO 4000-13 SI	3.4	0.008	0.008	1.5
ARLON 25N	3.38	N/A	0.0025	1.75
ARLON 25FR	3.38	N/A	0.0038	1.75
ROGERS 4003	3.58	0.0027	0.0027	2
ROGERS 4350	3.5	0.0031	0.0037	2
TEFLON GLAS	2.4	N/A	0.0014	2
SPEEDBOARD C	2.6	0.004	0.004	2
FASTRISE 27	2.7	<0.002	<0.002	2
TSM29	2.94	0.0012	0.0014	2

Figure 8.12 compares the insertion loss of different dielectric materials for a 482.6- μm (19-mil) wide inner stripline PCB trace using measured data from ATE test fixtures and simulated data from manufacturer material specifications. One immediate observation is that the simulated results are more optimistic than the measured ones. This is expected since the simulation model is based on the material manufacturer's data sheet which does not include the final "as-fabricated" properties such as surface roughness, moisture content, and lamination effects. The manufacturer's data sheet also

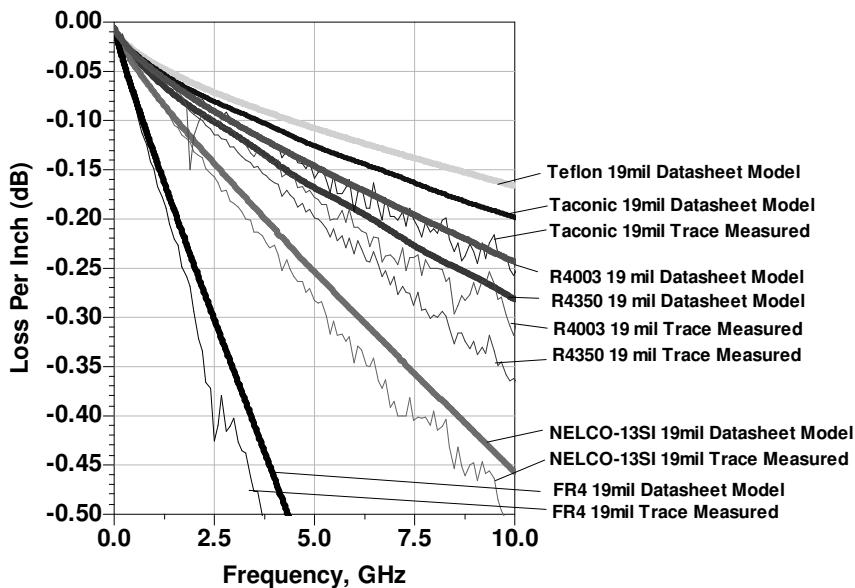


Figure 8.12 Simulation and some measurements for the insertion loss of different dielectric materials using the same trace width of 482.6 μm (19 mil).

relies on a simple IPC standard for measuring bulk dielectric constant and loss tangent in the z-axis direction, but PCB materials with glass weave are not homogeneous and signal traces running in the x- and y-axis directions can see different material properties as shown in Figure 8.13. The best way to improve the accuracy of high-speed test fixture design and optimization is to obtain measured data of the material properties and ensure that all of the loss factors are included in the simulations.

Knowledge of the loss per length versus frequency for the different PCB materials is valuable for running design simulations to optimize the selection of dielectric material and trace geometries, but for comparison of high-speed digital performance it is also helpful as shown in Table 8.2 to view the digital signal performance in the time domain using, for example, the data eye diagram. The eye diagrams for the different materials shown in Figures 8.14, 8.15, and 8.16 demonstrate that for increasing data rates the data eye diagram for standard low cost FR4 PCB material degrades quite rapidly. Performance differences between the higher end materials is less significant and often becomes a balance of understanding the priorities of performance, cost, and manufacturability.

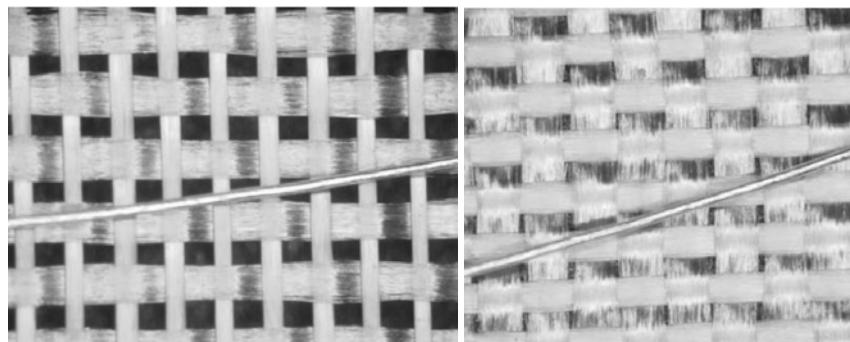


Figure 8.13 Photograph of two different types of fiber glass meshes with a copper wire on top. The glass mesh will change the local dielectric material properties which impacts the impedance of the signal trace and the propagation velocity. Tighter glass meshes mitigate this problem but at the expense of a higher dielectric loss due to the fiber glass.

Table 8.2

Comparison of the Measured Values for an ATE Test Fixture with a 25-cm (9.8-in) Length, 482.6- μ m (19-mil) Wide Trace in Different Dielectrics (FR4 Is Included for Comparison); Tr is the Measured Rise Time (20/80), Jpp is the Peak-to-Peak Jitter Value Measured at a Certain Data Rate

Material	Tr 10 Gbps	Jpp 1 Gbps	Jpp 5 Gbps	Jpp 10 Gbps
Taconic FastRise	27.0 ps	11.2 ps	10.9 ps	15.6 ps
ROGERS 4350	28.1 ps	11.2 ps	8.7 ps	14.9 ps
Nelco 4000-13SI	30.2 ps	11.2 ps	11.7 ps	16.7 ps
CCL EL230	30.2 ps	11.2 ps	10.9 ps	14.9 ps
ROGERS 4003	32.0 ps	11.2 ps	9.5 ps	14.9 ps
Hitachi FX-II	32.4 ps	11.2 ps	9.5 ps	13.2 ps
MEGTRON 6	33.8 ps	11.2 ps	11.7 ps	14.2 ps
FR4 (CEL 475SD)	57.9 ps	18.6 ps	20.4 ps	29.9 ps

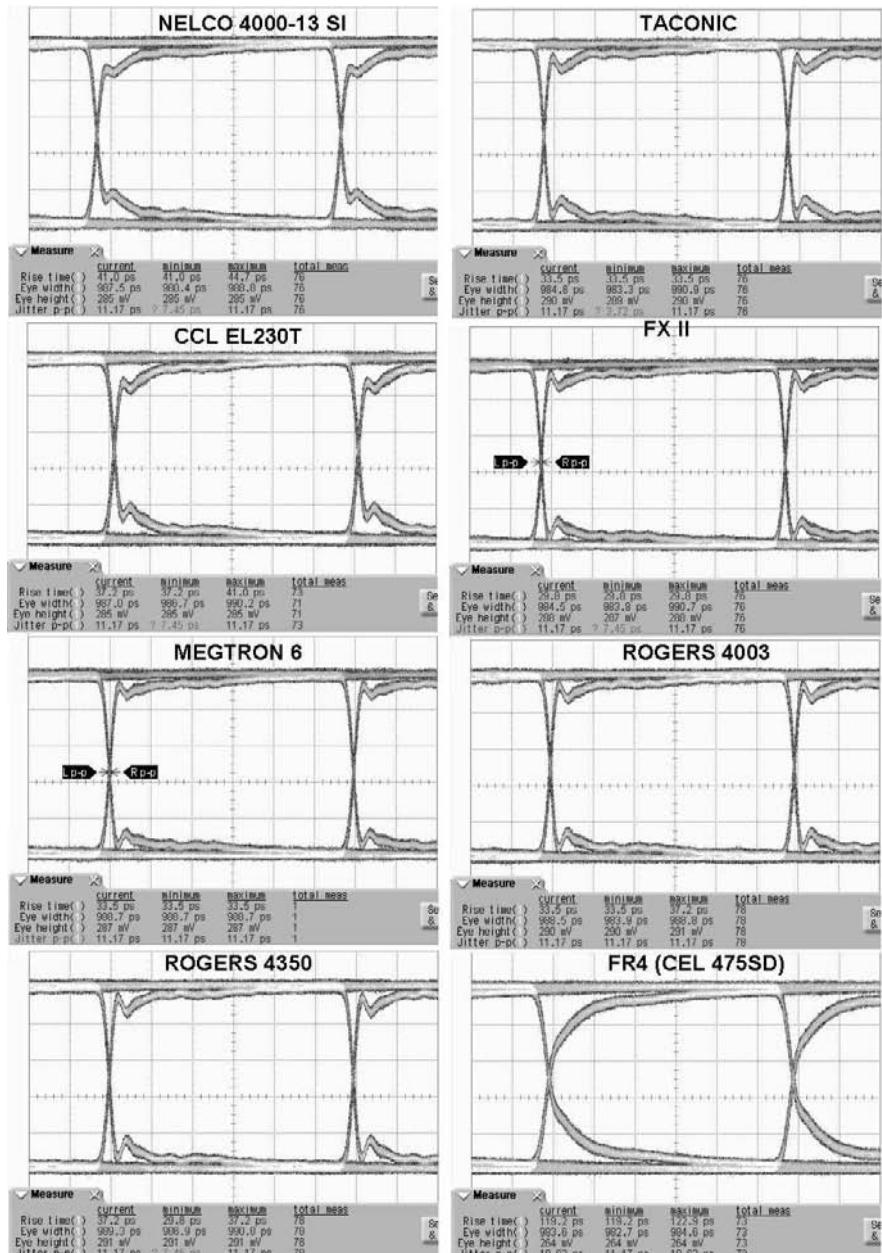


Figure 8.14 Data eye diagram comparison using different dielectric materials at 1 Gbps.

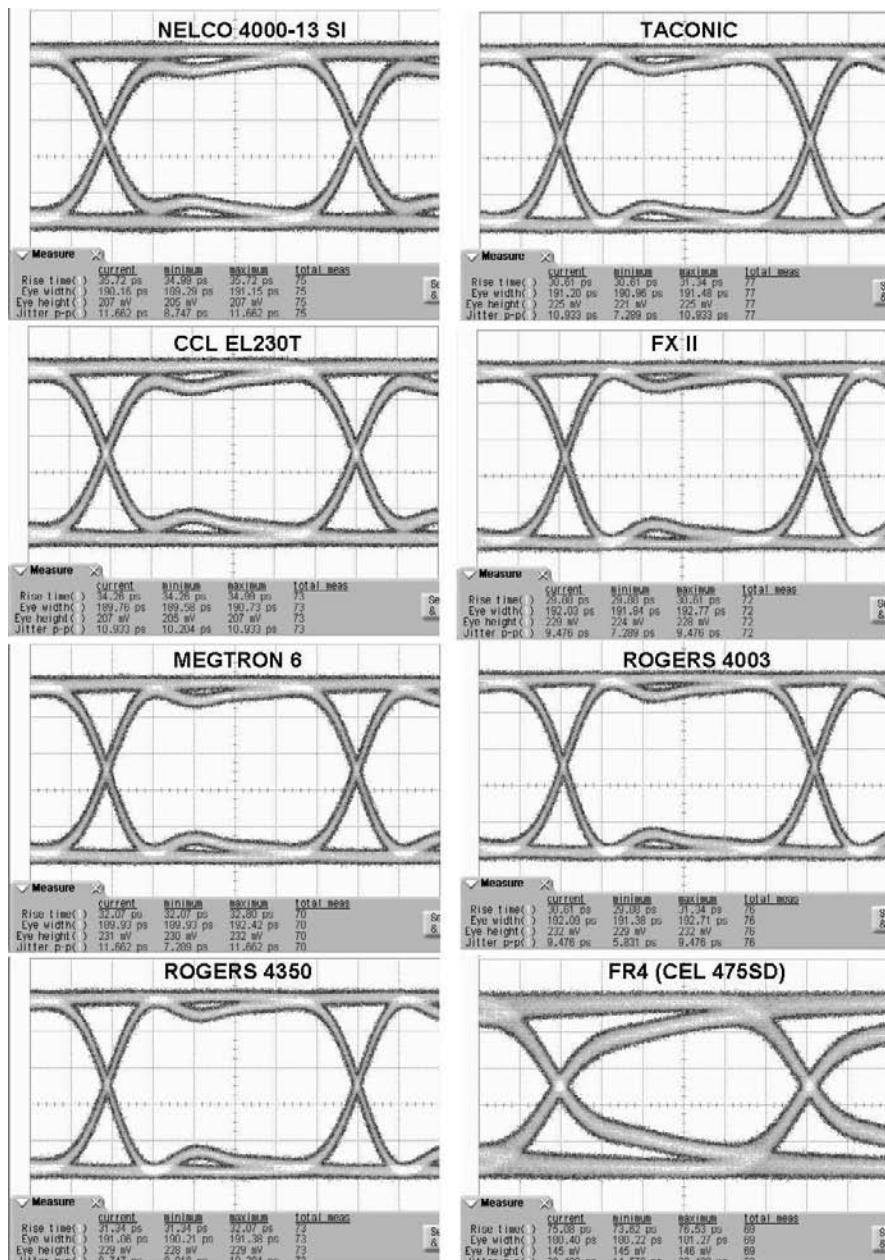


Figure 8.15 Data eye diagram comparison using different dielectric materials at 5 Gbps.

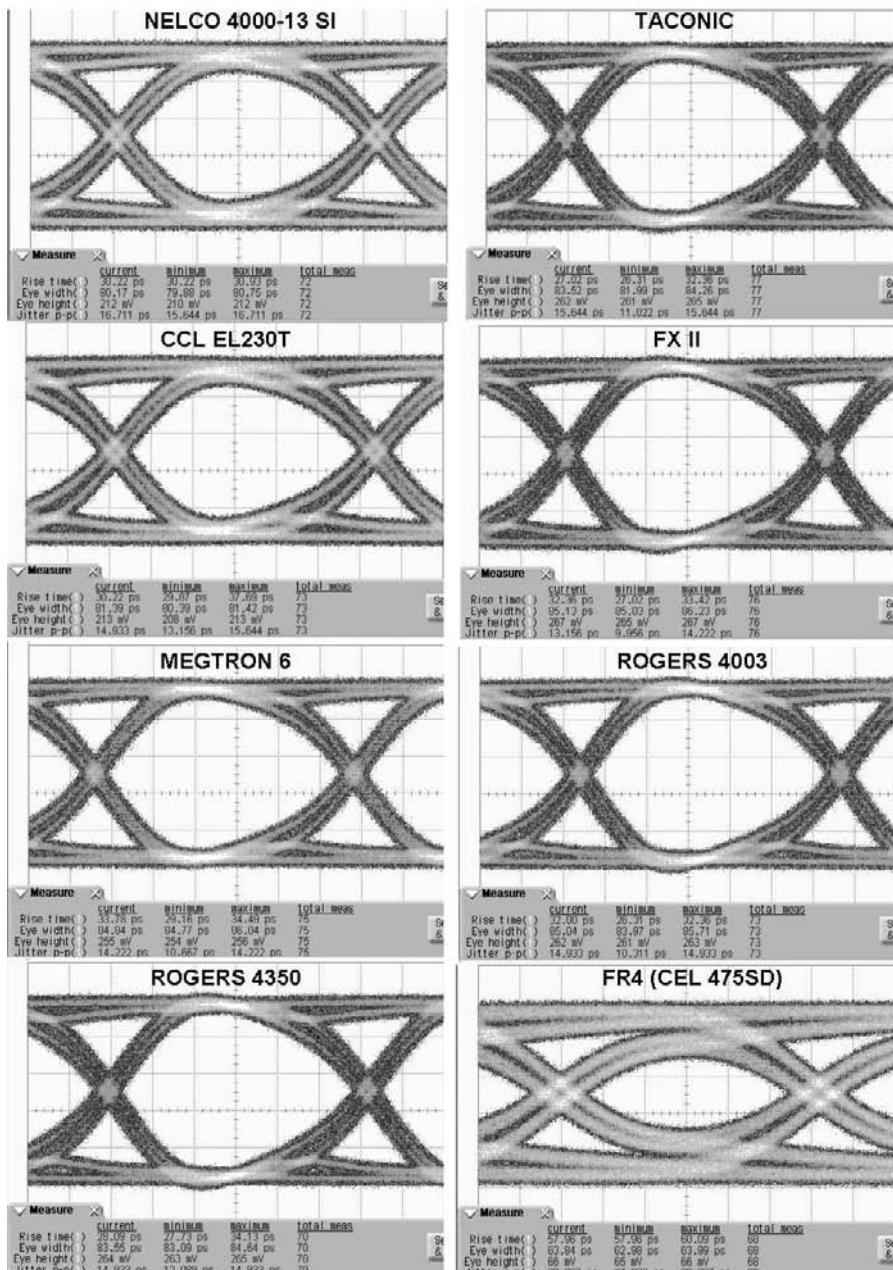


Figure 8.16 Data eye diagram comparison using different dielectric materials at 10 Gbps.

8.2.4 Crosstalk

The final source of signal loss that cannot be ignored is the capacitive and magnetic interaction or crosstalk between adjacent signal traces when using fast switching signals. The amount of crosstalk between an aggressor signal and a victim signal will be proportional to the aggressor voltage amplitude, its rise time, and the physical proximity between the aggressor and victim structures (e.g., the distance between two PCB signal traces on the test fixture). Figure 8.17 shows a diagram representing the possible crosstalk points on a typical ATE test fixture.

The higher pin counts of complex SOC devices along with faster signal rise times has increased the amount of crosstalk for high-speed digital applications. The amount of crosstalk depends on the routing topology of the ATE DUT test fixture which will differ from that of the end-user DUT application. A good introduction to the types of crosstalk between electrical signals, such as near-end crosstalk (NEXT) and far-end crosstalk (FEXT), can be found in the signal integrity references [4, 7]. Applications with only one DUT IC on the test fixture have some design flexibility when minimizing crosstalk effects while denser multisite test fixtures can become quite challenging.

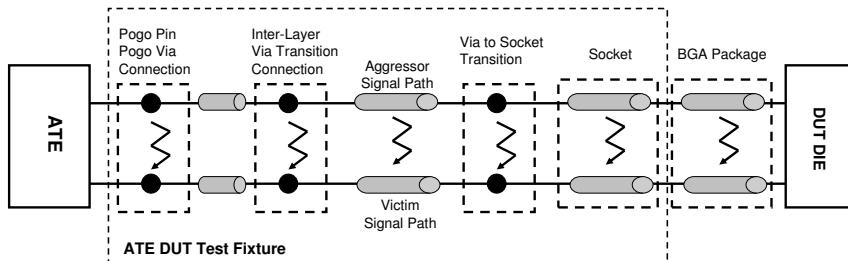


Figure 8.17 Diagram describing some of the areas where crosstalk might arise on an ATE test fixture.

The best way to minimize both NEXT and FEXT is to simply space the signals farther apart, and one finds that with just five times the trace width between microstrip traces on a PCB the NEXT can be reduced to less than 1%. Figure 8.18 shows this crosstalk reduction with gap spacing on the left for a long 250-mm routing distance and for decreasing risetimes on a short 25-mm routing distance. The maximum amount of NEXT happens when the coupling length or distance of interaction is greater than half the length of the full step edge. The NEXT is the sum of the capacitive and inductive coupled currents, while FEXT is the difference. In the special case of a homogenous dielectric around the conductor, like in stripline, then the capacitive and

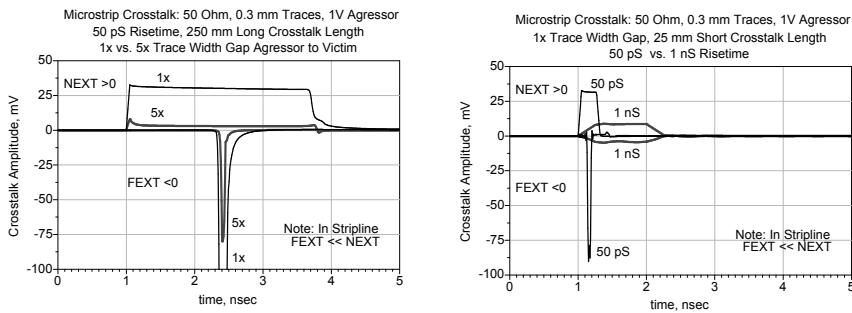


Figure 8.18 Simulation of adjacent stripline traces to show crosstalk reduction when increasing gap spacing from 1x to 5x on the left, and the change in crosstalk when the rise time is increased from 50 ps to 1 ns on the right.

inductive coupling is about equal and essentially cancels the FEXT. This cancelation of FEXT makes stripline the best routing method for high-speed signals, since even 25 mm of microstrip routing with 5x trace width spacing will have greater than 5% crosstalk for a signal with 50-ps risetimes. Stripline routing is already quite common on high density digital applications where multiple signal layers are required for routing into the DUT BGA via field.

Interlayer vias that transition between different PCB layers or the pogo vias that connect the test fixture to the ATE pogo pin assembly [15] do not have well-behaved transmission line EM fields and can be a significant source of unwanted crosstalk. Via transitions at low speeds are significantly shorter than the distance occupied by a rising or falling edge and crosstalk is not a major concern, but as frequencies increase even the short distance across a via can end up with a maximum amount of crosstalk. Crosstalk at vias is usually caused by an increase in the magnetic loop inductance of the ground path interacting with neighboring signals and the crosstalk is reduced by insuring adjacent return path ground vias. The magnitude of the crosstalk is difficult to predict without the use of a 3D EM field solver which also has the benefit of providing visualization of how the crosstalk is occurring.

There will be some cases where crosstalk exists that is outside of the control of the test engineer or test fixture designer. For example, if the package signal and ground ballout design is not optimized then one can get significant crosstalk on the vias and socket pins leading to the DUT. The ATE engineer cannot change the device ballout and is left with trying to shorten the distance of interaction by shortening the socket pogo pins and the PCB via length, or by adding additional ground shielding or return paths. To a certain extent the crosstalk in the ATE test fixture PCB via field for the DUT footprint is similar to what the application board will also see for a given ballout topology. The

ATE system typically has longer distances through the via and socket to the DUT so the crosstalk will be worse than in the end-user application, but still a good indication of the crosstalk problems that the end-user application can encounter.

8.3 Impedance Controlled Routing

8.3.1 Microstrip and Striplines

An ATE test fixture is a multilayer PCB where the signals are routed through copper lines etched into the copper layers of the board. There are several geometries that one can use to transmit signals on a PCB. They include microstrip on the outer layers, stripline on the inner layers, and coplanar microstrip with ground and signal on the same layer to name a few of the more common ones. ATE applications typically use microstrip and stripline geometries as shown in Figure 8.19. The more complicated coplanar structure is often used just for the short distances when transitioning between transmission lines such as on an edge launch electrical connector footprint to improve the impedance matching.

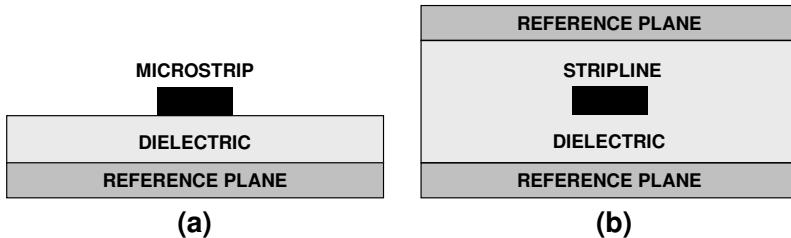


Figure 8.19 (a) Microstrip and (b) stripline geometries.

Figure 8.20 shows a cross-section picture of a PCB microstrip and stripline. The trace is not an ideal rectangle as one might expect, but it has a modified edge profile due to the etching process that etches sideways as well as down through the copper. The outer layer microstrip shown on the top with gold (Au) plating used as the mask for etching ends up with a combination of trapezoidal with a thin cap layer that overhangs the edges of the trace. The gold plating is typically required on ATE test fixtures to provide a robust nonoxidizing contact for the ATE pogo pins and DUT socket pin contacts and to prevent any exposed traces from oxidizing. Typical simulation tools assume rectangular shaped traces and will not always capture the additional high frequency losses due to this nonuniform etching at the edges and it is

helpful to obtain measurements to validate the losses for a given fabrication process.

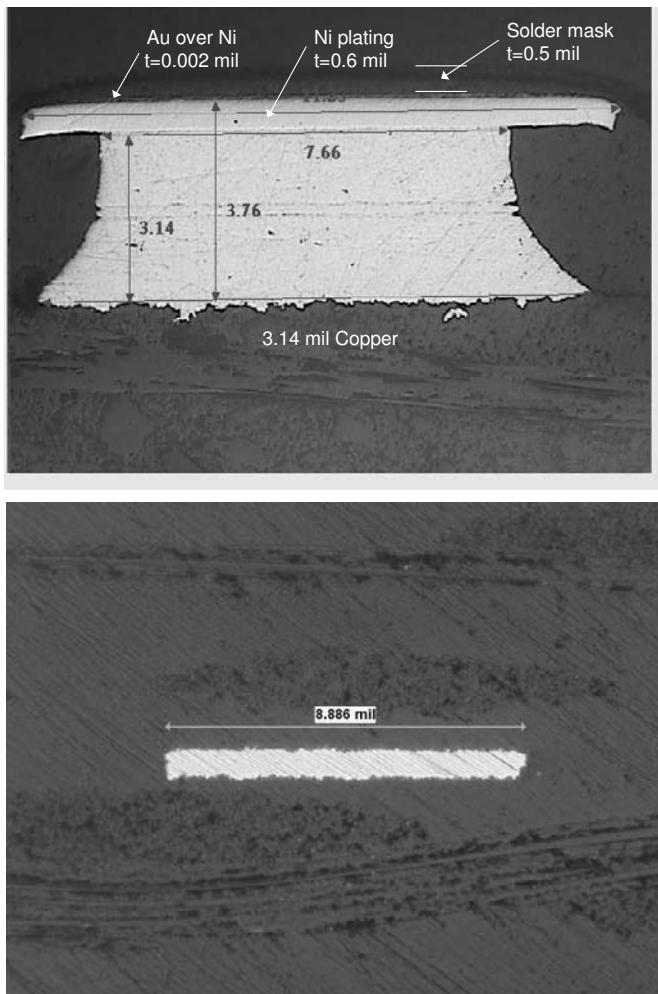


Figure 8.20 Cross-section of a microstrip (top) and a stripline (bottom).

The selection of a passivation layer to protect the outer layer copper microstrip traces impacts the signal performance due to the interaction with the skin effect and dielectric properties at the edge of the conductor. Table 8.3 presents several outer layer plating options for optimizing the electrical performance for a given microstrip design [16, 17]. At high frequencies the skin effect losses require the use of a low loss conductor for the outside plating with silver (Ag) providing the best performance. Hard gold is a good second

choice for passivation, but requires an additional nickel (Ni) barrier to keep the gold from diffusing into the copper. Since nickel is not a good conductor and is ferromagnetic (relative magnetic permeability can be as high as 100 at 1 GHz) which greatly increases the skin effect losses, it is important to use greater than 1.9 μm of gold for the outer plating on microstrip traces.

Table 8.3
Typical Surface Finish Options for a Microstrip Design

Plating Options	Advantages	Disadvantages	Cost
Thin Au: A thin <0.3- μm layer of gold (Au) is plated on top of a 5- μm nickel (Ni) barrier to the copper trace.	Lower Au content improves the soldering of SMT components.	Higher skin effect losses than thick Au plating.	1.05
Thick Au: A layer of >1.9 μm is plated on top of a 5- μm Ni barrier to the copper trace.	Low skin-effect losses, almost as good as silver plating.	Higher Au content in the solder joint can reduce SMT reliability.	1.1
OSP: Organic surface protective finish for improved soldering of tight pitch components.	Used on SMT pads, traces still require solder mask.	Loss is higher due to solder mask.	1.05
Silver: A layer of silver (Ag) is plated directly on top of the copper.	Lowest loss since no nickel barrier or solder mask is required.	Popular choice for lead-free SMT soldering.	1.15
Solder Mask: Typically a layer of 10 μm goes on top of the Cu trace that may or may not be plated with Ni-Au.	Protects against electro-migration, ESD, and Cu oxidation.	Increases the loss of the microstrip trace.	1

Figure 8.21 shows a comparison of the insertion loss for different plating approaches on a 533- μm (21-mil) wide microstrip. The results show that, as expected, silver plating is the lowest loss option but also the most expensive. The thick gold over nickel (without solder mask) has almost the same loss as silver plating at a lower cost. The very thin Au plating can actually be worse than putting soldermask on the thick Au plating option and clearly demonstrates the need for greater than 1.9 μm of Au plating on any outer layer microstrip traces. Even though the data shows that soldermask adds additional

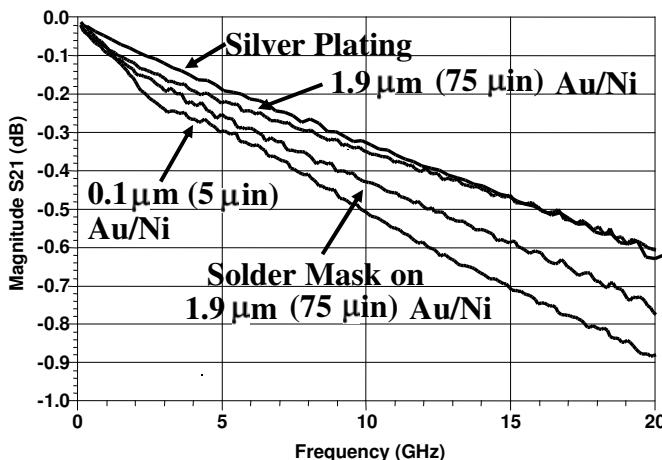


Figure 8.21 Comparison of the loss of different plating techniques for a microstrip trace.

losses, it is still used in cases where ESD protection is more important than the additional losses.

8.3.2 Differential Routing

Differential signaling, as the name implies, uses the difference of two signals to determine the digital signal of interest. This type of signaling benefits from a constant di/dt current demand on the power supply and common mode noise rejection; however, it doubles the number of I/O signal traces which increases the packaging and PCB costs. The difficulty with differential signaling is in understanding how the single-ended impedance and even mode signals interact with the differential impedance and the odd mode signals. A complete discussion of differential signaling and line impedances can be found in [7]. Figure 8.22 shows a diagram of an edge-coupled differential stripline pair with the key geometric values that determine the differential pair impedance. To compute the impedance of a given geometric configuration it is necessary to use a 2D field solver or an impedance computation tool as shown in Appendix G since an accurate closed formula for the impedance of an edge-couple differential stripline is not available.

A 100Ω differential signal trace can be composed of two single-ended 50Ω signal traces as long as they are length matched and separated by enough distance to prevent any strong coupling between the single-ended traces. Routing the two signals with a gap distance smaller than the trace width takes advantage of increased EM coupling between the traces to reduce

the signal sensitivity to ground discontinuities and common mode noise. However, the stronger the coupling the narrower the trace width needs to be designed to maintain 100Ω differential impedance. The coupled routing approach is very popular on end user application boards where space is limited, the environment is noisy, and ground discontinuities are likely. The longer distances used on an ATE test fixture drive a different set of trade-offs when routing differential signals.

The longer routing on an ATE application with the bends and turns can make it difficult to maintain a minimal skew between the coupled traces along the signal path. If the differences becomes too large, the hazard of common mode to differential conversion grows. Furthermore the coupling of traces requires a reduced trace width which generates increased losses for long routing distances. Single DUT characterization test fixtures often have dedicated internal signal layers with excellent noise immunity. This allows the test fixture designer to design a low loss differential pair by using a large trace width and using a distance between the single-ended signal traces of the differential pair that is at least three times the trace width.

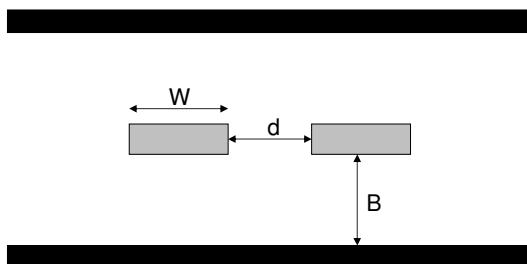


Figure 8.22 Diagram with key geometric values for determining the differential impedance of a differential PCB stripline.

8.4 Via Transitions

Vias are responsible for moving a signal between different layers of the PCB on an ATE test fixture as shown in Figure 8.23. At low speeds or DC levels the via topology is selected for robust manufacturing, high reliability, and available routing space. Looking back at Figure 8.7 it shows that at higher frequencies the impedance discontinuities from a change in the signal path topology will create reflections that can dominate the signal losses. The design of impedance controlled vias becomes critical for minimizing these reflections and applications like an ATE test fixture via design will depend on the required application performance and the available PCB topology [16, 18]. An ATE test

fixture can easily be 6 mm thick which is roughly 50 ps in length for the via transition and a significant portion of the rise time for a 10-Gbps signal.

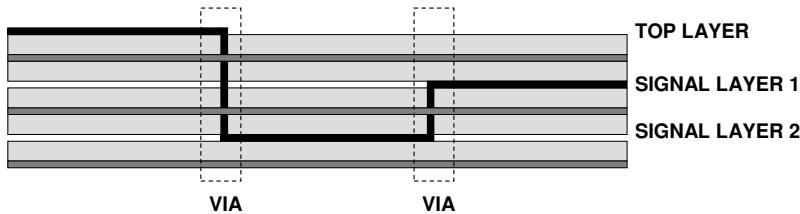


Figure 8.23 Vias on an ATE test fixture PCB.

The impedance of the via is determined by the relation of the signal via to the neighboring ground via structures. The parameters used to describe the via geometry are shown in Figure 8.24. Default combinations of these features defined by the PCB fabrication process may not have the required performance, and as speeds increase it is important that they are specified by the test fixture designer so that the appropriate engineering trade-offs can be made. The following parameters are important for defining a high-speed via:

- **Plane Clearance:** Plane clearance⁴ defines the diameter of a circle that must be cleared of copper surrounding the via so that isolation is maintained when going through a power or ground plane layer or through a copper pour on a signal layer. The default size that guarantees no unintentional shorts due to manufacturing tolerances between the via and other signals may be too capacitive for a high-speed via. A 3D EM simulation provides the best method for optimizing the plane clearance for high-speed performance.
- **Pad Size:** Pad size refers to a disk of copper that is required on a given layer to ensure that even for the minimum and maximum drill location tolerances a connection will be made with the via. The pad is required on the top and bottom of the PCB as well as any internal connecting signal layer. Sometimes a pad is added on all layers that the signal via crosses as a default in the layout tool even if there is no signal trace to be connected. These types of pads are called nonfunctional pads and should be removed for high-speed designs [19] to increase performance; however, it is best to confirm with the PCB fabrication house to make sure that there are no manufacturing trade-offs for a given stack-up.

⁴Sometimes called antipad by the PCB layout tools.

- **Drilled Hole Size:** This corresponds to the diameter of the drill used to make the via hole, prior to plating. Due to the skin effect, high-speed signals will see this larger outside diameter of the via hole for determining the impedance.
- **Finished Hole Size:** This corresponds to the inside diameter of the via hole after the plating process. Finished hole size (FHS) has been an industry standard since it is easy to measure with a feeler gauge at final inspection for PCB acceptance.
- **Return vias⁵:** When a signal follows a via between layers, its return current signal must also find a way to change ground layers. In high-speed digital applications, return vias connecting the reference planes on each layer are added close to the signal via to provide this return current path and set the desired impedance. The number of these vias and their location can be optimized through 3D EM simulation.
- **Backdrilling:** Backdrilling allows the removal of any stub that remains at the signal via after it connects to the intended signal layer. This controlled depth drilling process is done after lamination along with any other unplated drills. If the via stub is significantly less than the rise time of the high-speed signal then backdrilling may not be required.
- **Blind Via:** This is a via that stops at the connecting signal layer without the creation of any stub on the signal path. Blind vias on ATE test fixtures are typically done with sequential lamination steps.

A cross-section of the PCB can be used to verify the via structure dimensions as shown in Figure 8.25 for the case of a backdrilled signal via with a return via on either side. It is important to understand that the transitioning of the return currents on the ground layer are just as important as the signal via at high frequencies and it is the relation between the signal and return vias that determines the high frequency performance. A detailed discussion on the problems with discontinuities in the ground current path can be found in the reference literature [7].

An incorrect via design can significantly degrade a high-speed digital signal. Figure 8.26 shows two different via designs on an evaluation ATE test fixture. One of the via designs is a standard through via with no adjacent return vias while the other via design has been optimized for the PCB stack-up and includes four return current ground vias to maintain a 50Ω impedance topology. To evaluate each via, a stimulus data signal is sent through four

⁵Sometimes they are called shorting vias or ground vias to indicate the shorting connection to the ground plane layers.

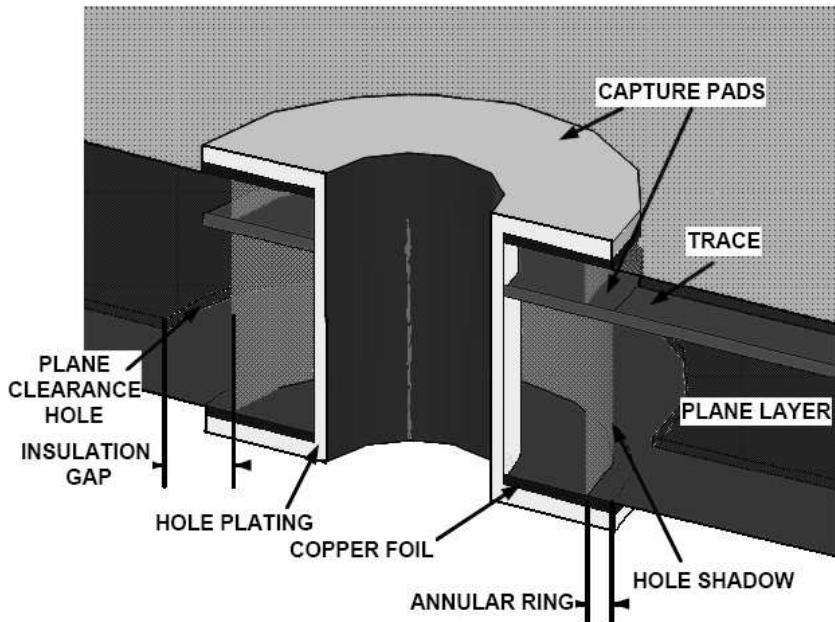


Figure 8.24 Diagram of a typical via with the respective important parameters. (*From: [3]. ©2003 Lee W. Ritchey. Reprinted with permission.*)

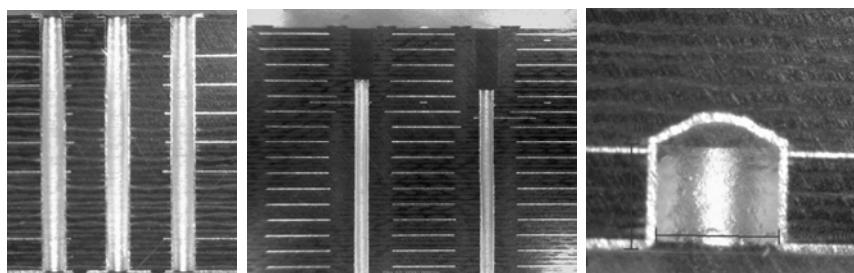


Figure 8.25 Micro photograph of a PCB cross-section showing a thru-hole signal via with adjacent return vias (left), a backdrilled via (center), and a controlled depth (blind) via on the right (courtesy of RD Circuits).

consecutive vias with the same design to look at a worst case signal path with multiple vias.

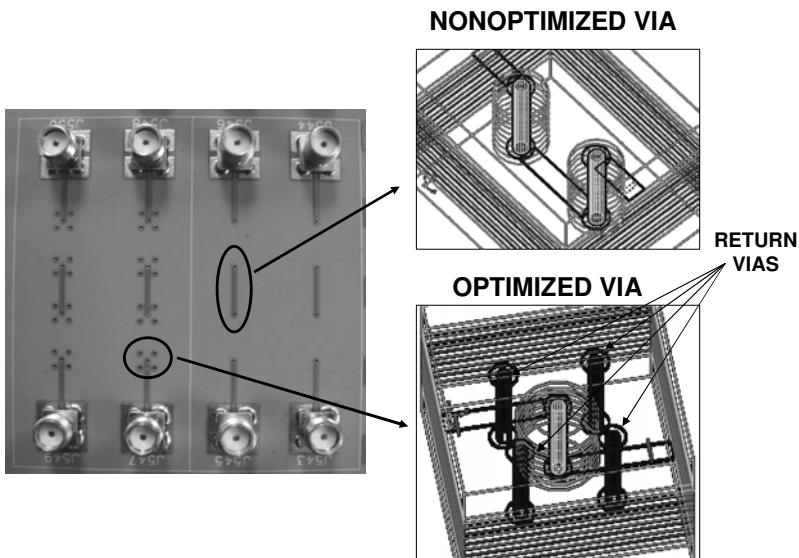


Figure 8.26 Test fixture for evaluation and demonstration of two different via designs (optimized and nonoptimized).

Figure 8.27 shows the frequency domain bandwidth of each of the via designs and the data eye diagrams obtained at 3.25 Gbps. The data eye clearly shows a significant degradation for the signal going through the four non-optimized vias and in the frequency domain it shows that above 1.5 GHz the signal has trouble getting through just a single via to transition between layers. Another parameter that easily might degrade the high frequency performance is the remaining length of the via stub after a connection to the signal layer. The physical length of this via stub can resonate at multiples of the quarter wavelength to create significant signal degradation.

It is important to understand the effect of a via stub at high data rates. Figure 8.28 shows a cross-section of two vias with different stub lengths, and the appropriate S-parameters prove that the short via with long stub starts to resonate in the frequency domain already at 4 GHz. The long via with short stub performs significantly better. The quarter wavelength resonance is a definite limit for the via stub length, but already for smaller values the added capacitance of the stub and its relation to traces, ground, and power planes on other layers can easily degrade the high frequency performance. The 3D EM simulations are useful in understanding the impact of the via stub on the

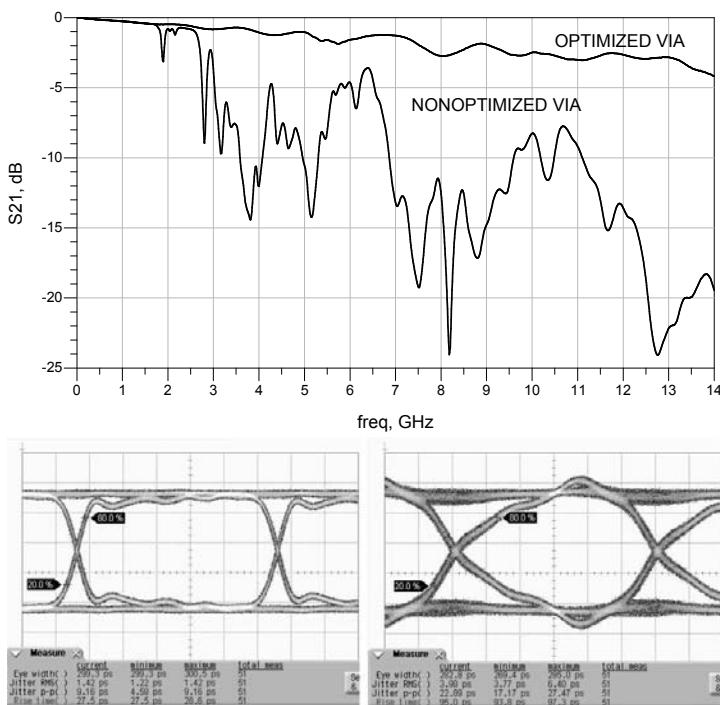


Figure 8.27 Performance comparison of a series of four via transitions for an optimized and nonoptimized via design in the frequency domain (top) and on the time domain (bottom) using the data eye diagram (PRBS $2^{31} - 1$ data pattern at 3.25 Gbps with an input rise time of 12 ps).

impedance profile as well as any probable radiation coupling into other layers of the PCB.

8.4.1 Interlayer Vias

The design of an interlayer via transition should not be blindly leveraged from one design to the next. Efforts to minimize the number of vias in a PCB for cost or routing densities may result in a single ground return via per signal via which may work for one application but in another the crosstalk or signal losses may be too high. Designs using four surrounding ground vias can have a significant advantage when leveraging from one design to another. Designing with two return vias on either side of the signal path entering a via transition and then two more as the signal exits help to contain the fields as they make the right angle via transitions and limit the amount of signal loss caused by coupling to other layers of the PCB. This type of via is shown in Figure 8.29.

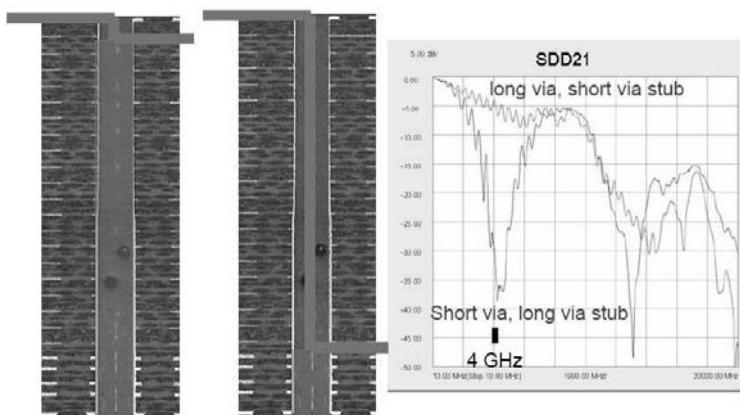


Figure 8.28 Cross-section of two vias, one with a long stub (left) and one with a short stub (right), with their measured differential insertion loss. (From: [20]. ©2009 Mike Resso. Reprinted with permission.)

By limiting the interaction with neighboring layers, the via is less sensitive to variations in the stack-up from one design to the next. The shielding provided by the return via structure reduces crosstalk between adjacent signals. The surrounding four return current vias also have the advantage of forcing the layout designer to keep other signal traces at a safe distance from the signal via, further minimizing crosstalk problems. The four ground via topology is also symmetrical and the connection to the via therefore well defined.

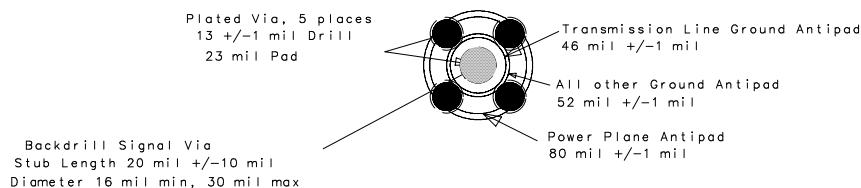


Figure 8.29 Example of an optimized via using four surrounding ground vias.

8.4.2 Pogo Pin Vias

ATE systems that use an array of pogo pins, like the assembly shown in Figure 8.30, will require a specialized “pogo via” to provide the best signal integrity when transitioning from the ATE coaxial cables to the PCB transmission line on the DUT test fixture. Pogo pin assemblies can handle very high data rates when properly designed, but the pogo pin is only half of the transition and the mating via and transition design on the test fixture can easily be a source

of significant signal degradation. This creates a challenge when specifying “at-the-pin” performance of an ATE system since the test fixture is under the control of the test fixture designer and not the ATE manufacturer. To achieve the best possible performance the test engineer can utilize 3D EM modeling to optimize the transition design as shown in Figure 8.31, or rely on the generalized transitions provided by the ATE manufacturer. The ATE manufacturer has a significant interest in designing a test fixture to measure the highest quality “at-the-pin” performance and is less concerned with complexity and fabrication costs. The test engineer, however, must balance performance with cost as well as design risk when deviating from a verified topology.

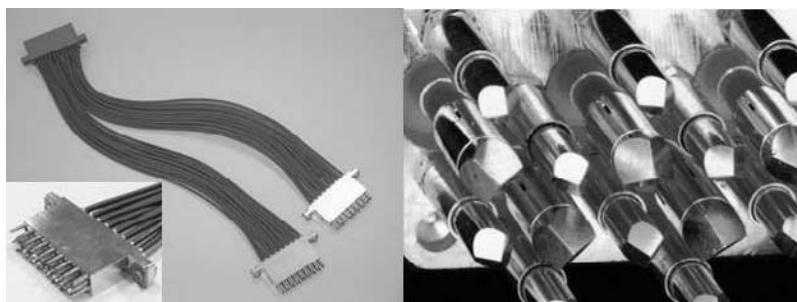


Figure 8.30 Example of a pogo pin assembly for an ATE system (courtesy of Verigry).

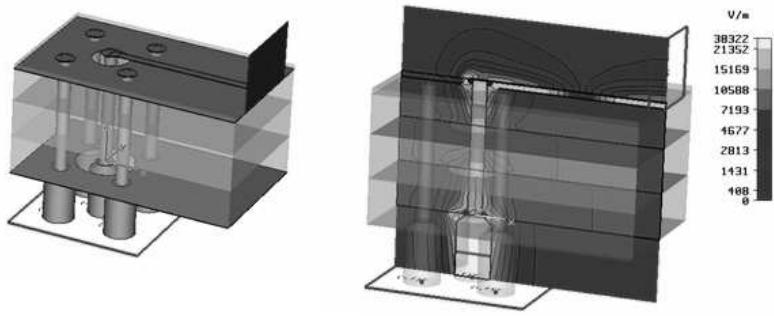


Figure 8.31 3D model of a pogo pin and pogo via for EM simulation.

The optimization of the pogo via transition can leverage various PCB via drilling processes to maximize performance. Utilization of controlled depth drilling for the pogo via contacts ensures a robust multipoint contact while at the same time freeing up the location of the through ground vias and the drill diameter of the signal via so that performance can be optimized. Figure

8.32 shows a cross-section of a pogo via that was the result of optimizing the various drilling features and ground clearances using 3D EM simulations that are verified and further optimized with measured data. More details on this type of approach for pogo via design can be obtained in [16, 21]. The ATE manufacturer may have a variety of optimized pogo via transition designs depending on the interfacing pin electronics card and the performance range. This clearly shows that the test engineer must work closely with the ATE manufacturer to select the best pogo via transition for a given application.

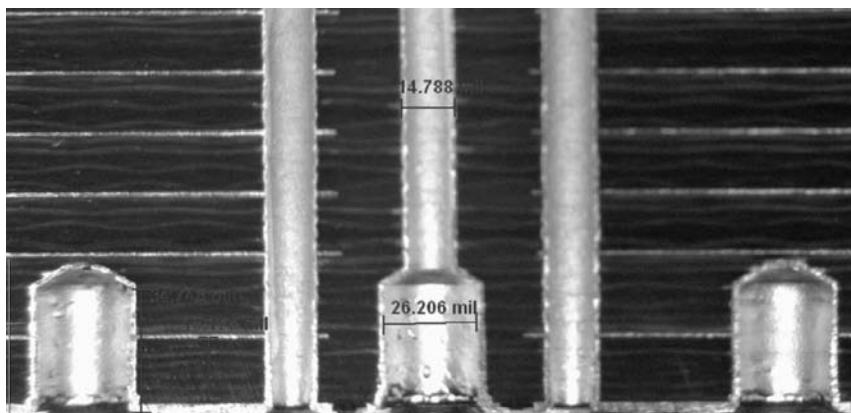


Figure 8.32 Cross-section photograph of the manufactured pogo via for high-speed digital applications (courtesy of RD Circuits).

8.5 DUT BGA Ballout

The DUT pin connections or ballout of a ball grid array (BGA) device are defined by the DUT package. The ATE test fixture cannot change the ballout of a device and must instead work with the topology of the BGA power, ground, and signal pin locations to provide the best performance possible. One of the first challenges is in routing signals through the BGA via field to reach the desired signal pad. The designer will find that for BGA pitches of 1 mm and smaller the allowable trace width can quickly push one to the limits of the PCB technology with 0.125-mm trace widths or less. To guarantee the specified transmission line impedance all the way to where the trace connects to the BGA via requires that this small trace width is used for the entire signal path between the ATE pogo via and the BGA via. However, this small trace width will have higher resistive and skin effect losses that must be considered when routing over the long distances found on ATE test fixtures and it will be more sensitive to etching tolerances that cause impedance variations. The test

engineer must decide whether a wider trace and an impedance discontinuity when necking down into the BGA via field, as shown in Figure 8.33, has higher performance than a single narrow trace routed into the BGA via field.

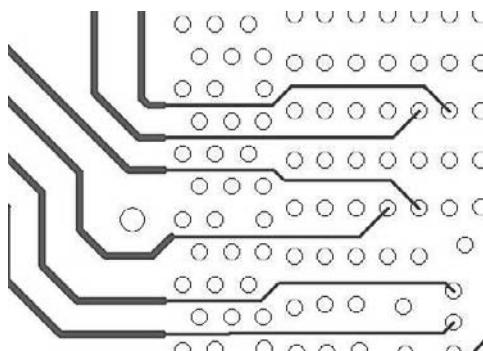


Figure 8.33 Addressing the routing challenges on a tight-pitch BGA through the reduction of the trace width (necking) on the BGA area.

As usual the right answer will depend on the geometry of the specific application (i.e., how long is the signal trace, how large is the impedance discontinuity, and how long is the routing in the BGA via field). These parameters are shown in Figure 8.34 where L_T is the length of the signal trace before entering the BGA array area, W_T is the trace width that guarantees the $50\ \Omega$ impedance, L_N is the trace length where the signal trace width is reduced (necked), and L_B is the trace length with the reduced width of W_B until the BGA pad.

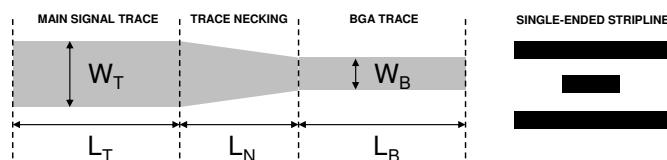


Figure 8.34 Signal trace necking parameters.

The challenge is to find the optimal value of W_T and W_B since the values of L_T and L_B are fixed from the layout requirements. L_N allows for some optimization but it is not a major factor. Figure 8.35 shows the data eye diagrams from one example of a signal trace with $L_T=26\text{ cm}$ (10.2 in), $L_B=15\text{ mm}$ (600 mil), and $L_N=2\text{ mm}$ (80 mil) comparing a geometry of $W_T=381\text{ }\mu\text{m}$ (15 mil) and $W_B=101.6\text{ }\mu\text{m}$ (4 mil) with a geometry of $W_T=215.9\text{ }\mu\text{m}$ (8.5 mil) and $W_B=101.6\text{ }\mu\text{m}$ (4 mil). The data shows that for an ATE test fixture with long routing distances one should consider wider trace widths

with trace necking. Measured trace loss data along with a transmission line simulator make it possible to quantify the losses for different layout topologies as shown in Figure 8.36.

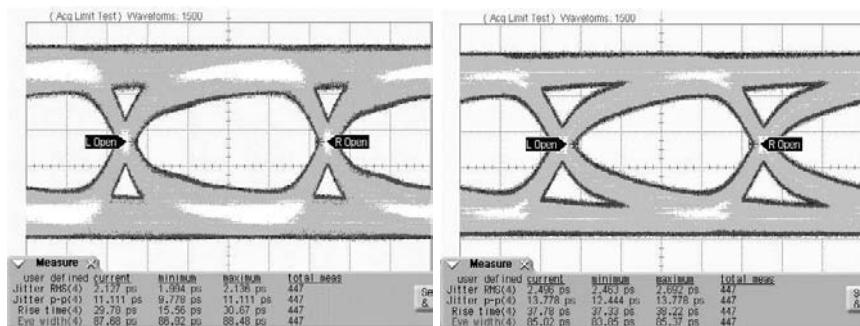


Figure 8.35 Example of a trace necking optimization with the data eye on the left obtained with a trace necking from 381 µm (15 mil) to 101.6 µm (4 mil) and on the right from 216 µm (8.5 mil) to 101.6 µm (4 mil) on a 26-cm (10.2 in) trace. (From: [16]. ©2006 Jose Moreira. Reprinted with permission.)

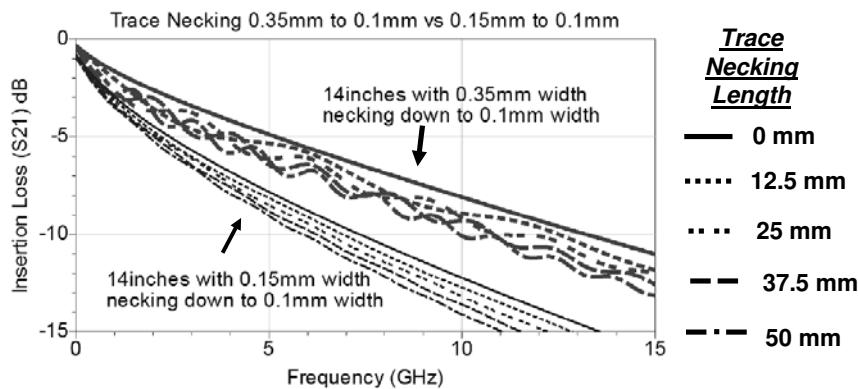


Figure 8.36 Loss versus frequency data for decreasing trace necking impedance discontinuity. The 50 Ω trace width is varied from 381 µm (15 mil) down to 152.4 µm (6 mil), and the routing distance of the 100 µm (4 mil) is varied from 0 to 50 mm (2 in).

Another PCB layout item related to the BGA ballout is the implementation of the via from the stripline where the signal arrives from the ATE pin electronics to the pad on the top of the PCB test fixture. Two options are possible. One is to make the via very close to the pad position and then connect the via to the pad forming what is called a dog bone connection (see Figure

8.37). Another option is to have the via directly in the pad. This approach shown also in Figure 8.37 requires that the via is filled with epoxy so that it is possible to have a flat solid pad on top of the via. Although more expensive this is the recommended solution for high-speed digital applications since it has a straighter signal path and less potential for crosstalk.

Note that it is not necessary to use a conductive epoxy like a silver filled one since due to the skin effect we know that at higher frequencies the signal will only flow on the edge of the via. By keeping the via core nonconductive, we are already providing some minimal equalization by forcing the signal at low frequencies to also flow on the edge of the via and have similar resistive losses. The other concern might be that for power distribution the vias need conductive silver filling but it is important to understand that a conductive epoxy uses silver "flakes" to create a conductive path and is not solid metal, while the copper plating of the via barrel is solid and provides the lowest resistance path. Although conductive filling does not hurt, in the case of large ATE test fixtures any increase in plating thickness and copper planes for heat spreading is more effective for thermal cooling than a conductive via fill.

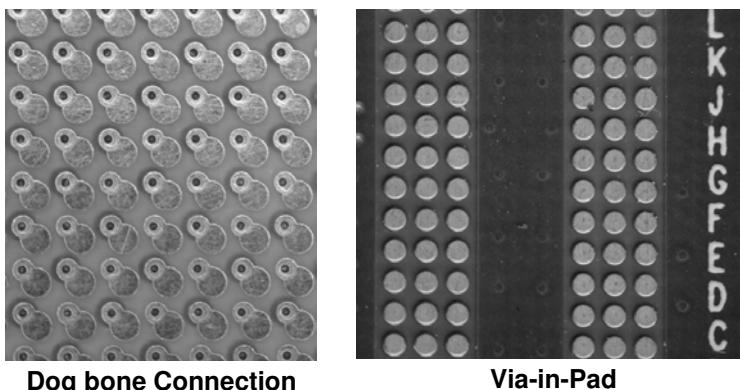


Figure 8.37 Comparison of a via in the pad approach with the "dog bone" type approach.

The physical dimensions of the DUT package ballout are not the only constraints for the performance of the test fixture design; it is also important to note that the performance of a given I/O connection will depend on crosstalk from the neighboring signals and the location of the reference pads (ground and power). The distribution of signal, power, and ground pins may work well for a DUT on a thin application PCB without a socket, but the longer path of the ATE via and socket could exacerbate any performance issues of the I/O due to a nonideal distribution of surrounding reference pads.

8.6 Sockets

The test fixture socket corresponds to the “last mile” of the signal path from the pin electronics to the packaged DUT and is no less important. Historically the key design goal of a socket was mechanical reliability in the form of repeatable low resistance DC connections. At high speeds one must also consider the impedance discontinuity determined by the socket materials, contactor dimensions, and if the high frequency signal path of the contactor in the compressed position is repeatable. Production test socket requirements are often at odds with high-speed performance since one of the simplest ways to improve high-speed performance is to make the socket as thin as possible which leaves little room for mechanical compliance. Reference [22] provides an introduction to IC sockets and Figure 8.38 shows examples of different sockets types.

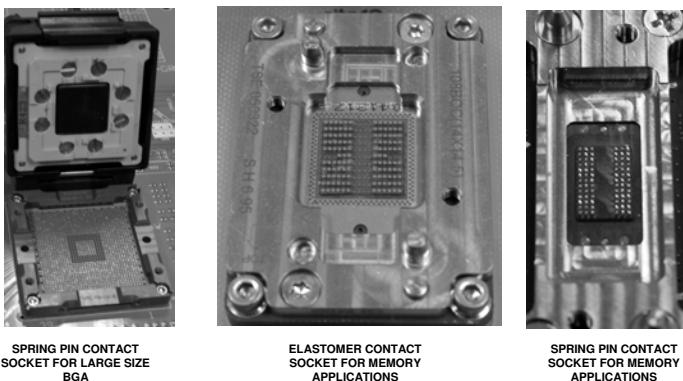


Figure 8.38 Examples of different sockets for BGA packages.

The socket is the most stressed element of a test fixture. The ATE test fixture might be docked to an ATE test head two to ten times per day, while in one day a socket might have dozens of IC insertions for characterization/design verification or several thousands of IC insertions for production testing. Although it is expected that a socket should be repairable or replaceable, any reduction in the lifetime or reliability of the socket directly impacts cost-of-test [23].

A socket for a test and measurement application can be divided in three parts: the socket housing, the contactor, and the socket lid as shown in Figure 8.39. The socket housing is typically built from a polymer material like polyethylene terephthalate (PET) selected for mechanical stability and precision machining properties. The socket lid provides a way to correctly push the DUT package into the socket contacts and keep it there. The socket

lid will usually contain a heatsink and sometimes a fan for temperature control of the DUT, although in some cases more complex cooling approaches are necessary. But for applications requiring automatic DUT package insertion in the socket through a robotic handler, the socket will not have a lid and in this case uses an open top structure with the DUT package being “pushed” into the socket by the handler plunger. For BGA type packages that correspond to the majority of high-speed digital applications, Table 8.4 presents a list of available contact technologies with the spring pin type contact being the dominant approach in sockets for high-speed digital ATE test fixtures [22].

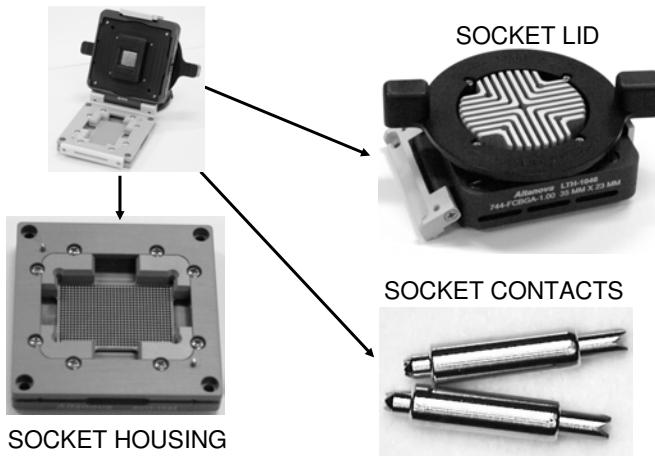


Figure 8.39 Breakup of a typical socket into its individual parts (courtesy of Altanova).

8.6.1 Socket Electrical Characterization

Evaluating the electrical performance of a socket is a complex task. Socket specifications are defined in a very specific framework that usually does not correspond to the environment where the socket is used on a test fixture. Table 8.5 shows one example of a manufacturer's specification for a socket.

The first item described in the specification is the contact resistance. This is an important parameter especially for DC type tests. The challenge with contact resistance is that one desires it to be very low and not changing even after a large number of insertions. This usually conflicts with the requirement to use very low resistance materials (especially in the plating of spring pin type sockets) since low resistance is typically found in higher purity softer metals that do not have the mechanical properties to allow the socket to handle a large number of insertions. On the other side, there are alloys that can withstand a very large number of insertions but have a higher contact resistance value. It

Table 8.4
Comparison of Contact Technologies for BGA Sockets

Contact Technology	Advantages	Disadvantages
Spring pin contact	High-bandwidth, high compliance	Spring contact high frequency path length
Cantilever spring contact	Conventional technology, low force relaxation, replaceable contact	Limited pitch capability, long contact path, contacting solder ball bottom, no penetration action, z-axis loading
Fuzz button contact	Replaceable contact, short electrical path, high frequency application	No wipe action due to straight compression, no penetration action, high normal force, contact force relaxation, z-axis loading
Z-axis conductor in elastomer contact	Fine pitch capability, minimal socket thickness	No single contact replacement, elastomeric creep or stress relaxation, solder ball bottom contact, z-axis loading
Tweezer contact	Penetration through the oxide layer, double-sided contact, horizontal contact force	Long electrical path
Crown contact	Multipoint contact, fine pitch capability	Z-axis loading, added height with fuzz buttons and elastomeric

Table 8.5
Example of a Manufacturer Specification for Socket with a Spring Type Contactor

Electrical Specifications	Value
Contact resistance	18.62 mΩ
Self inductance	0.6 nH
Insertion loss	< 1 dB to 10 GHz
Pin to pin capacitance	0.075 pF
Current carrying capability	3 A

is then important when looking into the contact resistance value for a specific socket to understand how it will change with the number of insertions.

Self-inductance is also an important value since it represents the physical dimensions of the socket and provides important data to understand how the socket will impact items like the power distribution to the DUT (see Section 8.11.3). It is important to note that self-inductance is usually measured in a very specific configuration (e.g., four ground pins surrounding the socket pin being measured) that might not correspond to the pin configuration in the final application.

The pin-to-pin capacitance specification allows a judgement of the capacitive coupling between the pins in the socket which can be directly related to the possible crosstalk between the pins. However, a 3D EM field solver is needed to accurately predict both the capacitive and inductive coupling in the socket for a given DUT ballout.

The insertion loss specification provides an idea of the performance of the socket in terms of its frequency range. Note again that this specification might be based on a very specific pin configuration that may not represent the final application BGA ballout. The maximum current specification indicates the upper limit of the continuous current the socket can handle.

8.7 Relays

Relays are probably the most used component on test fixtures to extend the capabilities of an ATE system [24]. One very typical application is to use relays to multiplex the ATE resources by allowing several DUT channels to be tested by a single ATE channel as shown in Figure 8.40.

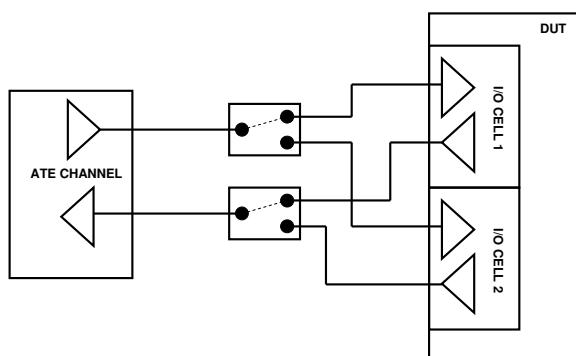


Figure 8.40 Using a relay to duplicate the number of DUT channels being measured by a single ATE channel.

Of course, there is a penalty in test time and also on signal integrity, but depending on the application requirements (e.g., test equipment cost is a major factor), it might be a good alternative. Another usage could be the ability to change between an ATE measurement setup and a loopback test setup as shown in Figure 8.41.

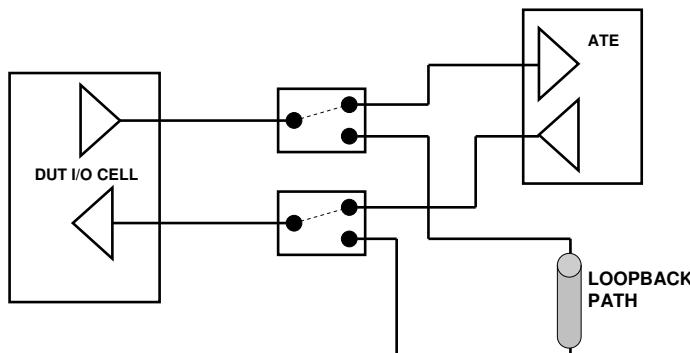


Figure 8.41 Using a relay to switch between an ATE measurement setup and a loopback test setup.

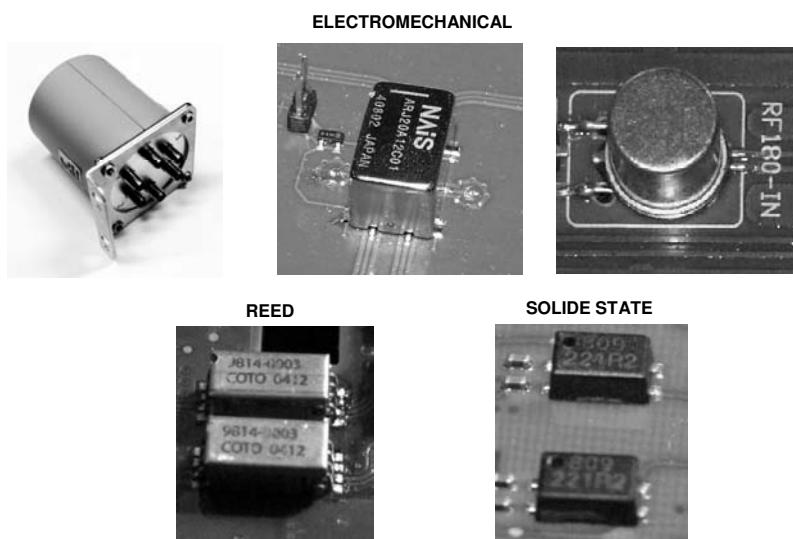


Figure 8.42 Pictures of different relay families.

Figure 8.42 shows pictures of different types of relays and Table 8.6 shows a comparison of various relay technologies. Electromechanical relays have historically provided a robust low loss signal path with stable

performance over temperature and time but can be expensive, larger in size, and have a lower reliability than other families such as solid-state switches. Solid-state switches have several disadvantages. They have higher losses, temperature dependencies, broadband nonlinearities, and a very limited DC common mode range. REED relays from a performance point of view are typically worse than mechanical relays due to the stub of the open path and the capacitance over the open contact. Their advantages are price, reliability (statistically better than mechanical but can have a higher percentage of early failures), and typically a smaller footprint than mechanical relays. MEMS based relays have received much attention lately but this relay family is still struggling with reliable long-term contacts. MEMS-based relays can be fragile, highly susceptible to contamination, and cannot support hot switching which is a significant issue when being considered for test and measurement applications. The potential for extremely small size and high frequency performance of the MEMS relay technology continues to drive significant research and investigation but for now the traditional methods still provide the necessary bandwidth for high-speed I/O applications.

Table 8.6
Relay Families

Relay Type	Advantages	Disadvantages
Nonhermetic Electromechanical: Switching by mechanical means	Repeatability, bandwidth	Size, cost, reliability, life expectancy
Hermetic Electromechanical: Switching by mechanical means	Repeatability, bandwidth, reliability	Size, cost, life expectancy
Solid-State: Switching by a FET switch or PIN diode	Reliability, size, cost	Trade off between DC common mode range and performance
Mechanical REED: Switching by mechanical means in a glass casing	Long life, size, cost	Bandwidth with open stub capacitance in a SPDT configuration, reliability
MEMS: Switching by electrostatic mechanical means	Size, bandwidth	Fragile, hot switching concerns, bias circuitry

Relays are also classified by the number of ports and their functionality.

Figure 8.43 shows the commonly used nomenclature for the description of the relay functionality and the associated schematic. For REED type relays a different nomenclature is sometimes used as shown in Figure 8.44.

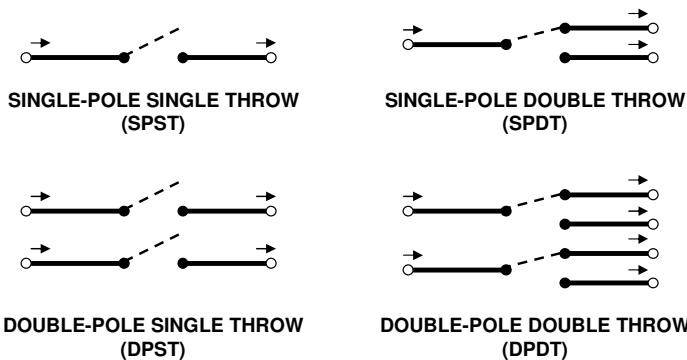


Figure 8.43 Nomenclature and schematics used to describe the different types of relay configurations.

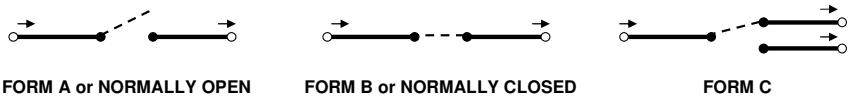


Figure 8.44 Nomenclature and schematics used for REED type relays.

One important question when selecting relays for DUT test fixtures is the influence of the relay on the overall measurement performance. Although proper choice of the relay model/family is important (i.e., choosing the relay with the required bandwidth for the intended data rate), the PCB footprint is also of critical importance. Typically relay vendors will provide a suggested relay footprint but unfortunately this footprint is in most cases optimized for an RF type application (i.e., requires high isolation between ports) and uses microstrip trace design. Usually in test fixture design for high-speed digital applications stripline routing is used and high isolation is not needed and it is preferred to utilize a minimum number of ground vias for improved routing space. Figure 8.45 shows one example of an optimized PCB footprint for a mechanical relay where the optimized transition vias from the stripline to the relay and back are also included in the footprint.

Figure 8.46 shows the measured insertion loss comparison of a test fixture signal path with and without a relay for a 10-Gbps application. Although a state-of-the-art relay was chosen with an optimized footprint, there is a clear effect on the insertion loss.

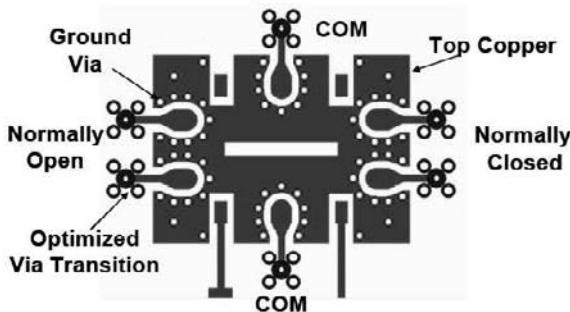


Figure 8.45 Footprint example for a mechanical relay.

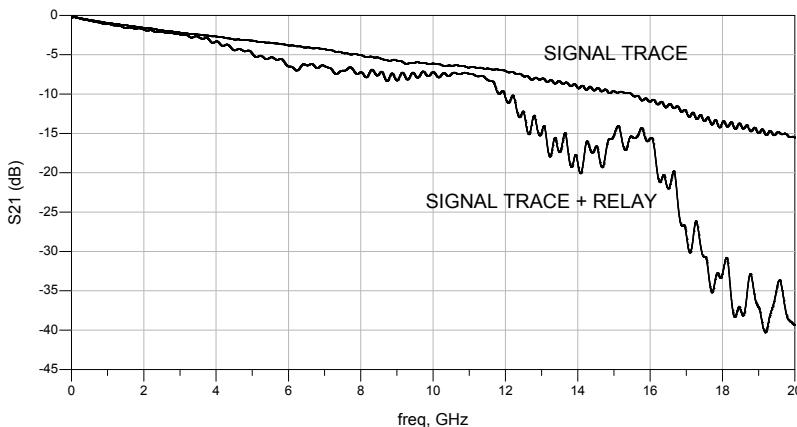


Figure 8.46 Insertion loss comparison of a signal path with and without a high-frequency relay [34 cm (13.4 in) 482.6 μ m (19 mil) stripline in Rogers 4350 with a Matsushita NAIS ARJ relay].

The influence of the relay was also measured in the time domain. Figure 8.47 shows a comparison of the measured data eye diagram of the signal path with and without the relay. One can observe a degradation on the rise time but the jitter is very similar, showing that in this example the relay choice together with an optimized footprint provided an acceptable solution for the application.

In conclusion, it is possible to use relays on DUT test fixtures for high-speed digital applications but it is important that in conjunction with the appropriate relay choice special attention is given to the optimization of the PCB footprint for the relay.

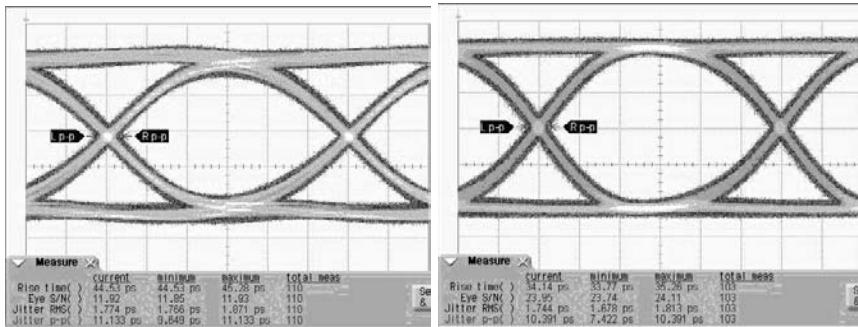


Figure 8.47 Data eye comparison of a signal path with (left) and without (right) a high-frequency relay [34 cm (13.4 in) long, 482.6 μ m (19 mil) wide stripline in Rogers 4350 with a Matsushita NAIS ARJ relay].

8.8 Bidirectional Layout

Bidirectional interfaces can require special layout techniques depending on the capabilities of the ATE pin electronics. If the pin electronics has inherent support for bidirectional interfaces, then the layout for a bidirectional interface presents no additional challenges compared to a unidirectional interface. In the case where the pin electronics does not have an inherent support for bidirectional interfaces, it is necessary to use special techniques for their test [25]. For the test fixture layout, these techniques might imply the need to connect the ATE driver and receiver to the DUT I/O. The simplest option is the so-called dual transmission line or fly-by approach where the ATE driver and receiver are connected to the DUT at a point very close to the DUT as shown in Figure 8.48.

This approach introduces a discontinuity at the point where the ATE driver and receiver are connected together. To reduce the length of this discontinuity to a minimum, one possibility is to connect the ATE driver to the receiver at the socket via that goes to the DUT. This can be achieved with minimal coupling between driver and receiver signals if the ATE driver and receiver traces are on different signal layers as shown in Figure 8.49.

Another option to address the discontinuity of connecting the ATE driver to the receiver is to use a power divider (Section 7.11.1) to maintain a $50\ \Omega$ impedance through the entire signal as shown in Figure 8.50.

Although this approach solves the discontinuity problem, it raises other challenges due to the need to add a surface mount component (the power combiner) which adds additional transitions and by design reduces the signal power by half. In memory applications, one could argue that when the DUT is receiving, then the fly-by connection is really just a long transmission line

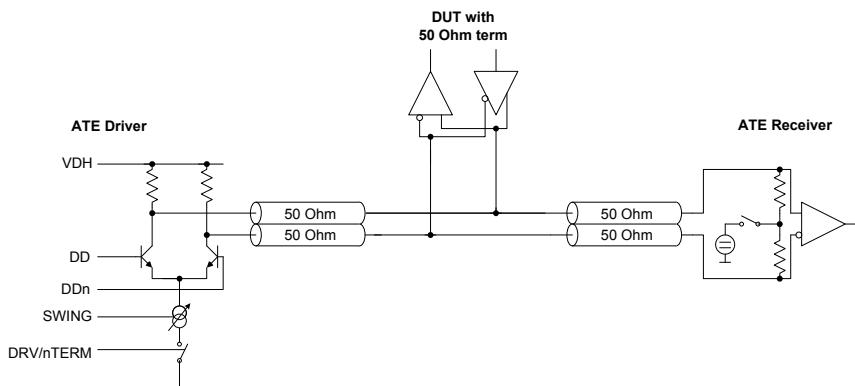


Figure 8.48 Bidirectional interface to unidirectional ATE channels connection diagram using a dual transmission line (fly-by) approach.

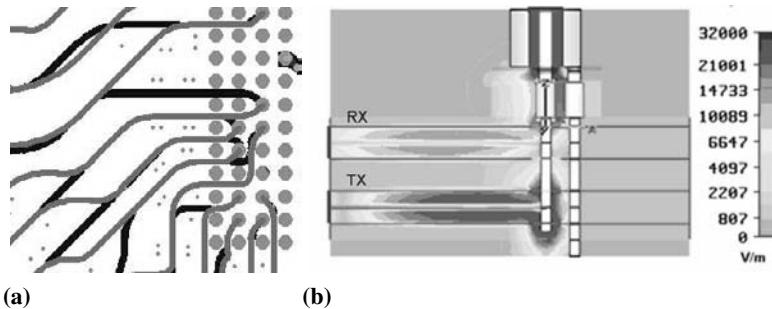


Figure 8.49 (a) Example of a bidirectional interface layout (the driver and receiver lines are on different layers and they join on the via below the I/O package pad) and (b) a 2D cross-section of a 3D EM simulation.

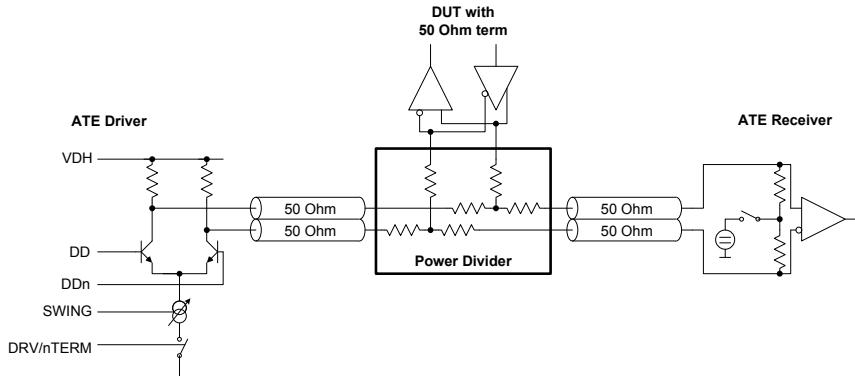


Figure 8.50 Bidirectional interface to unidirectional ATE channels connection diagram using a dual transmission line with a power divider.

with a high impedance DUT receiver tapping into the center with minimal impact on the signal integrity. When the memory device is transmitting, the transmitter impedance is actually lower than $50\ \Omega$ and may find it better matched with the two fly-by transmission lines in parallel having a total of $25\ \Omega$ impedance as seen from the DUT. Additional comments regarding the application side of this approach are discussed in Section 9.4.

8.9 Wafer Probing

Wafer probing is such a large and important topic that it cannot be properly addressed in this short section. One good source for up-to-date information is the yearly IEEE Semiconductor Wafer Test Workshop [26] which is dedicated to semiconductor wafer testing. Wafer probing is part of almost every production test-flow [27] but in this section we will only address wafer probing for at-speed digital testing.

At-speed wafer probing has been in use for some time in the ATE industry even at data rates of 10 Gbps, although usually only for devices with limited pin count, in the wired communication area. An example is shown in Figure 8.51. The increased cost of packages and the need to provide known good die (KGD) for multichip module (MCM) integration has raised the need for KGD testing. In KGD testing the die must be fully tested for failures which usually implies not only the standard DC type tests but also at-speed functional testing [28].

A typical test fixture for wafer probing is composed of three elements as shown in Figure 8.52. A wafer probe interface PCB connects the ATE pin electronics to the probe pogo tower. The pogo tower connects the wafer probe interface to the probe card. The probe card is a PCB board that contains the probes which connect the DUT on the center of the probing interface. This type of configuration can typically accommodate a new application or device with just a redesign of the probe card. The wafer probe interface and the pogo tower are normally application independent and designed by the ATE manufacturer since they are only dependent on the specifics of the ATE platform. Figure 8.53 shows an example of a vertical probe card for a high-speed digital application.

An alternative solution for the wafer probing test fixture design is to connect the probe interface directly to the test fixture as shown in Figure 8.54. The three separate interface parts (probe interface PCB, pogo tower, and the probe card) are substituted by a single PCB test fixture. The advantages of this approach is increased space available for items like relay or support circuitry, improved bandwidth, and the overall setup is closer to the final test fixture design for a packaged DUT.

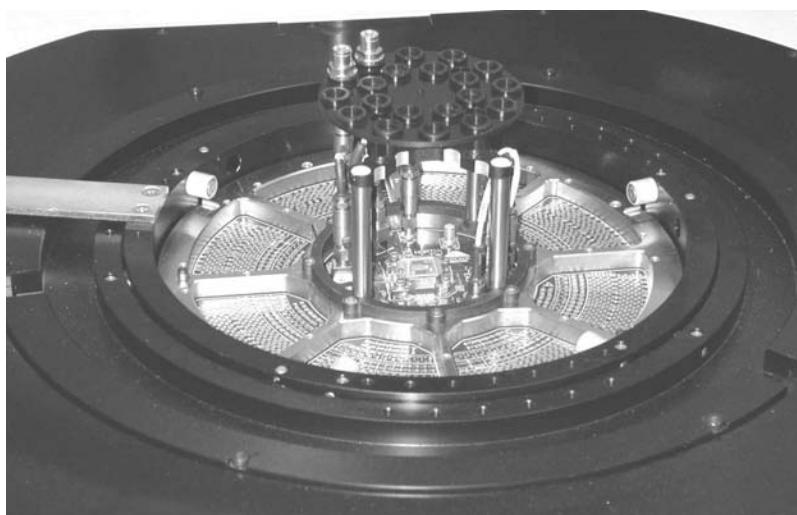


Figure 8.51 Probe card connected to a wafer prober for 10-Gbps at-speed wafer testing. (From: [29]. ©2005 Jose Moreira. Reprinted with permission.)

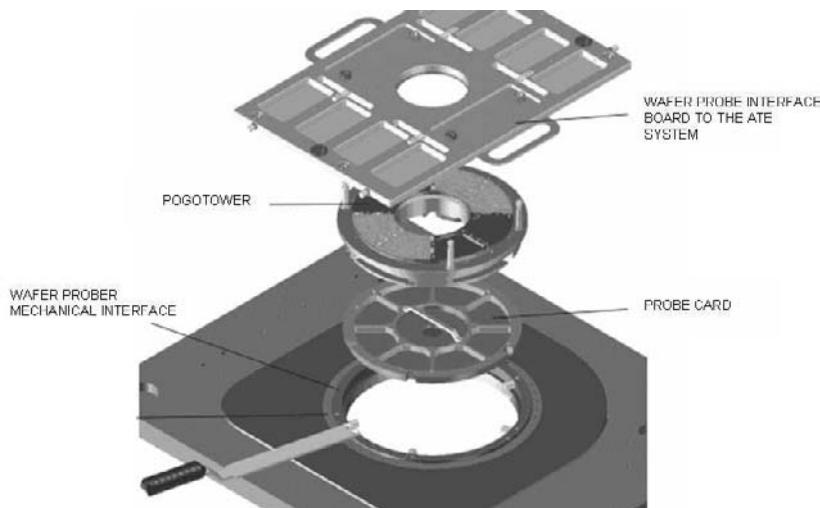


Figure 8.52 A typical standard wafer probing test fixture setup (courtesy of Verigy).

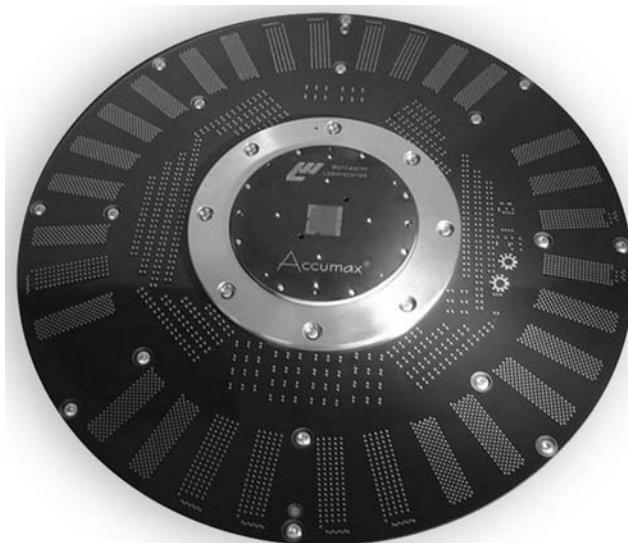


Figure 8.53 Vertical probe card (Accumax probe card courtesy of Wentworth Laboratories, Inc.).

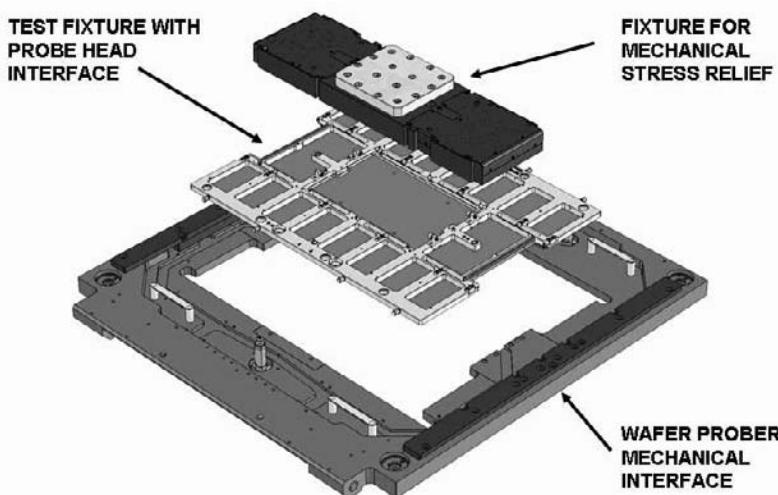


Figure 8.54 A wafer probing test fixture with direct docking of the wafer prober to the ATE interface PCB to eliminate the need for a pogo tower and probe card (courtesy of Verigy).

For the die probing interface itself there exist different probing technologies that can be used [30]. Table 8.7 describes some of the main wafer probing technologies and Figure 8.56 shows pictures of the wafer probes in those technologies. Note that there are several other technologies like blade, pellicle, and elastomeric hybrids. that are not described in this section. Although the 6-mm length of a typical vertical probe appears to be less than ideal for high speed, simulations actually show that the pin diameter and the distance to the neighboring pin can provide a reasonable impedance match for high-speed multigigabit signals. Figure 8.55 shows the results of 3D EM simulations and the excess inductance when used in a $50\ \Omega$ environment. The excess inductance is 0.58 nH for the single ended $50\ \Omega$ simulation, but due to coupling the differential impedance of two adjacent pins actually has 0.17 pF of excess capacitance. The large amount of coupling between adjacent pins is also a source of unwanted crosstalk which can be reduced by going to a membrane probe or modifying the test sequence to minimize adjacent channel noise.

Another important topic related to wafer probing is probe tip cleaning. This is a complex topic that is dependent on the probing technology used. The lack of proper probe tip cleaning will result in reduced measurement performance and will cause yield loss in a production setup. On the other hand improper cleaning technique might also reduce measurement performance by degrading the probing tips and reduce the probe interface lifetime [30].

When using wafer probing for at-speed digital testing applications there is also the possibility of applying the loopback techniques described in Section 6.4.3.2 on the probe card and in this way use a low-speed ATE system for at-speed I/O testing. The challenge is that for high pin count devices, the space needed for implementing the loopback approach might not be enough especially if DC testing and at-speed testing are to be done with the same probe card.

8.10 Stack-Up

The stack-up is one of the key aspects of a test fixture PCB and needs to be documented in an appropriate way as shown in Figure 8.57. The stack-up describes how the test fixture layers are assembled, which dielectric materials are used, and in the example of Figure 8.57 also shows the drill sizes that are needed for the pogo and interlayer vias. This information is typically found on the “fabrication document” that is included with the CAD files sent to the PCB fabricator.

On signal layers the dielectric height together with the trace width will define the impedance for that layer. For layers corresponding to power

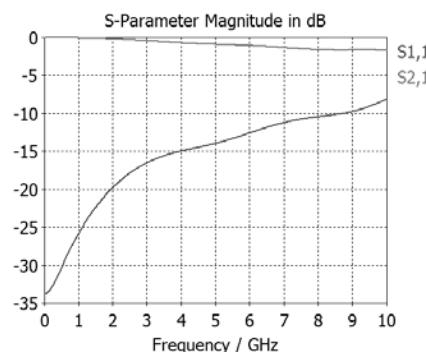
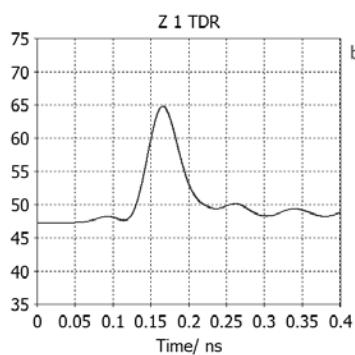
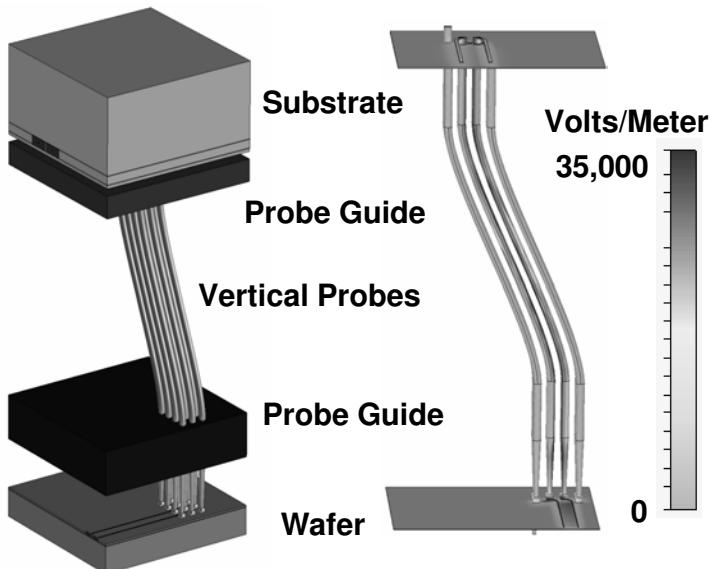


Figure 8.55 Example of a 3D EM model of the ATE test fixture substrate with vertical probe connections to the DUT wafer to demonstrate multigigabit performance. TDR impedance discontinuity shows 0.58 nH of excess inductance but still allows multigigabit signals to pass with minimal high frequency insertion loss.

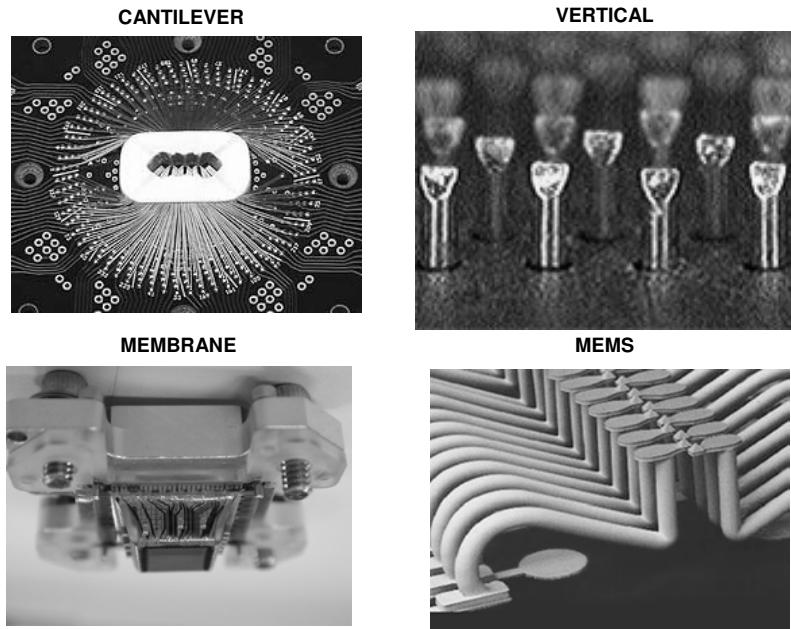
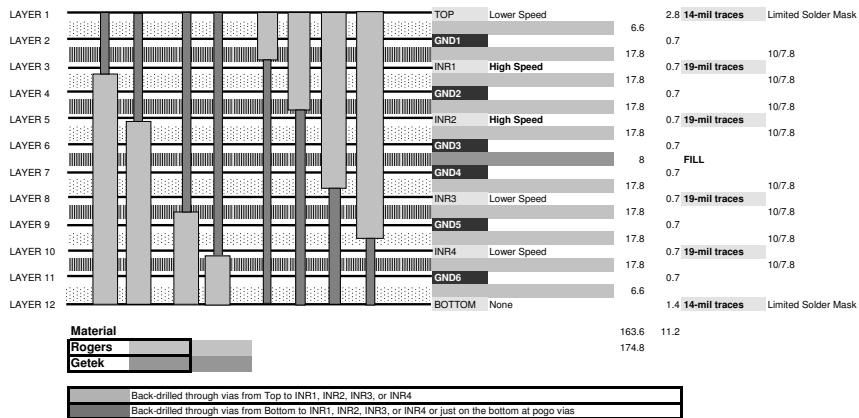


Figure 8.56 Picture of different types of wafer probing technologies (top left: cantilever; top right: vertical; bottom right: membrane; bottom left: MEMS) (FormFactor MicroSpring, photo used with permission of FormFactor, Inc. (c)2009).



- Inner layers are of 0.5oz Cu and outer layers are of 0.5oz Cu with plating.
- Overall board thickness 175 MILS +/- 10%
- Impedance should be 50 ohms with tolerance +/- 5% for inner layers and +/- 10% for outer layers.
- Material is Rogers 4350 for inner signal layers and Getek in the center.
- Tuned trace widths on layers are 19.0 mils or 14 mils.
- 8 stub drill depths - 4 from top to layers INR1, INR2, INR3, and INR4, and 4 from bottom to layers INR1, INR2, INR3, and INR4.

Figure 8.57 Example of a stack-up for a high-speed digital test fixture (courtesy of Altanova).

Table 8.7
Types of Wafer Probing Technologies for High-Speed Digital Applications

Type	Description	BW	Number of Pins
Cantilever	Needle soldered to probe card and held in place through an epoxy ring	< 400 MHz	Several hundred
Vertical probe	Vertical needle that delivers a tangential force at the top of the solder pads	< 5 GHz	Many thousand
Membrane	Photolithography defined metal deposited on a flexible membrane	< 20 GHz	Several hundred
MEMS	Photolithography defined with portions or all of the supporting substrate etched away leaving the structure free	< 5 GHz	Many thousand

planes the dielectric height will define the capacitance and the loop inductance between the power plane(s) and the ground plane(s). One challenge is that due to manufacturing constraints, the maximum stack-up height of a multilayer PCB is limited to around 7.6 mm (300 mil) with the exact value dependent on the manufacturer and yield issues. This creates restrictions on the combination of number of layers and the respective height of each layer which in turn impacts the signal trace loss.

One question that might arise when designing and manufacturing a multilayer test fixture PCB is if for a given multilayer board with several identical layers the signal trace performance in each layer will be identical. Figure 8.58(a) shows an example of a high-speed digital test fixture with identical signal traces in each layer and where the insertion loss of each layer was measured with the result of almost identical graphs of the insertion loss versus frequency for every layer. Although this does not provide a guarantee for any other test fixture PCB, it is the typical result one can expect if using a good PCB manufacturer. Another more complicated question is if the signal trace performance of the exact same PCB design manufactured by two different companies might change. Figure 8.58(b) shows one example where a test fixture was manufactured by two different companies and although the trace loss is very similar, due to a difference in performance on the backdrilling (i.e., remaining via stub length) from one of the fabricators, the

performance at frequencies higher than 23 GHz is very different. This example shows how important it is to reverify a design when changing PCB vendors for a test fixture PCB even if nothing else has changed. To verify the test fixture performance detailed measurements must be made on the manufactured test fixture. Appendix H provides some discussion on different techniques.

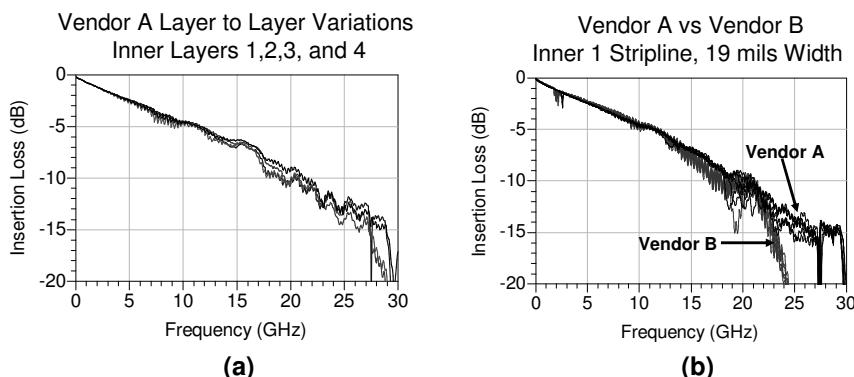


Figure 8.58 Comparison of (a) the insertion loss of identical traces in different layers of the test fixture and (b) the insertion loss comparison of identical traces and layers in two DUT test fixtures manufactured by different PCB fabrication houses.

8.11 Power Distribution Network

Power distribution on an ATE test fixture is a complex topic for which a complete detailed treatment is outside the scope of this book. Proper design of the power distribution network⁶ (PDN) in the ATE test fixture is critical for the performance of high-speed digital devices. Applications like a microprocessor might require peak currents in the hundreds of Amperes, creating significant challenges for the ATE power supplies and the test fixture PDN design [31, 32]. The references [3, 33–35] provide good starting points for a general treatment on this topic.

The PDN design challenges are directly related to the transistor density in the DUT, I/O pin count, data rate, and clock frequency as shown in Figure 8.59. This means that the challenges will continue to increase with the expected move to higher data rates and I/O pin counts in the future.

Noise in the PDN can have a direct impact on the DUT performance by increasing the maximum delay and the delay uncertainty inside the DUT, increasing the DUT jitter [36], degrading the noise margin, and also degrading

⁶Also sometimes referred as the power distribution system (PDS).

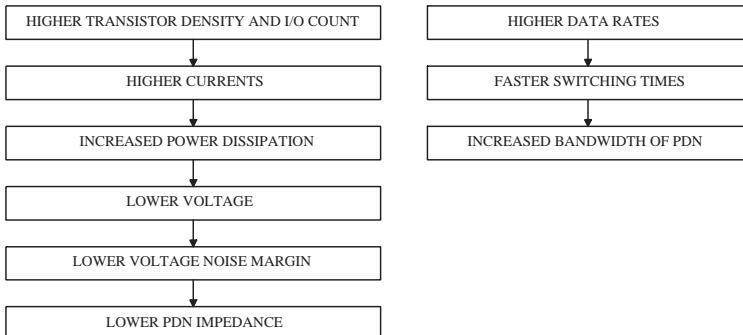


Figure 8.59 The PDN design challenge with the increased integration and data rates.

the gate oxide reliability [37]. A low noise PDN on a digital application PCB is also important to reduce EMI emission although this is usually not an issue on test fixtures for ATE.

One important point to note is that the power distribution requirements of a DUT while being tested or characterized can be significantly different from the power requirements when the DUT is being used on its target application [i.e., when one designs a PCB board that will contain this device (and normally several others) to perform a certain task]. The reason is that when a DUT is being tested or characterized on an ATE system, the switching behavior might be different from its mission mode. Typically the switching behavior is higher, meaning that there are higher current requirements from the DUT (e.g., during scan test the number of flip-flops switching simultaneously is very high). On the other hand, when testing the DUT PDN performance there is the question of whether the test fixture PDN should be the same as that on the PCB where the DUT will reside for its final application (e.g., a desktop computer motherboard). Some engineers advocate that the PDN performance on the test fixture should be the best possible independent of the PDN performance of the final application PCB. Note that in some cases, especially due to the need to have a socket in the test fixture, the performance of the test fixture PDN might actually be worse than that of the final PCB (system board) even when using the best possible PDN design as shown in Figure 8.60. The inductance of the test fixture socket, which will be discussed later in this section, is a significant bottleneck for the test fixture PDN.

For ATE manufacturers, the challenge is not only to develop device power supplies (DPS) with the required current/voltage capabilities but also to be able to integrate a sufficient number of these power supplies into an ATE system, since ICs can require multiple different power supplies.

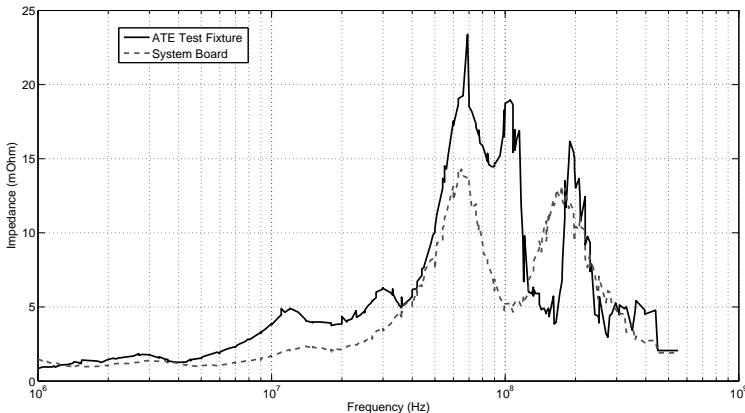


Figure 8.60 Example of the self-impedance difference between the application PCB board the DUT is intended to be used on and the ATE test fixture (reprinted with permission from [38]).

The test fixture PDN design challenge is presented in Figure 8.61. A PDN is not a simple power transmission line from the ATE DPS to the DUT, but it is a complex hierarchical system where power is stored and delivered by different components as shown in Figure 8.62. Each component of the PDN is responsible for providing power to the DUT in a different frequency range. The ATE DPS is responsible for supplying the bulk power but its response time is very slow, which means it cannot compensate for the power requirements of the DUT when it is switching its transistors at GHz frequencies. The decoupling capacitors and the power planes are responsible for storing and delivering power to the DUT for its needs in the hundreds of MHz range. Above these frequencies, the power delivery is the responsibility of the package and the on-die capacitance since the vias from the DUT socket to the PCB power planes and the DUT package will limit the maximum frequency of the PDN on the test fixture. All these components must work together to provide the proper power distribution to the DUT.

When designing a power distribution system for a test fixture, the objective is to keep the voltage change at the DUT power supply inputs as small as possible when the DUT forces a change in the supply current (dI/dt) due to transistor switching. Typically this requirement is defined in terms of the maximum ripple allowed on the power supply at the DUT power input package pads. Assuming a maximum possible transient current for the DUT, it is possible to compute the needed impedance value using Ohms law:

$$Z_{\text{TARGET}} = \frac{\text{Power Supply Voltage} \times \% \text{ Allowed Voltage Ripple}}{\text{Transient Current}} \quad (8.6)$$

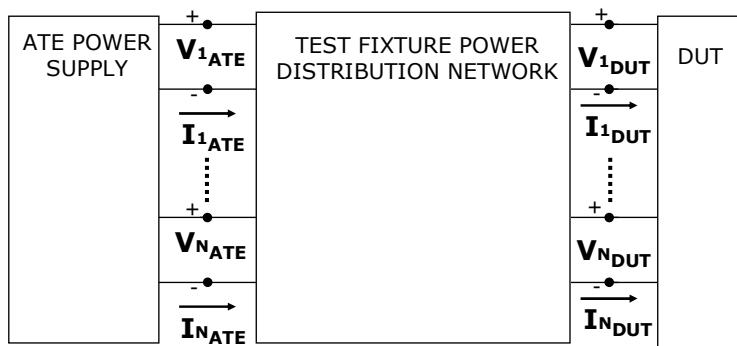


Figure 8.61 Diagram of the power distribution network (PDN) design challenge.

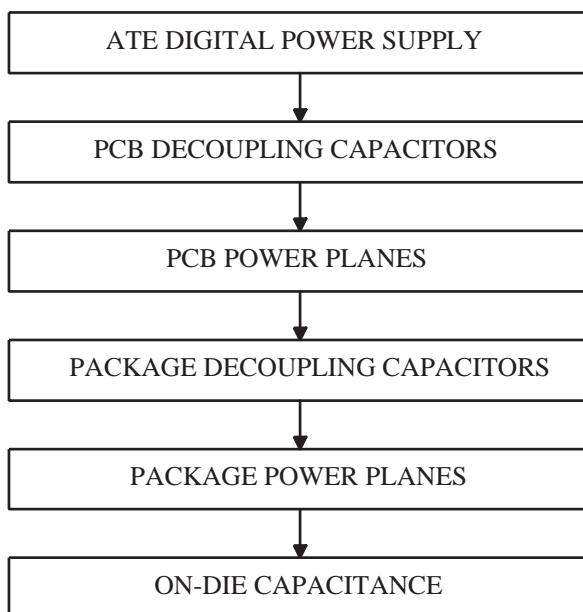


Figure 8.62 Hierarchical description of the components of a power distribution network.

This value will correspond to the maximum allowed impedance that guarantees that the voltage ripple will be below the required threshold. Below is an example of an impedance computation using this methodology:

$$\begin{aligned} \text{Power Supply Voltage} &= 1.8 \text{ V} \\ \text{Allowed Voltage Ripple} &= 5\% \\ \text{Transient Current} &= 1 \text{ A} \\ Z_{\text{Target}} &= \frac{1.8 \times 0.05}{1} = 0.09 \Omega \end{aligned} \quad (8.7)$$

Unfortunately, a frequency-independent value for the PDN is not possible. This is because each component of the PDN (power planes, decoupling capacitors, vias) has an impedance behavior that is not constant with frequency as will be shown later. This means that one needs to define also a frequency range where the impedance of the PDN must be below the target impedance as shown in Figure 8.63. This frequency range will depend on the rise/fall time of the transient current to the DUT. While the magnitude of the transient current determines the target impedance, the rise/fall time determines the upper corner frequency where the target impedance should be met.

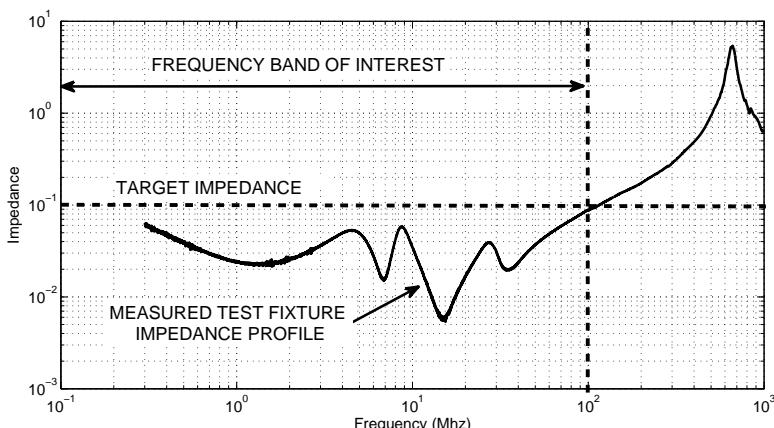


Figure 8.63 The challenge of maintaining a target impedance for a real PDN that has a frequency dependent impedance profile.

For an analytical analysis of a PDN design, the impedance matrix (Z) is used to describe the relationship between the voltage and current at the ATE power supply pins (V_{ATE}, I_{ATE}) and the voltage and current at the DUT package power supply (V_{DUT}, I_{DUT}) as shown in Figure 8.61 and described in (8.8).

$$\begin{bmatrix} V_{ATE} \\ V_{DUT} \end{bmatrix} = \begin{bmatrix} Z_{11}(f) & Z_{12}(f) \\ Z_{21}(f) & Z_{22}(f) \end{bmatrix} \begin{bmatrix} I_{ATE} \\ I_{DUT} \end{bmatrix} \quad (8.8)$$

Z_{11} is the self-impedance seen from the ATE power supply pins side and Z_{22} is the self-impedance seen at the DUT power supply pads side. Z_{12} and Z_{21} are the transfer impedance values from the ATE power supply pins to the DUT power supply pads and vice versa. Since we assume the PDN network to be reciprocal [39], then $Z_{12} = Z_{21}$. Figure 8.64 shows one example of the measured impedance matrix Z magnitude from a real test fixture (Section H.2 discusses in detail the measurement of the PDN impedance in a test fixture).

The most important result is the self-impedance at the DUT (Z_{22}). By analyzing the impedance profile it is then possible to validate if the PDN impedance is below the target impedance in the frequency range of interest. The self-impedance at the DPS (Z_{11}) and the transfer-impedance (Z_{12}, Z_{21}) are of less importance although Z_{11} is sometimes of interest to evaluate possible issues of the PDN design in regards to the ATE DPS (e.g., the DPS feedback loop stability). For time domain analysis (8.8) can be converted into the circuit shown in Figure 8.65 [40]. The next sections will discuss in more detail the different components of the test fixture PDN.

It is important to understand that the PDN performance will have an impact on the DUT performance. Figure 8.66 shows a comparison of three different test fixture PDN designs for the VDDQ power supply of a DDR3 memory application. The figure shows the measured data eye at one DQ pin for each PDN design. The DQ pin was running at 1.6 Gbps. Note that in one of the PDN examples there are no decoupling capacitors mounted on the test fixture, just the capacitance of the power planes is available to the DUT. The results show the impact that the PDN design can have on the data eye in the form of a reduced data eye width, especially in the case of single-ended I/O interfaces like DDR3. In this case the PDN design with the lowest impedance profile provides the best data eye performance in terms of the data eye width.

It is also interesting to note that even in the case where there were no decoupling capacitors assembled on the test fixture, the DUT was still able to function showing the importance of the power planes for the PDN performance as will be discussed in the next section. One of the reasons is that when a high-speed digital DUT is switching, the requirements on the power supply are across the frequency range and not at a specific frequency as shown in Figure 8.67, which implies the need to design the PDN taking into account its frequency response.

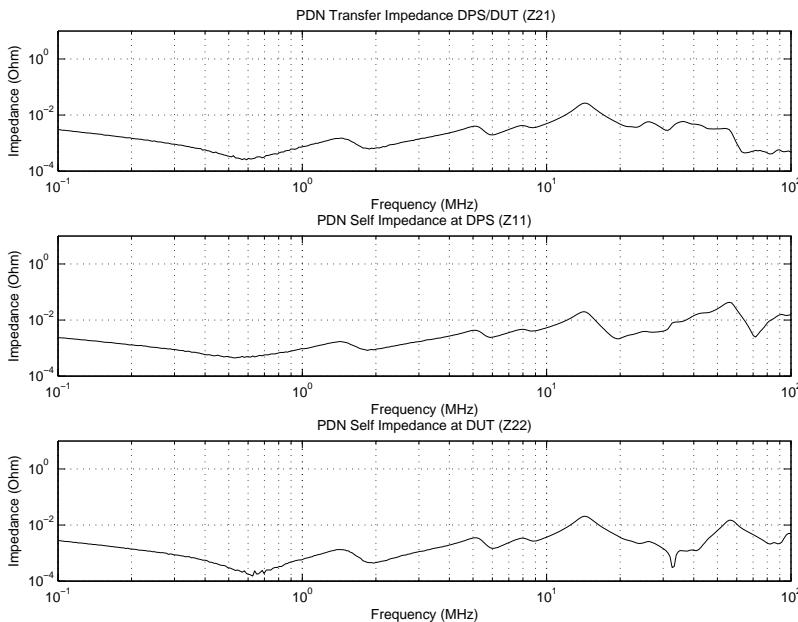


Figure 8.64 Impedance magnitude values for the power distribution network of a test fixture: Z_{21} transfer-impedance from the DPS to the DUT (top), Z_{11} self-impedance at the DPS side (center), and the Z_{22} self-impedance at the DUT side (bottom). The Z_{12} transfer-impedance is equivalent to Z_{21} .

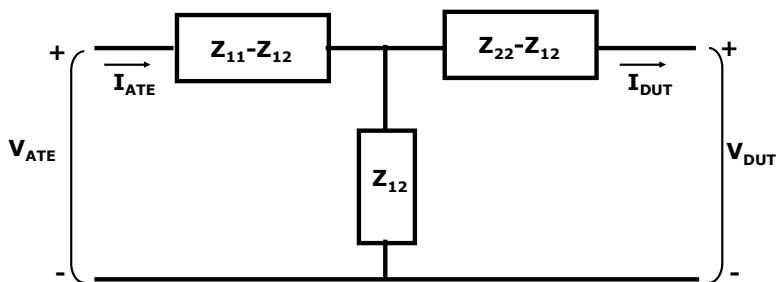


Figure 8.65 Equivalent circuit for time domain analysis of a PDN based on the impedance matrix Z .

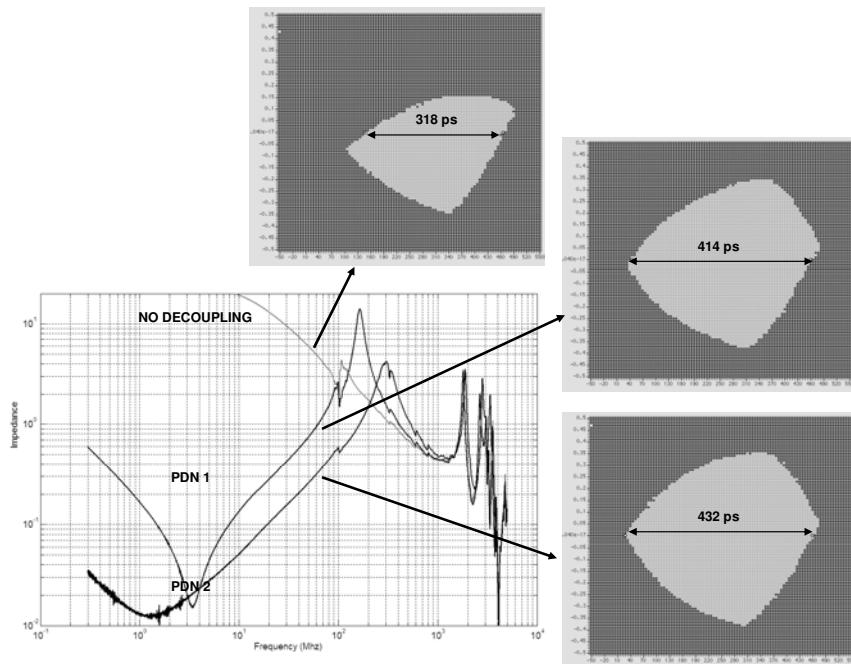


Figure 8.66 Comparison of the effect of different test fixture PDN designs on the data eye from a DQ pin of a DDR3 memory running at 1.6 Gbps.

8.11.1 Power Planes

The task of designing a power distribution network starts with appropriate choices for the power and ground planes for the ATE test fixture PCB stack-up. The power and ground planes serve as a storage capacitor that is able to store and provide small amounts of charge at very high frequencies when compared to the ATE DPS or the decoupling capacitors. Its ability to store charge is characterized by the power plane capacitance which can be approximately computed by the following equation [4]:

$$C = \frac{0.0885 \varepsilon_r A}{d} \text{ pF} \quad (8.9)$$

where ε_r is the relative dielectric permittivity of the dielectric material used for the power plane, A is the area of the power plane in cm^2 , and d is the height of the dielectric in cm between the power and ground planes as shown in Figure 8.68.

Since the objective is to minimize the loop inductance between the power plane and ground plane pair, the power plane should be made with

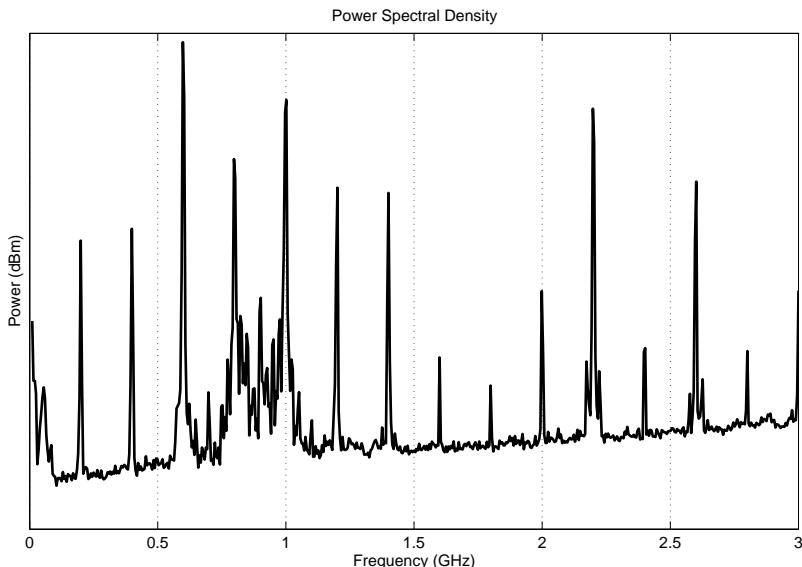


Figure 8.67 Power spectral density of the noise present on the power plane of an ATE test fixture for a high-speed memory running at 1.6 Gbps. The noise spectrum was measured by connecting a spectrum analyzer to a test point on the ATE test fixture power plane with a $50\ \Omega$ probe (high impedance compared to the PDN) while the DUT was running a pattern that generated a large number of logic transitions.

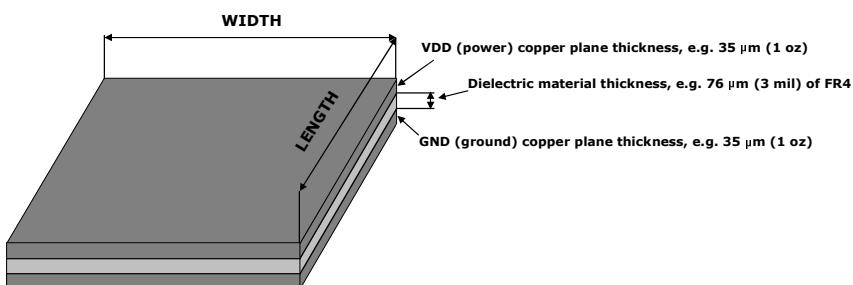


Figure 8.68 Geometry of a power plane.

the smallest dielectric height possible, typically 50.8 to 101.6 μm (2 to 4 mil) using a dielectric material with a large ϵ_r value. This is not the same as the low loss requirement for high-speed digital signal routing where a low ϵ_R and loss tangent are preferred. There are dielectric materials that have been developed specifically to be used in power planes, with a very small thickness and a large ϵ_r value. This type of dielectric material can have significant advantages as shown in [41].

Figure 8.69 shows an example of a stack-up for a power intensive application (a microprocessor). Note that each power supply has a separate power plane. This stack-up also shows one of the main challenges associated with the layer choice for the power planes in a multilayer test fixture PCB. On one side the power planes should be as close as possible to the top of the test fixture PCB to reduce the inductance of the vias connecting the socket power pins to the power planes. To maintain stack-up symmetry for minimal PCB warpage, manufacturers prefer to place an equal number of power planes on the other side of the PCB, or place them all together in the middle. In the example of Figure 8.69 the power planes were placed at the middle of the stack-up due to manufacturing reasons, which implies additional inductance for the vias connecting the power planes to the DUT socket.

It is important to understand that the impedance response of a power plane in regards to frequency will depend significantly on the geometry of the power plane and the specific locations where the DUT power pins reside. Figure 8.70 shows an impedance plot versus frequency of two different power planes with different areas but the same dielectric height. The impedance is measured at the geometric center of each power plane. The figure shows that the large power plane although having a higher capacitance, has its resonant frequency around 50 MHz. On the smaller power plane the resonant frequency is around 900 MHz.

ATE test fixtures are usually large in size which means that their resonant frequencies will be very low if the power plane covers the entire PCB area. Figure 8.71 shows a picture of an investigation board created to measure the resonant frequency in an ATE test fixture. The board size is 58.2 by 42.9 cm (22.9 by 16.9 in) corresponding to a full-size test fixture for a particular ATE platform. Two power planes were implemented using an FR4 dielectric with a thickness of 63.5 μm (2.5 mil) and 190.5 μm (7.5 mil) and another power plane was implemented with an 11 μm (0.43 mil) 3M embedded capacitance dielectric material. The measured results shown in Figure 8.72 show that the resonant frequency is below 100 MHz for the full-size and half-size test fixture. Note also the results for the thicker power plane where it is possible to observe the reduced capacitance when compared to the thinner power plane.

The resonant frequencies can have an impact on the PDN performance

Layer Number		Layer Name	Layer Type
LAYER 1		Top	Signal
LAYER 2		DGND	Ground Plane
LAYER 3		INR1	Signal
LAYER 4		DGND	Ground Plane
LAYER 5		INR2	Signal
LAYER 6		DGND	Ground Plane
LAYER 7		INR3	Signal
LAYER 8		DGND	Ground Plane
LAYER 9		INR4	Signal
LAYER 10		DGND	Ground Plane
LAYER 11		PGND1	Ground Plane
LAYER 12		PWR1 - MS-DPS Supplies	Power Plane
LAYER 13		PGND2	Ground Plane
LAYER 14		PWR2 - VDDIO	Power Plane
LAYER 15		PGND3	Ground Plane
LAYER 16		PWR3 - VDD	Power Plane
LAYER 17		PGND3	Ground Plane
LAYER 18		PWR3 - VDD	Power Plane
LAYER 19		PWR3 - VDD	Power Plane
LAYER 20		PGND3	Ground Plane
LAYER 21		PWR3 - VDD	Power Plane
LAYER 22		PGND3	Ground Plane
LAYER 23		PWR4	Power Plane
LAYER 24		PGND4	Ground Plane
LAYER 25		PWR5	Power Plane
LAYER 26		PGND5	Ground Plane
LAYER 27		DGND	Ground Plane
LAYER 28		INR5	Signal
LAYER 29		DGND	Ground Plane
LAYER 30		INR6	Signal
LAYER 31		DGND	Ground Plane
LAYER 32		INR7	Signal
LAYER 33		DGND	Ground Plane
LAYER 34		INR8	Signal
LAYER 35		DGND	Ground Plane
LAYER 36		Bottom	Signal

Nelco 4000-13SI
Nelco 4000-13

Figure 8.69 Example of the stack-up for a power intensive DUT (courtesy of Altanova).

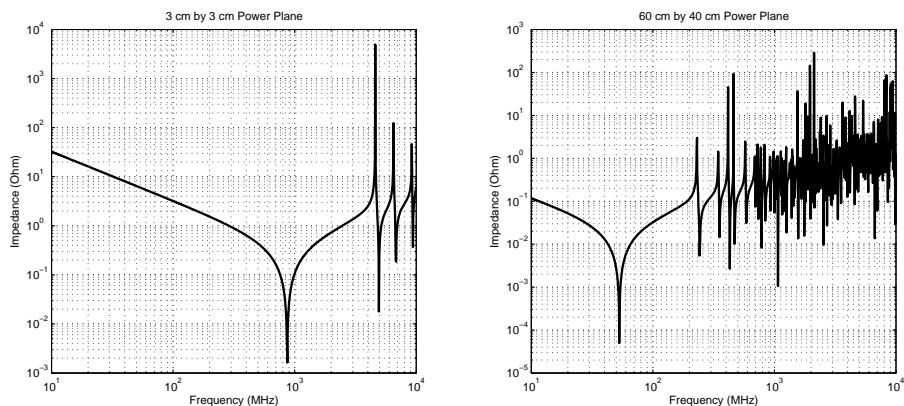


Figure 8.70 Example of the self-impedance profile for two different power planes at the geometric center. A 3 by 3 cm (1.2 by 1.2 in) power plane (left) and a 40 by 60 cm (15.8 by 23.6 in) power plane (right). Both power planes have a 76.2- μ m (3 mil) dielectric thickness using a FR4 type dielectric [42].

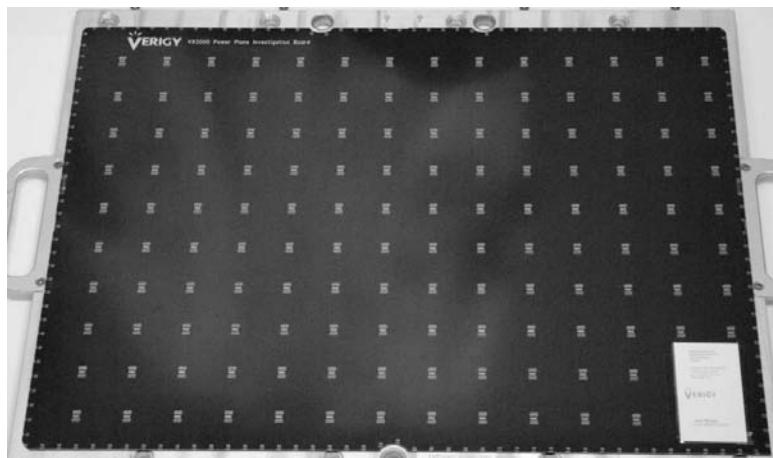


Figure 8.71 Picture of a printed circuit board designed to investigate the power plane characteristics of a typical ATE test fixture (courtesy of RD Circuits and Verigy).

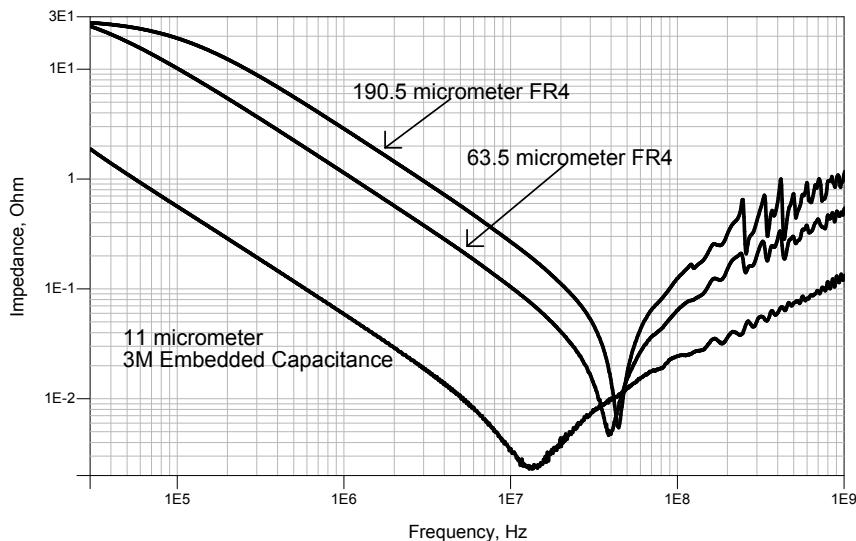


Figure 8.72 Impedance results at the center of the power plane for a full-size (58 by 43 cm) ATE test fixture with a 63.5- μ m (2.5 mil) and 190.5- μ m (7.5 mil) FR4 dielectric thickness and an 11- μ m (0.43 mil) 3M embedded capacitance dielectric material.

since they can interact with the decoupling capacitor resonant frequencies. One approach developed to smooth the resonant frequency peaks at higher frequencies is to use a lossy decoupling or edge termination around the power planes [43, 44]. Notice that this is opposite to the need for lower loss dielectric materials on high-speed signal layers and often results in a mixture of laminate materials for a high performance PCB test fixture. Note also that the impedance profile of a power plane does not only vary with frequency but also with the geometric location where the impedance is being measured or calculated especially at high frequencies as shown in Figure 8.73 [33].

8.11.2 Decoupling Capacitors

Decoupling capacitors, also known as bypass capacitors, are used to provide a low impedance source of energy to middle and high frequency changes in the DUT load current. One can imagine that they are like an energy reservoir for fast changes on the load current that cannot be compensated fast enough by the ATE power supply. The challenge is that capacitors are not ideal elements, which forces the power distribution network design to use a mixture of different decoupling capacitor types and multiple numbers of the same capacitor to reduce the parasitic inductance. Figure 8.74 shows a picture of

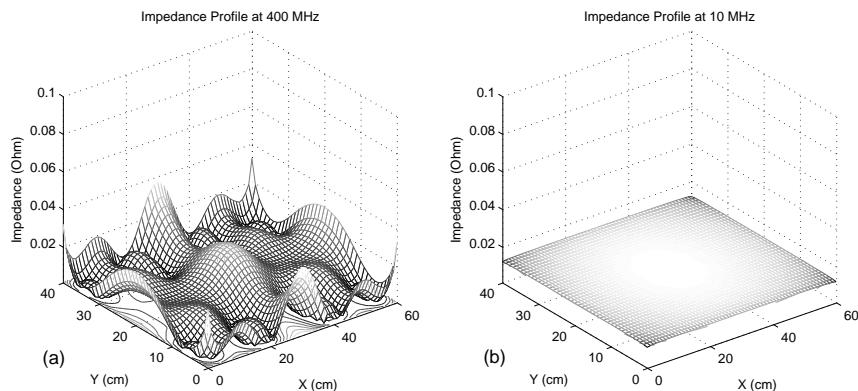


Figure 8.73 Example of (a) the impedance distribution at 400 MHz and (b) 10 MHz for a 40 by 60 cm (15.8 by 23.6 in) power plane with a 76.2- μ m (3 mil) dielectric thickness using an FR4 type dielectric [42].

the decoupling capacitors on the bottom of an ATE test fixture under the DUT socket.

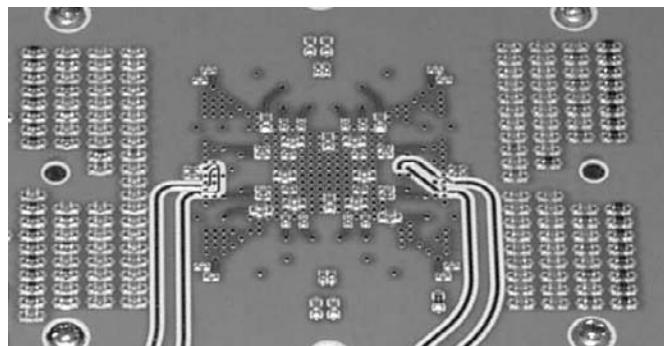


Figure 8.74 Example of the decoupling capacitors on the bottom of the test fixture below the DUT socket and also surrounding it.

A real capacitor does not behave as an ideal capacitor in the entire frequency range and is typically modeled by a capacitance value together with an equivalent series resistance (ESR) and an equivalent series inductance (ESL) as shown in Figure 8.75. Since a real capacitor is an RLC circuit, the ESR and ESL values will determine the resonant frequency for the capacitor. Note that this is still only an approximation since the ESR and ESL values of a capacitor will vary with frequency [33].

To better understand the effect that the ESR and ESL have on a capacitor impedance, Figure 8.76 shows a plot of the impedance versus frequency for an

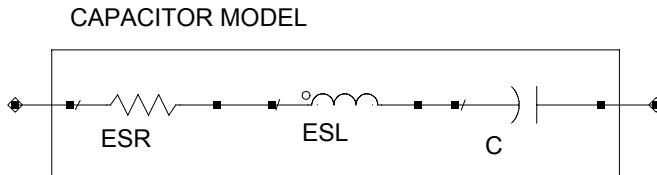


Figure 8.75 Typical model for a surface mounted decoupling capacitor.

ideal 1- μF capacitor and a 0603⁷ type 1- μF capacitor with an ESR of 15 m Ω and an ESL of 2 nH.

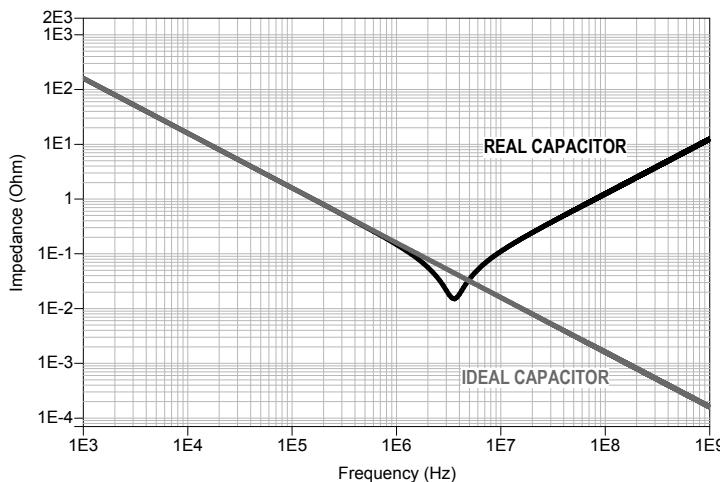


Figure 8.76 Comparison of an ideal 1- μF capacitor with a 0603 1- μF capacitor with an ESR of 15 m Ω and an ESL of 2 nH.

The figure shows that there is a resonance due the capacitor reactance being equal to the inductor reactance [2]. After this resonance the capacitor no longer behaves like a capacitor but instead like an inductor. It is important to note that the ESL value will depend on the PCB footprint and also on the length of the via connecting the decoupling capacitor to the respective power plane. Because of this, an appropriate footprint should be used for the decoupling capacitors to keep the inductance to a minimum as shown in Figure 8.77.

To understand the effect of increasing or decreasing the ESR and ESL value of a decoupling capacitor on its frequency domain response, Figure 8.78 shows a comparison of an ideal capacitor with that of a more realistic

⁷0603 refers to the physical size of the capacitor which is standardized.

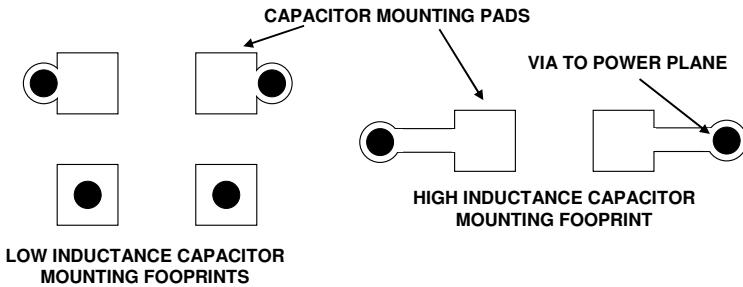


Figure 8.77 Capacitor mounting footprint influence on the capacitor ESL value.

ESR/ESL capacitor model. The left figure shows the effect of increasing the ESR value of the capacitor while keeping the ESL value constant. In this case the resonance of the capacitor is smoothed and the capacitor is no longer able to achieve the same low impedance value at the resonance frequency as before. This effect will be important when we discuss the combination of multiple different decoupling capacitors later. Figure 8.78 (right) shows the effect of reducing the ESL value of the capacitor while keeping the ESR value constant. In this case, the capacitor frequency response shows that the capacitor resonance happens at a higher frequency meaning that it is able to behave as a capacitor to a higher frequency. These results show that ideally one should try to use capacitors with the lowest possible ESL value.

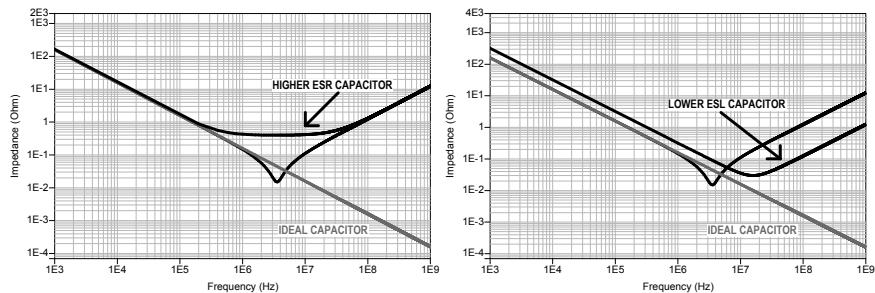


Figure 8.78 Comparison of the effects on the capacitor frequency response of an increase on the ESR value or a decrease on the ESL value.

Figure 8.79 shows how the vias connecting the decoupling capacitor to the power plane add to the decoupling capacitor parasitic inductance value and in this way increase its ESL value. To minimize this value the power planes should be as close as possible to the decoupling capacitor although this is not always possible. The common approach to the capacitor parasitic inductance challenge is to reduce the inductance by using multiple capacitors

to achieve a given capacitance and parasitic inductance value. For example, if the PDN design requires a 4 μF capacitor but the associated ESL and parasitic inductance are too high, it is possible to use four 1 μF capacitors instead. If the ESL value of each is the same, the parasitic inductance will be reduced by a factor of four when compared to using a single 4 μF capacitor. This effect is shown in Figure 8.80. The drawback of this approach is, of course, the layout space these capacitors will take while their additional cost is only a minor factor for an ATE test fixture.

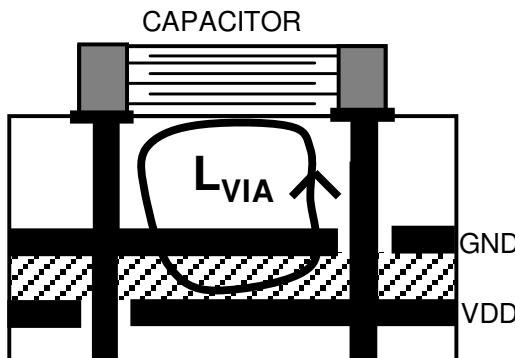


Figure 8.79 Graphic demonstration of the parasitic inductance created by the connection of a surface mounted decoupling capacitor to the power plane through vias.

Apart from the physical sizes, a capacitor type also depends on its underlying manufacturing technology. Table 8.8 describes some of the capacitor technologies available for use in a PDN for high-speed digital applications.

One important topic when discussing decoupling capacitors is what happens when assembling multiple different decoupling capacitors in parallel. Since each decoupling capacitor has an ESR and an ESL value, this parallel circuit can have a complex behavior. To better understand this, Figure 8.81 shows two different circuits of two parallel decoupling capacitors where all the capacitors have the same ESR and ESL value but their capacitance value separation is different (one circuit contains a 1 $\mu\text{F}/100 \text{nF}$ capacitor combination and the other a 1 $\mu\text{F}/10 \text{nF}$ combination). The frequency domain self-impedance results for both circuits show that after the self-resonance of the first capacitor occurs, another resonance between the two capacitors' self-resonances exists. The problem with this resonance is that its peak corresponds to a high impedance value which is bad for the power distribution performance. The results also show that in the case where the capacitors are

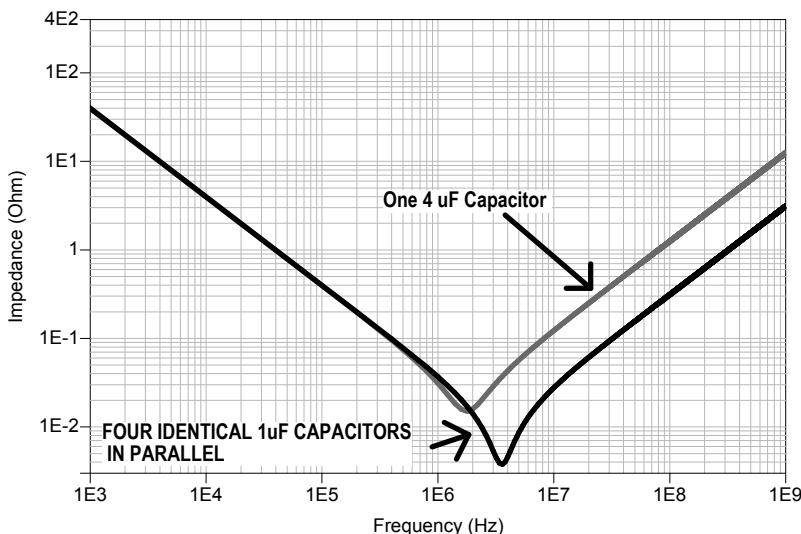


Figure 8.80 Comparison of the effect of adding four identical capacitors in parallel.

Table 8.8
Comparison of Capacitors Types [2]

Capacitor Type	Comments
Low ESR tantalum	1 to 1,000 μ F, medium physical size, low voltage, low ESR, failure mode is a short and a fire. Usually considered as a low frequency bulk capacitor
Tantalum organic	1 to 1,000 μ F, low voltage, low ESR, self-healing failure mode. Usually considered as a low frequency bulk capacitor
Ceramic	Small capacitance, small physical size, high and low voltage, very low ESR, highest reliability
Arrays	Ceramic capacitors with an array of contacts for very low ESL
Low inductance capacitor (LICC) chip	Sometimes referred to also as a reverse geometry capacitor (RGC), it has a very low ESL but with a nonstandard footprint
Controlled ESR	ESR value for better resonance control (1 μ F and 10 μ F)

further apart in capacitance value (10 nF compared with 100 nF) the peak of the resonance between the two capacitors is higher. The conclusion to be taken from this small experiment is that when using multiple capacitor values in a PDN, it is important to use multiple capacitance values that are not very far apart in their capacitance value. This effect can also be reduced by using capacitors with higher ESR values as will be discussed next.

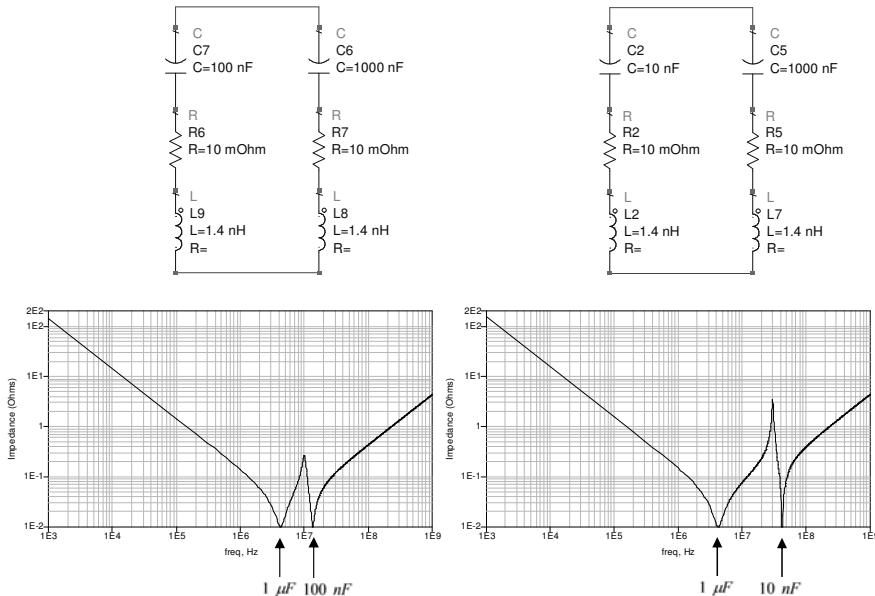


Figure 8.81 Example of the resonances of a two-capacitor decoupling configuration.

One new category of decoupling capacitor that has become available is the controlled ESR capacitor [45]. The ESR value of a capacitor can have a significant impact on the shape of the resonance associated with the decoupling capacitor by smoothing its resonance as already shown in Figure 8.78 (left). The important point is that in a configuration with multiple different capacitors, a higher ESR value generates smaller resonances between the different capacitors as shown in Figure 8.82. The figure shows the impedance profile of two circuits each composed of a 1- μ F and a 10-nF capacitor. The ESL of each capacitor is 1.4 nH and the ESR of each capacitor is 10 mΩ, with the exception that in the second circuit the ESR value for the 1- μ F capacitor is increased to 100 mΩ. This increase on the ESR is enough to bring the peak of the resonance below 1 Ω.

In the cases where layout space is at a premium (e.g., multisite applications) and given the fact that a significant number of capacitors might be needed to achieve the desired power distribution network, one option to

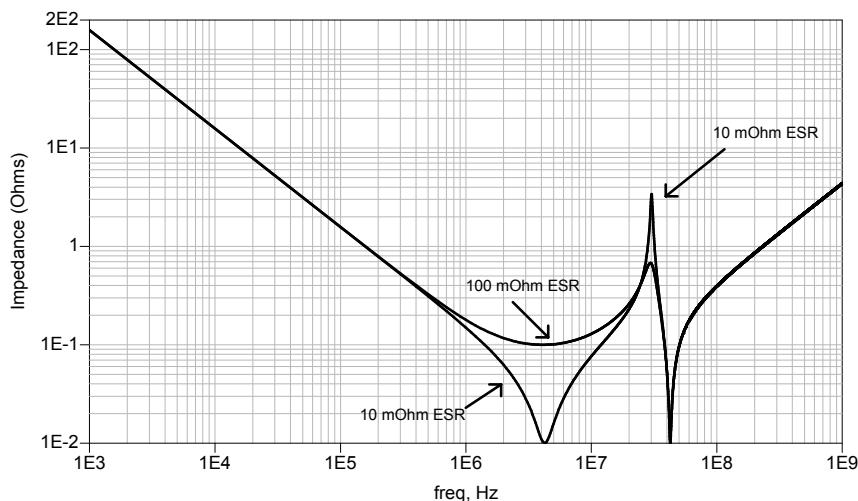


Figure 8.82 Comparison of the effects on the capacitor frequency response with an increase in the ESR value.

address this challenge is to stack the capacitors as shown in Figure 8.83. The drawback is that the higher the capacitor is on its stack, the larger will be its loop inductance. Because of this, if a stacked configuration is used it is important to have the lowest value capacitors on the bottom of the stack.

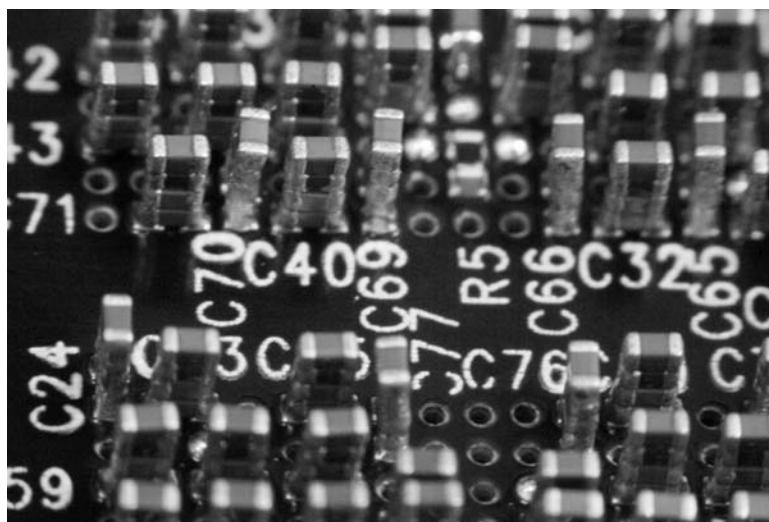


Figure 8.83 Stacked capacitors in a test fixture (courtesy of RD circuits).

Another technique that has started to gain acceptance is to embed the capacitors in the test fixture PCB or even in the socket. This approach allows the decoupling capacitors to be mounted much closer to the DUT BGA power pin so that the parasitic inductance of any mounting vias or connection traces can be reduced [38, 46].

8.11.3 Socket Inductance

The need to use a socket for the DUT on an ATE application raises the issue of the inductance of the socket power pins since the socket pins will have a certain length. This issue becomes more problematic if in its final application the DUT is soldered to the PCB without any socket as shown in Figure 8.84.

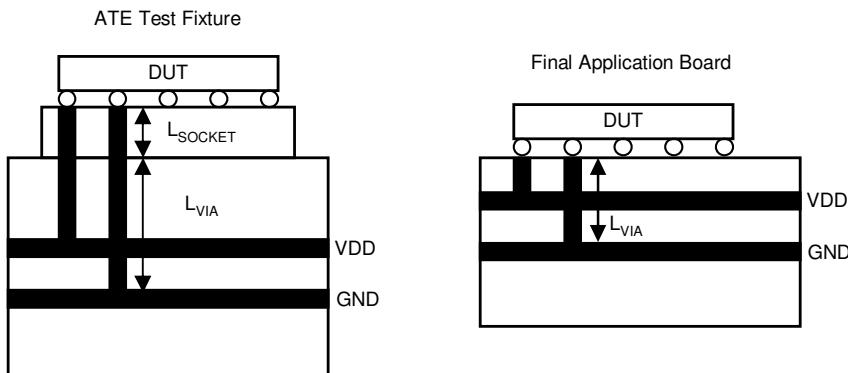


Figure 8.84 Comparison of the inductance of the DUT connection to the power planes for an ATE test fixture and a final application PCB where the DUT is soldered without any socket.

As already discussed in Section 8.11.1, this problem can be exacerbated if the power planes cannot be set close to the top of the test fixture PCB due to stack-up restrictions. This would make the length of the vias from the socket to the power plane longer than in the final application, thus increasing the inductance.

There are not too many alternatives available to address this issue. One option is to reduce the socket inductance by making the socket pins shorter or to use some type of membrane socket. Reducing the socket pin length has mechanical limits and using a membrane type contactor instead of a socket also raises other problems (reduced mechanical compliance being one of them). Another option is to add power decoupling into the socket itself by creating a power plane and adding decoupling capacitors [38]. This approach is especially suited to applications where the DUT BGA contains

a central area that is dominated by power pins (like on a high performance microprocessor). The drawbacks are the additional complexity and cost and also the fact that it still cannot guarantee that the performance would be equal to the case of the DUT BGA being directly soldered into the PCB. Another approach is to evaluate if the added inductance from the socket (and the via to the power plane) is compensated enough by the package substrate and on-die power decoupling. The problem with this approach is that it requires a good knowledge and model of the DUT PDN.

8.11.4 Power Distribution Network Design

Designing a power distribution network for a test fixture consists first of identifying the power requirements of the DUT and its impact on the PDN impedance requirements as described in (8.6). The next step is to choose the stack-up for the power planes taking into account the number of power supplies in the DUT and the discussion on power planes in Section 8.11.1 followed by the choice of the needed decoupling capacitors. Note that choosing the decoupling capacitors entails not only choosing their values but also their type and location as discussed in Section 8.11.2.

Power distribution design is sometimes seen as a “black-magic” art, with several design philosophies available in the technical literature that are sometimes contradictory [47]. Some of the available design methodologies are:

- Frequency domain target impedance method (FDTIM);
- Multipole (MP);
- Capacitors-by-the-decade (CBD);
- Big “V”;
- Distributed matching bypassing.

Reference [35] provides more detailed discussion into some of the above methods. In most PDN designs the guiding objective is not only achieving the target impedance across the PDN frequency band of interest but also to make the impedance response as flat as possible across the frequency band of interest.

8.11.5 Power Distribution Network Simulation

It is important to be able to simulate a DUT loadboard PDN before physically building the test fixture to evaluate the choice of the decoupling capacitors and verify if the PDN fulfills the impedance requirements. Usually one starts with an initial idea consisting of the power planes and the number and types of

decoupling capacitors. This could be based in the previous test fixture design for a similar application. The next step is to model each item using the ESR and ESL values for the capacitor and power planes. This is the tough part since sometimes these values are not readily available and one might need to make an educated guess, although some manufacturers provide design kits with capacitor models. It is also critical that the inductance value of the vias from the decoupling capacitors to the power planes is added to the ESL value of the decoupling capacitors. This value will be dependent on the stack-up and must be computed based on the via length (see [33] for some guidelines on computing the via inductance).

This type of simulation can be easily performed on a SPICE-like simulator as shown in Figure 8.85. The impedance profile is obtained by using a 1 A amplitude sinusoidal current source and measuring the voltage. The values will correspond to the impedance value since $Z = V/I = V/(1 \text{ A}) = V (\Omega)$.

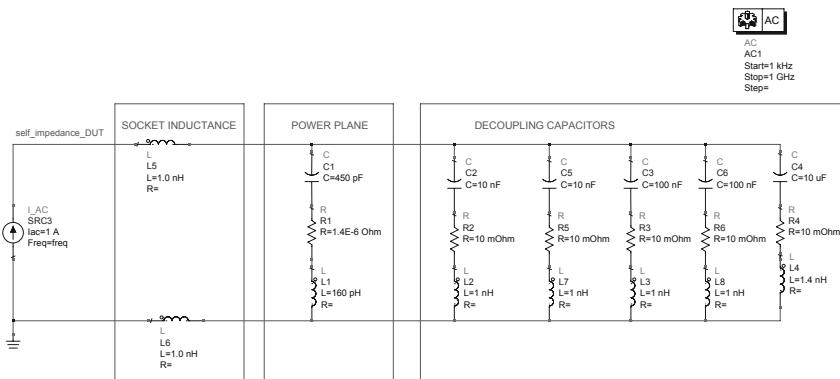


Figure 8.85 Example of a simplified test fixture power distribution network simulation example using an RLC model for the decoupling capacitors, socket via, and power planes.

Figure 8.86 shows the simulation results from the setup in Figure 8.85. In the simulated impedance profile it is easy to identify the resonance peaks due to the different decoupling capacitors. This allows the designer to identify where to try to improve the impedance profile by adding more decoupling capacitors with different capacitance values or changing to a different type of capacitor like a controlled ESR one. Because this type of PDN model treats the PDN as a lumped circuit (i.e., neglects the distributed nature of the design), one should not expect a high degree of accuracy. For example, such a model will overestimate the severity of resonances (and thus cause a conservative design in this respect), but also overestimate the high frequency performance

(where propagation times become important for the PDN's reaction to fast changes).

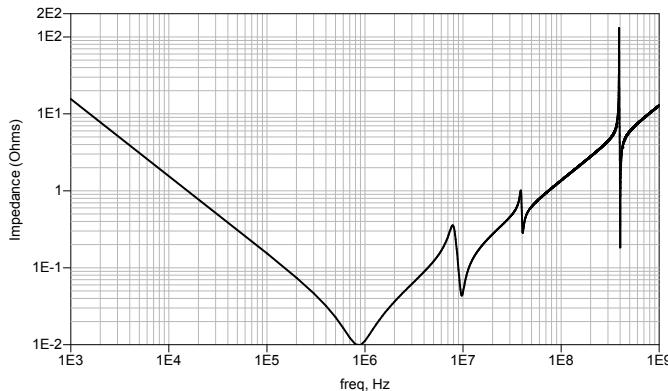


Figure 8.86 Simulated impedance results from the simulation setup of Figure 8.85.

Apart from SPICE several more complex tools exist that are specifically designed for power distribution network simulation. Some of these tools are discussed in Appendix G.

References

- [1] A. Mayer and R. Plitschka, *High Speed Testing: Introduction and Reference Guide to Testing High Speed Digital ICs*. Hewlett-Packard, 1989.
- [2] L. W. Ritchey, *Right the First Time: A Practical Handbook on High Speed PCB and System Design, Volume One*. Speeding Edge, 2003.
- [3] L. W. Ritchey, *Right the First Time: A Practical Handbook on High Speed PCB and System Design, Volume Two*. Speeding Edge, 2006.
- [4] H. Johnson and M. Graham, *High-Speed Digital Design*. Addison-Wesley, 1993.
- [5] H. Johnson and M. Graham, *High-Speed Signal Propagation*. Addison-Wesley, 2003.
- [6] S. H. Hall, G. W. Hall, and J. A. McCall, *High-Speed Digital System Design: A Handbook of Interconnect Theory and Design Practices*. Wiley-IEEE Press, 2000.
- [7] E. Bogatin, *Signal and Power Integrity Simplified*. Prentice-Hall, 2010.
- [8] D. Beaulieu, *Printed Circuit Board Basics*. UP Media Group, 2003.
- [9] C. Coombs, *Printed Circuits Handbook*. McGraw-Hill, 2007.

- [10] R. Rajsuman, N. Masuda, and K. Yamashita, "Architecture and Design of an Open ATE to Incubate the Development of Third-Party Instruments," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, Oct. 2005.
- [11] D. Brooks, *Signal Integrity Issues and Printed Circuit Board Design*. Prentice-Hall, 2003.
- [12] G. Brist, S. Hall, S. Clouser, and T. Liang, "Non-Classical Conductor Losses Due to Copper Foil Roughness and Treatment," *ECWC 10 Conference at IPC Printed Circuits Expo*, 2005.
- [13] H. Barnes, J. Moreira, T. McCarthy, W. Burns, C. Gutierrez, and M. Resso, "ATE Interconnect Performance to 43 Gbps Using Advanced PCB Materials," *IEC DesignCon*, 2008.
- [14] M. W. Jawitz and M. J. Jawitz, *Materials for Rigid and Flexible Printed Wiring Boards*. John Wiley and Sons, 2007.
- [15] B. B. Szendrenyi, H. Barnes, J. Moreira, M. Wollitzer, T. Schmid, and M. Tsai, "Addressing the Broadband Crosstalk Challenges of Pogo Pin Type Interfaces for High-Density High-Speed Digital Applications," *International Microwave Symposium*, 2007.
- [16] J. Moreira, M. Tsai, J. Kenton, H. Barnes, and D. Faller, "PCB Loadboard Design Challenges for Multi-Gigabit Devices in Automated Test Applications," *DesignCon*, 2006.
- [17] G. Brist, D. Cullen, D. Luttrull, and J. Martin, "Reduction of High-Frequency Signal Loss through the Control of Conductor Geometry and Surface Metallization," *International Conference of the Surface Mount Technology Association (SMTA)*, 2002.
- [18] M. Kikuchi and R. Yongho, "Via Structure Analysis for 10Gbps Signal Transmission on PCB," *Semicon Japan*, 2008.
- [19] A. C. Segura, "Signal Integrity Analysis of a 26 Layers Board with Emphasis on the Effect of Non-Functional Pads," *IEEE International Symposium on Electromagnetic Compatibility*, 2008.
- [20] E. Bogatin, L. Simonovich, S. Gupta, and M. Resso, "Practical Analysis of Backplane Vias," *IEC DesignCon*, 2009.
- [21] H. Barnes, J. Moreira, H. Ossoining, M. Wollitzer, T. Schmid, and M. Tsai, "Development of a Pogo Pin Assembly and Via Design for Multi-Gigabit Interfaces on Automated Test Equipment," *Asia-Pacific Microwave Conference*, 2006.
- [22] W. Liu and M. Pecht, *IC Component Sockets*. John Wiley and Sons, 2004.
- [23] H.-C. Teng, M.-K. Chen, C.-H. Yeh, Y.-J. Huang, and S.-L. Fu, "Study of Contact Degradation in Final Testing of BGA Sockets," *IEEE International Conference on Electronic Materials and Packaging*, 2007.

- [24] J. Moreira, H. Barnes, and G. Hoersch, "Analyzing and Addressing the Impact of Test Fixture Relays for Multi-Gigabit ATE I/O Characterization Applications," *IEEE International Test Conference*, 2007.
- [25] D. C. Keezer and Q. Zhou, "Alternative Interface Methods for Testing High-Speed Bidirectional Signals," *IEEE International Test Conference*, 1998.
- [26] IEEE, "Southwest Test Workshop." <http://www.swtest.org>.
- [27] W. R. Mann, F. L. Taber, P. W. Seitzer, and J. J. Broz, "The Leading Edge of Production Wafer Probe Test Technology," *IEEE International Test Conference*, 2004.
- [28] T. Ueno and Y. Kondoh, "Membrane Probe Technology for MCM Known-Good-Die," *IEEE International Test Workshop*, 1994.
- [29] J. Moreira, G. Haensel, and F. Koban, "Addressing the Challenges of Implementing an At-Speed Production Test-Cell for 10Gb/s Wafer Probing," *DesignCon*, 2005.
- [30] J. Broz and B. Mann, "Wafer Probe Test Technology Tutorial," *IEEE International Test Conference*, 2008.
- [31] G. H. Johnson, "Challenges of High Supply Currents During VLSI Test," *IEEE International Test Conference*, 2000.
- [32] J.-P. Mallet, "High Current DPS Architecture for Sort Test Challenge," *IEEE International Test Conference*, 2002.
- [33] I. Novak and J. R. Miller, *Frequency-Domain Characterization of Power Distribution Networks*. Artech House, 2007.
- [34] M. Swaminathan and A. E. Engin, *Power Integrity Modeling and Design for Semiconductors and Systems*. Prentice-Hall, 2007.
- [35] I. Novak, *Power Distribution Network Design Methodologies*. IEC, 2008.
- [36] H. Lan, R. Schmitt, and C. Yuan, "Prediction and Measurement of Supply Noise Induced Jitter in High-Speed I/O Interfaces," *IEC DesignCon*, 2009.
- [37] M. Popovich, A. V. Mezhiba, and E. G. Friedman, *Power Distribution Networks with On-Chip Decoupling Capacitors*. Springer, 2008.
- [38] O. Kikinski, S. Lupo, G. Sizikov, and C. Y. Chung, "Embedded Power Delivery Decoupling in Small Form Factor Test Sockets," *IEEE International Test Conference*, 2008.
- [39] W. C. Johnson, *Transmission Lines and Networks*. McGraw-Hill, 1950.
- [40] D. M. Pozar, *Microwave Engineering*. John Wiley and Sons, 2005.
- [41] J. Peiffer, B. Greenlee, and I. Novak, "Electrical Performance Advantages of Ultra-Thin Dielectric Materials Used for Power-Ground Cores in High Speed, Multilayer Printed Circuit Boards," *IPC Expo*, 2003.

- [42] I. Novak, "Electrical Integrity: Signal and Power Integrity," 2009. <http://www.electrical-integrity.com>.
- [43] X. Wu, "Matched Lossy Decoupling Termination for Power Plane Noise Mitigation," *International Symposium on Electromagnetic Compatibility*, 2005.
- [44] I. Novak, "Reducing Simultaneous Switching Noise and EMI on Ground/Power Planes by Dissipative Edge Termination," *IEEE Transactions on Advanced Packaging*, vol. 22, Aug. 1999.
- [45] H. Ishida, J. Prymak, C. Burket, and I. Novak, "Controlled ESR Capacitors Have Arrived," *IEC DesignCon TecForum*, 2007.
- [46] A. Detofsky, O. Vikinski, S. Lupo, and T. Swettlen, "Power Integrity Ingenuity at Test," *Burn-In and Test Socket Workshop*, 2009.
- [47] I. Novak, "Comparison of Power Distribution Network Design Methods: Bypass Capacitor Selection Based on Time Domain and Frequency Domain Performances," *IEC DesignCon TecForum*, 2006.

9

Advanced ATE Topics

This chapter presents some advanced topics that are of interest for the test of high-speed digital I/O interfaces with automated test equipment.

9.1 ATE Specifications and Calibration

Understanding ATE specifications and calibration can be challenging for the test engineer. This is due to the fact that ATE manufacturers, although using the same nomenclature for a given specification parameter, might use different definitions or measurement methodologies to guarantee that parameter. This makes it more difficult to compare specifications across different ATE platforms. Also, unlike with bench instrumentation, the large number of channels associated with an ATE system presents challenges for defining and measuring specifications for the ATE system and its calibration. The next subsections discuss some of these topics in detail.

9.1.1 Accuracy and Resolution

It is important to understand the difference between resolution and accuracy. Resolution is the smallest change that can be forced or detected for an electrical or timing parameter (e.g., 1 mV for the receiver threshold voltage or 1 ps for an edge placement of a low to high transition on a tester pin driver). Note that although there is an inherent resolution in the hardware (e.g., quantization of the digital to analog converter that sets the threshold voltage value), the instrument software can implement any type of resolution. When discussing the resolution of a tester module we should be aware to address the hardware resolution and not a theoretical software resolution without correspondence to the test hardware.

The accuracy of a tester module is the degree of deviation of a measured parameter from its true or reference value (e.g., $+/- 10$ mV accuracy for the receiver threshold). The precision of an ATE instrument, also expressed as reproducibility or repeatability, is the degree to which repeated measurements under identical conditions show the same results as illustrated in Figure 9.1 [1]. In the ATE world the terms “accuracy” and “precision” are normally summarized as accuracy (e.g., edge placement accuracy in the following section). This means that for a specified accuracy, like the threshold voltage from above, we expect its validity for all tester channels on all test systems. We should never forget the statistical nature of measurement events and focus on a single measurement, which could accidentally lead to perfect accuracy with zero deviation.

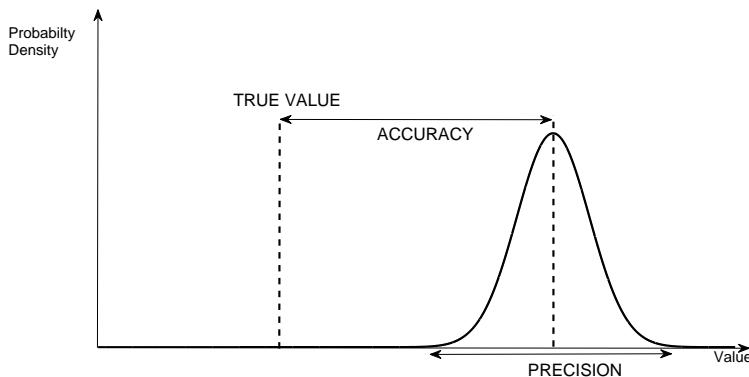


Figure 9.1 Comparing the definition of accuracy and precision for a measured value in regards to its true reference value.

9.1.2 Understanding OTA and EPA

Overall time accuracy (OTA) and edge placement accuracy (EPA) are two important specifications for an ATE system. OTA refers to the worst case error between two drive or receive actions (or edges) in all the drive and receive channels of the ATE system. On the other side, the EPA refers to the worst case error that a single drive or receive action can have in any drive or receive channel of the ATE system. The definition of OTA and EPA is shown graphically in Figure 9.2 where all ATE channels have one single drive or receive edge that was programmed to occur at the same time instant but due to the timing accuracy of the ATE system, each drive and receive edge will have a certain timing error. In the figure the channel 1 drive edge is used as the reference. Note that any channel can be used as a reference without any impact on the measured EPA value.

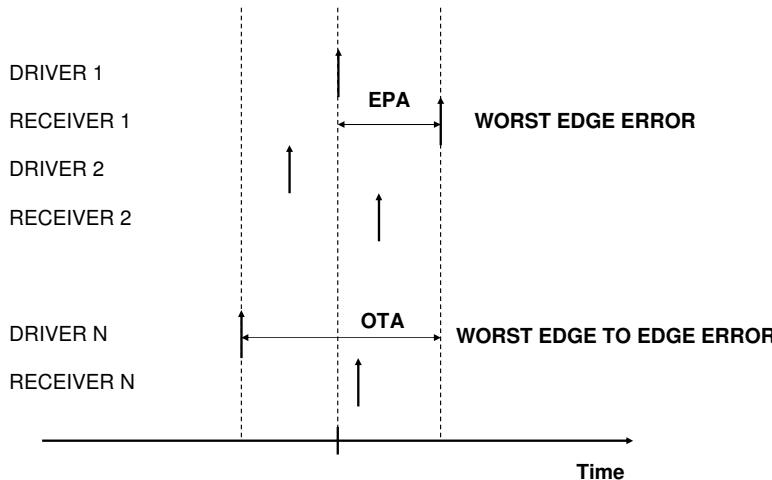


Figure 9.2 Diagram showing the OTA and EPA definitions.

Usually the EPA value is defined as a signed value (e.g., $+/- 60$ ps) because typically ATE manufacturers will define the maximum achievable edge error to be the same for either a positive or negative difference to the programmed value. If one assumes that the EPA of the driver edges is equivalent to the EPA of the receiver edges, which by itself is a significant assumption, the OTA value can be computed as twice the EPA value (i.e., $OTA = 2 \times EPA$).

Measuring OTA based on its definition can be very time consuming for ATE systems with a large number of channels, especially if the large number of different parameters that can have an influence on the OTA of an ATE system is taken into account. Reference [2] is an ANSI standard that presents a procedure for measuring the OTA of an ATE system. To address the OTA measurement challenge for a large amount of channels, the standard describes some time saving strategies like using a loopback between the driver and receiver of each channel for most of the measurements instead of measuring one ATE channel against all others.

9.1.3 Linearity and Edge Placement Accuracy

Another important difference that needs to be highlighted is the difference between linearity and edge placement accuracy. To understand the difference between these parameters, Figure 9.3 describes how the timing of a drive or receive edge of an ATE pin electronics channel is programmed at a very high level.

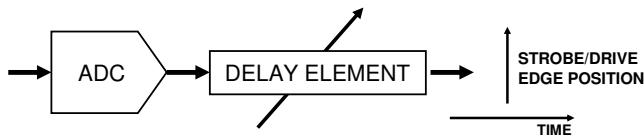


Figure 9.3 Conceptual block diagram of a driving/strobing edge delay programming.

The value to be programmed for an edge timing is transferred to an analog voltage provided by an ADC that controls a delay line which sets the timing of the edge. Any offset error (i.e., an error that is independent of the programmed values) will have an effect on the EPA which is important for measurements like the skew between two channels. For measurements like a jitter histogram, however, a fixed error is negligible since it only has influence on the mean value of the distribution but not on its peak-to-peak or variance value. For such self-related measurements, only the errors between programmed values are crucial as shown in Figure 9.4. In the presented architecture this kind of inaccuracy is caused by nonlinearities of the ADC (gain errors) or the delay element.

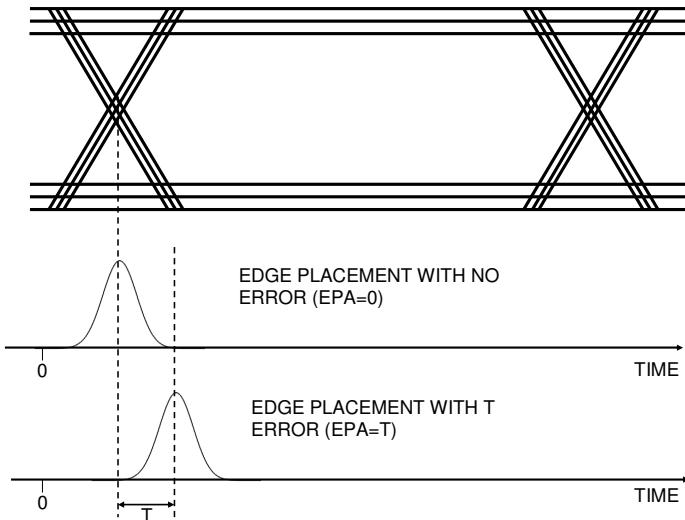


Figure 9.4 Comparison of the edge placement accuracy influence in a jitter histogram measurement.

One way to visualize the linearity of a drive/receive channel is to execute a shmoos (see Section 5.3) of the drive and compare timing for each ATE channel with pairs of an ATE driver and receiver connected together (ATE driver to receiver loopback) as shown in Figure 9.5. If the linearity of the

ATE pin electronics is perfect, then one would expect a straight 45 degree transition boundary from the pass to the fail region in the shmoo plot. Reality, however, shows that this transition is not perfectly linear and sharp and reveals the nonlinearity of the driver and receiver timing.

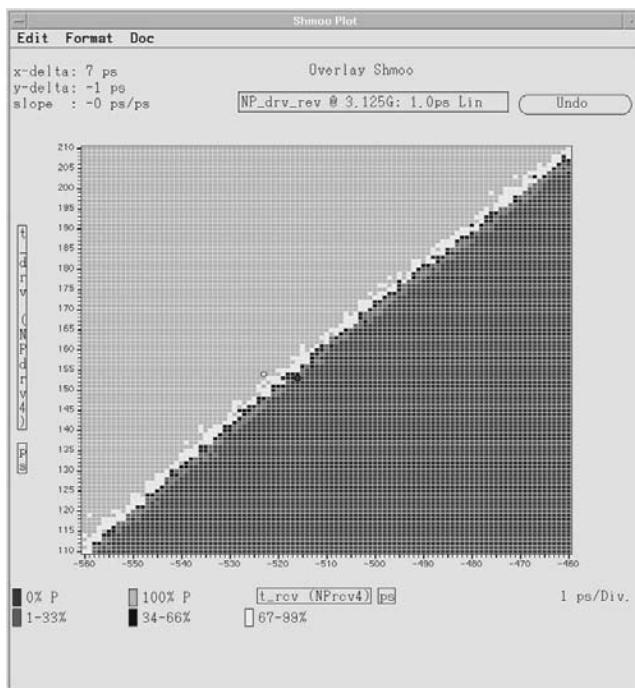


Figure 9.5 Example of a shmoo plot of a driver to receiver loopback for evaluating the driver/receiver edge linearity (courtesy of Verilog).

9.1.4 Calibration

If we discuss calibration in an ATE-based test solution environment, the calibration of the ATE system itself is only one aspect. The other aspect is the application specific part of the overall solution that changes with each device to be tested. The most critical component of this application specific part that has influence on test accuracy certainly is the test fixture and potential external instrumentation. ATE manufacturers acknowledge this fact and typically structure their calibration tooling in a way that the system calibration which guarantees the performance up to the test fixture interface is separated from the test fixture specific calibration tasks.

System Calibration

Calibration of an ATE system is by itself a large and complex topic that is very dependent on the ATE platform and pin electronics specifics. One challenge for ATE calibration is to determine the right compromise between the number and accuracy of the calibration steps and the time required to calibrate an ATE system. This is especially important in modern ATE systems with several thousands of pins that need to be calibrated. As a further complication of the situation, there are always calibration steps that cannot be done in parallel on all pins because they might require a unique external reference or measurement unit to be connected to each channel. Due to cost reasons, this type of calibration step needs to be performed serially, one channel at a time. The large number of channels that modern ATE systems contain already demand the use of robotic calibration units as the one shown in Figure 9.6.

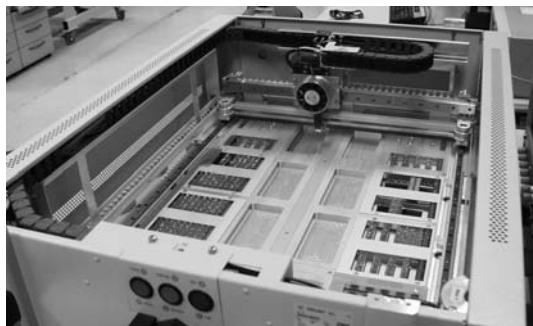


Figure 9.6 Picture of a robotic calibration unit docked to an ATE system with the top cover removed, showing the calibration robot (courtesy of Verigy).

To reduce the calibration time of an ATE system, parameters that need to be calibrated but are considered to be stable over time might simply be calibrated only at manufacturing time. The resulting calibration data is stored closely linked to the hardware it corresponds to (e.g., in a nonvolatile memory on the pin electronics ATE card). This approach avoids the need to calibrate those parameters on a regular basis and eliminates the corresponding calibration time from maintenance calibrations. This procedure is referred to by some ATE manufacturers as trimming.

An important distinction exists between the standard calibration of an ATE system and its performance verification (PV). Typically the ATE manufacturer will perform a PV of the ATE system after manufacturing and calibration by measuring the parameters that are part of the ATE system specifications. The measurement allows the assessment of the real performance of the ATE system, especially how much margin it has with

regard to its specifications. This type of data across multiple manufactured ATE systems provides the ATE manufacturer with an insight into whether the ATE specifications are too loose or whether additional margin is required. Some ATE manufacturers allow the purchaser of an ATE system to get access to the PV data of the delivered system.

The system calibration that needs to be executed in regular time intervals to keep the ATE system within the specified accuracy boundaries is typically divided into a DC and an AC calibration. The DC calibration ensures that the specified voltage levels and current drive capabilities for all ATE resources are valid. The main task during this calibration step is to measure the characteristics of analog-to-digital (ADC) and digital-to-analog (DAC) converters. The result of these measurements is used to determine the required compensation parameters that allow adjustment of the DACs and ADCs to their target behavior. It is important to note that DC calibration not only covers the typical DC resources of an ATE such as device power supplies (DPS) and parametric measurement units (PMU) but also the level generators for digital drivers and receivers.

The AC calibration takes care of the adjustable timing parameters that are specified for the respective ATE system like for example all specified flavors of skew values or timing linearities. Different ATE system architectures require the measurement of very different resources to achieve the specified timing values. Also a substantial subset of the timing values is typically guaranteed during PV measurements and does not require regular recalibration due to the stability of these parameters over time. The compensation of the timing errors determined during the AC calibration typically also involves DACs that are used to control timing delay lines in the ATE pin electronics.

Fixture Calibration

As discussed in Chapter 8, the test fixture is a critical part of the overall ATE system performance for high-speed digital applications. This means that for high-speed digital applications in some cases the standard ATE calibration is not sufficient and additional application specific calibration steps are required to compensate the accuracy impact of the test fixture. These calibration steps can be separated into components that are the same for all test fixtures and into components that need to be adapted to the operating conditions of a DUT. The latter of these typically are handled by focus calibration procedures as discussed in Section 9.3.

The most prominent test fixture specific calibration step that is independent of the DUT to a large extent is the fixture delay calibration. This

calibration determines the electrical length for each of the different signal paths of the test fixture. With the knowledge of these values, the ATE system can compensate the fixture delays for each signal path. This allows the referencing of the timing system of the ATE to the pins of the DUT which is not possible if there are unknown differences on the electrical lengths of the test fixture signal paths. While the fixture delay compensation mechanism is a standard part of ATE systems, the way to obtain the fixture delay values might vary from ATE system to ATE system. For ATE systems that do not have integrated support for fixture delay measurements, either external equipment is required for this measurement, the test fixture has to be designed with identical signal path lengths for all connections, or the electrical delays are determined by simulation. Especially for high-speed applications it is difficult to achieve the required matching accuracy by design over a large amount of signals and the accuracy of the simulation approach always will depend on the quality of the simulation models.

Thus, the extraction of the fixture delay values by measurements typically is mandatory for high-speed I/O applications. Fixture delay values can be derived from skew measurements because the ATE channels are typically deskewed up to the test fixture already by the system calibration. This, however, will require signal probing in the test socket as for a focus calibration. Since automated probing in the socket is difficult to implement and will result in expensive instrument additions (e.g., a probing robot), fixture delay values typically are measured via TDR (see Appendix E). This method is implemented by contacting the test fixture at its interface to the ATE with a single contact point per measurement. Another advantage of the TDR methodology is that in addition to being executed using dedicated TDR equipment, it can also be done with bidirectional ATE pin electronics. The driver of the pin electronics is used to launch a voltage step into the test fixture trace and the receiver of the bidirectional frontend scans for the response to that step stimulus. The ATE receiver typically uses the methodology described in Section 4.1.9 to retrieve that response. The accuracy of the TDR result measured with an ATE pin, like for dedicated TDR equipment, depends first of all on the rise time that the driver can achieve and the receiver bandwidth. Due to the way the response is measured on the ATE receiver, also timing linearity and level threshold accuracy have an impact on the resulting accuracy.

In general, the ATE-based TDR measurement is possible into an open or a shorted socket. For dual transmission lines, the implementation of the dual transmission line (DTL) on the fixture determines whether a TDR into open or a TDR into short can be used. If a TDR into open is to be performed, an impedance adaptation is required at the DTL joint so that each of the three connections is loaded with an effective $50\ \Omega$ impedance. The drawback of

this implementation is that the level swing of the response to the measurement step is reduced, which has an impact on the accuracy for the measurement of the response at the ATE receiver. Thus, a DTL topology with a direct DTL connection that requires a TDR into a short as shown in Figure 9.7 typically results in more accurate fixture delay values.

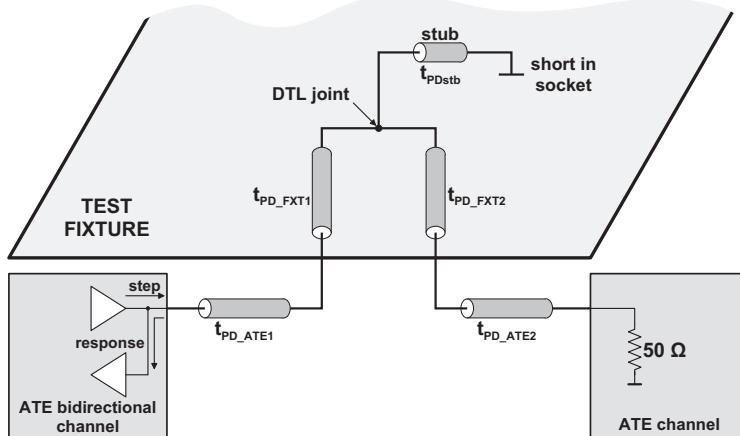


Figure 9.7 Dual transmission line structure with shorted socket pin.

A critical parameter for this kind of DTL topology is the electrical length of the stub from the DTL joint to the socket tip. This is due to the fact that the joint represents a discontinuity and reflections go back and forth on the stub between the socket tip and the DTL joint when the TDR measurement is done. The shorter the stub is, the less accuracy impact these reflections have on the measured fixture delay result.

Another parameter that influences the level of reflections on the stub and thus the accuracy of the shorted TDR measurement is the quality of the short in the socket. The lower impedance this short has, the more accurate measurement results are possible. Good low impedance shorts in a device socket are usually achieved by using gold plated device package dummies in the socket during the shorted TDR measurement as shown in Figure 9.8.

9.2 Multiplexing of ATE Channels

One typical approach to achieve higher data rates on ATE is to multiplex several low-speed ATE channels into a single high-speed digital signal [3]. This approach can be very appealing from a cost point of view since it makes use of low-speed ATE equipment to test higher data rate I/Os. For

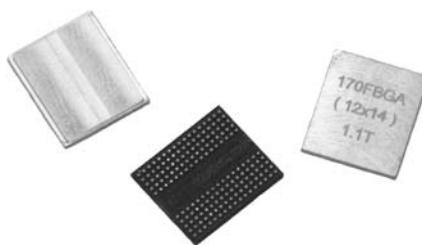


Figure 9.8 Device with corresponding gold plated package dummies (short device).

example, this could allow the use of an already available ATE system to test an application above its maximum data rate without buying a new ATE system or higher performance ATE cards.

The challenge with this approach is to design the multiplexing circuitry on the DUT test fixture, especially for multigigabit digital applications [4, 5]. ATE channel multiplexing can be divided into two major categories, standard and retimed channel multiplexing.

Standard Channel Multiplexing In this case a logic gate (e.g., a XOR) is used to generate a high-speed signal derived from the logical combination of several low-speed ATE channels. The ATE channels are skewed in time so that at the output of the logic gate, the waveform is running at the speed of the ATE multiplied by the number of multiplexed channels as shown in Figure 9.9.

Retimed Channel Multiplexing In this approach the signal derived from the logic gate multiplexer is retimed using a clean reference clock that runs at the same rate as the multiplexed signal. Retiming removes jitter from the multiplexing operation, and the jitter of the output signal is only defined by the jitter of the retiming circuitry.

Another major challenge with channel multiplexing is related to the receive side of the ATE. Creating a receiver on the test fixture that is integrated into the ATE timing system and then demultiplexing the DUT output data to the low-speed ATE channels can be a tough task. One option is to use a non-symmetric ATE configuration where the receive instrumentation (e.g., digital pin electronics or sampler) have enough bandwidth to handle the application data rate, while using low-speed ATE channels on the drive side. For the DUT

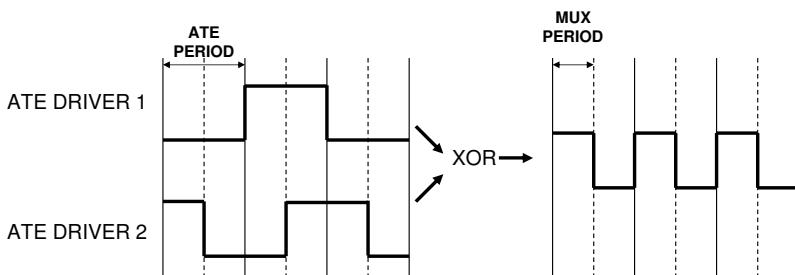


Figure 9.9 Multiplexing of two ATE driver channels using a XOR logic gate.

stimulus side, the ATE channel multiplexing approach provides the digital signal with the required data rate.

9.3 Focus Calibration

A commercial ATE system needs to serve a broad range of different applications but it is defined by a restricted set of specifications. This creates the situation that for some high-speed digital applications, the ATE specifications guaranteed by the standard calibration process are not sufficient. One possibility to address this challenge is to use the knowledge of the specific requirements of the application for which one intends to use the ATE equipment, and improve on the ATE specifications by the use of a focus calibration procedure.

Focus calibration can be applied to address several issues [6]. One possibility is to verify whether an ATE system is able to address a set of specifications that are outside the guaranteed set for the ATE system. For example, ATE manufacturers must define the edge placement accuracy for the entire ATE system taking the maximum possible configuration which corresponds to the worst case scenario into account. But if for a specific application only the EPA of a set of eight pins that stimulate an I/O bus of the DUT is important, and if those eight pins belong to the same ATE pin electronics module, it is likely that the EPA of those eight pins is significantly better than the specified EPA for the entire ATE system which is of little meaning for the application.

In this scenario, since the “improved” specifications cannot be guaranteed by the ATE manufacturer, it is important to measure the specification with the specific application setup for those eight pins. That is why although no focus calibration is required in this example, a focus measurement setup might be necessary for verification purposes.

In other situations, the ATE or measurement instrumentation manufacturers do not provide the necessary calibration for a given application measurement. This is especially true for high-speed digital applications where the DUT test fixture can have a significant influence that is not considered in the standard ATE calibration. This section presents three types of focus calibration items that are common for high-speed digital applications. These focus calibration items are skew calibration, data eye height calibration, and jitter injection calibration. The concept of data eye profile that provides a methodology to analyze the influence of the data rate on the data eye diagram parameters will also be discussed.

9.3.1 Skew Calibration

One very important test for some high-speed interfaces is the measurement of the minimum setup/hold time (Section 5.6.1). The accuracy of this measurement depends on the ability to guarantee the relative timing between the different ATE pins. Not only is there an inherent error from the pin electronics, but there is also an error induced by the different electrical lengths on the test fixture. One can minimize the effect of the electrical lengths in the test fixture by matching the signal trace physical lengths but this does not guarantee a perfect timing skew matching.

One approach to address this issue would be to measure the specific timing of each ATE pin used on the test application including the test fixture. This can be accomplished by using external instrumentation with the required accuracy to measure the ATE timing [7, 8]. The idea would be to correct the ATE timing with correction factors until the timing between all channels required for the application are within the applicable specifications (note that these correction factors might depend on variables like the programmed data rate). This might even be an interactive process where calibration values are measured and programmed on the ATE system and measured again in order to obtain more accurate calibration factors. Figure 9.10 shows a high-level diagram of the procedure. This approach could also be used to verify the OTA/EPA of an entire ATE system.

9.3.2 Data Eye Height Calibration

Another important specification that might lead to a focus calibration procedure is the data eye height of the ATE driver or receiver at the DUT [9]. This specification is very important in tests like the DUT receiver sensitivity or a data eye height measurement. The problem is that due to the frequency dependent loss inherent to any test fixture, the data eye height at the DUT will not correspond to the expected value (the one programmed in the ATE

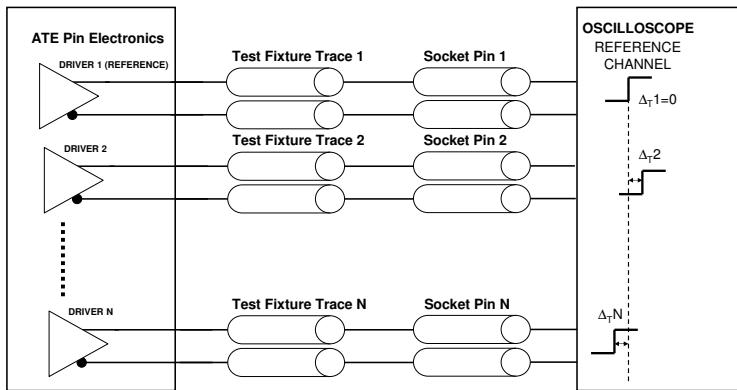


Figure 9.10 High-level diagram of a focus skew calibration measurement setup.

software) due to the signal path attenuation that results from added level jitter in the case of a data pattern. To make this problem even more complex, the resulting data eye height will depend on the data rate and pattern as shown in Figure 9.11. The same reasoning applies to measuring the data eye from the DUT with an ATE receiver.

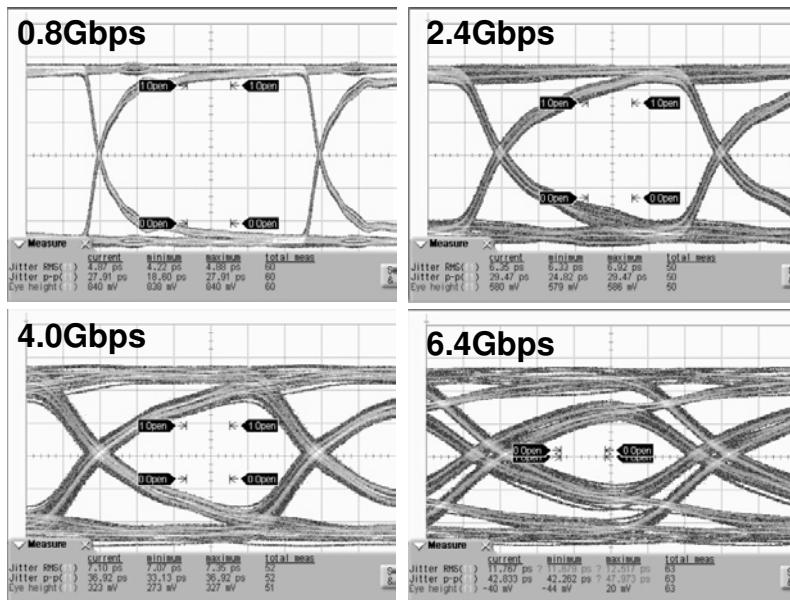


Figure 9.11 Example of the degradation of the data eye with increased data rate for a standard calibrated ATE system at the end of the test fixture.

This problem can be significantly reduced by using techniques like equalization (Section 9.8.3). Another option is to simply overdrive the levels of the stimulus signal to obtain the desired data eye height at the DUT. Figure 9.12(a) shows a possible setup for an ATE driver eye height focus calibration measurement using an external oscilloscope and Figure 9.12(b) shows the setup for an ATE receiver eye height calibration where a pattern generator is used to provide a reference data eye to be measured by the ATE receiver.

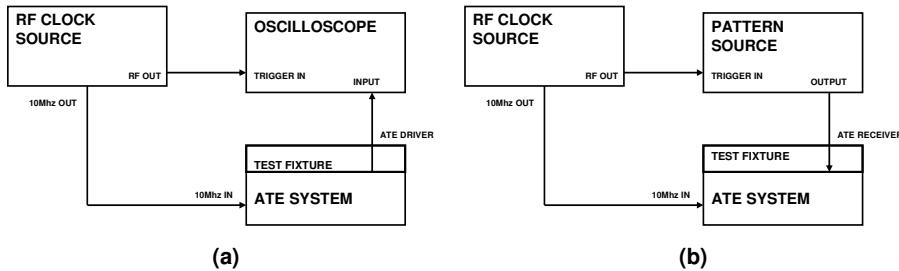


Figure 9.12 Possible approach (a) for an ATE driver data eye height focus calibration setup and (b) for an ATE receiver data eye height focus calibration setup.

Since the obtained calibration factor will be data rate and pattern dependent, the measurements have to be executed at the different data rates and with the patterns required by the application as shown in Figure 9.13 where an example for a lossy test fixture is shown. It is important to notice that this graph also can provide detailed data on the test fixture performance when comparing the focus calibration results with the measurements done without a test fixture. For the data eye height case, the calibration factors are obtained from the inverse of the measured data eye height curve as shown in Figure 9.14.

9.3.3 Jitter Injection

Jitter injection is another important parameter that benefits from a focus calibration approach to compensate for the test fixture effects, especially the data dependent jitter (DDJ) that is added by the test fixture due to its loss. In a receiver jitter tolerance test, it is typical to inject a certain amount of deterministic jitter (DJ) in the form of periodic sinusoidal jitter on the stimulus waveform to close the data eye by a certain amount. Even if the programmed amount of injected jitter is calibrated at the ATE pogo pin interface, the added DDJ from the test fixture will further reduce the date eye opening and increase the amount of DJ to values beyond the test limits for the receiver jitter tolerance. One solution is to calibrate the injected jitter value by measuring it

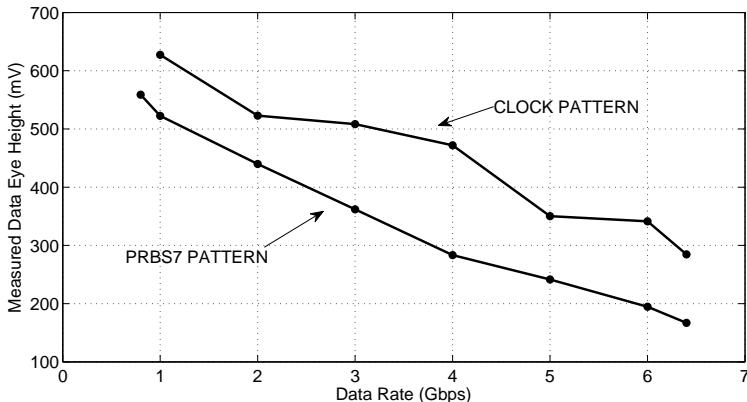


Figure 9.13 Example of the data eye height profile for the driver data eye height of an ATE system including a lossy test fixture. (From: [6]. ©2009 Jose Moreira. Reprinted with permission.)

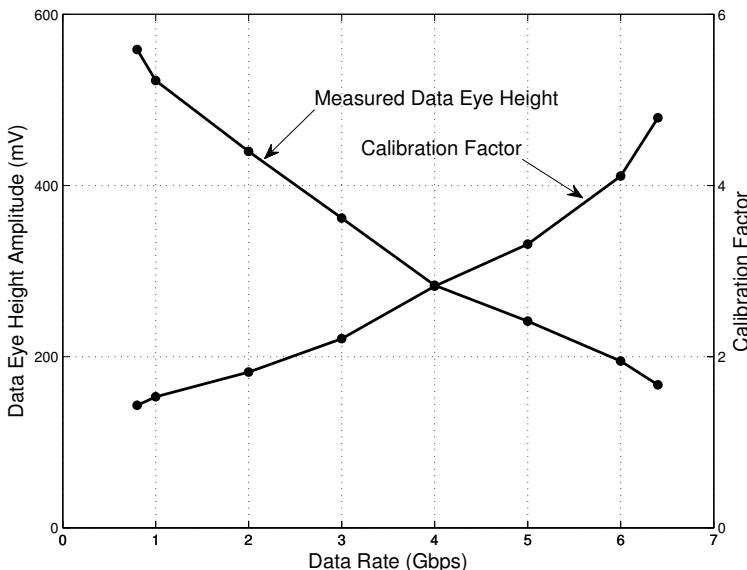


Figure 9.14 Example of the resulting focus calibration plot for the measured ATE driver data eye height at different data rates and the corresponding calibration factor curve to achieve an 800 mV target data eye height at the DUT.

at the test fixture DUT socket. Figure 9.15 shows one example of a measured focus calibration curve for jitter injection on an ATE system with a lossy test fixture. This curve can then be used for injecting a calibrated amount of jitter at the DUT by only injecting the sinusoidal jitter amplitude that is required to achieve the desired peak-to-peak jitter amplitude at the DUT. For example, to achieve 85 ps of peak-to-peak jitter at the DUT, only 40 ps of sinusoidal jitter should be injected by the ATE pin electronics on the driver signal. The remaining jitter will be added by the test fixture loss.

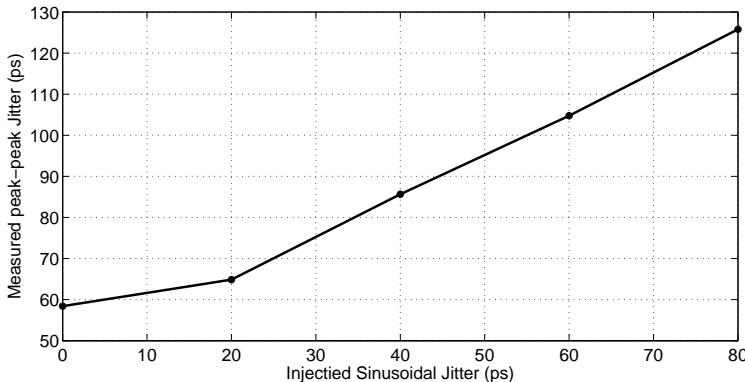


Figure 9.15 Example of the resulting focus calibration plot for the measured ATE driver jitter injection calibration when using 1 MHz injected sinusoidal periodic jitter for a PRBS $2^7 - 1$ data pattern at a data rate of 4 Gbps.

9.3.4 Data Eye Profile¹

The concept of the data eye profile [6] is tightly connected to the characterization and focus calibration of an ATE system. As discussed in the previous subsections, the parameters that characterize the ATE channel performance, including the test fixture, will change depending on the data rate the ATE system is running at. This means that to properly characterize the performance of an ATE system at the DUT, including the test fixture, some of the characterization measurements will be data rate dependent. There are multiple parameters that can be defined to characterize the high-speed performance of an ATE system at the DUT. In this section we will use the following measurements:

Effective data eye height: The vertical data eye height at the middle of the data eye.

¹In collaboration with Bernhard Roth.

Nominal data eye height: The vertical data eye height 0.5 UI away from the time zero calibrated point (note that in systems with multiple channels, this time zero is the same for every ATE channel).

Maximum data eye height: The maximum eye height one can achieve at any point on the data eye.

Timing error (max and min): This value corresponds to the worst case timing offset for a given bit transition which is before the calibrated time zero for the entire system (timing error max), or after the time zero (timing error min). The difference between these two values will correspond to the peak-to-peak deterministic jitter value. For an ATE system this value is directly related to the edge placement accuracy.

Timing shift: This value corresponds to the difference between the calibrated time zero and the average of the maximum and minimum timing error defined on the previous bullet.

Jitter: This is the peak-to-peak deterministic jitter value which corresponds to the maximum timing error variation.

Figure 9.16 shows one example of a data eye profile for the measurements defined above that are plotted in relation to the ATE system data rate. These graphs provide a profile of the ATE driver data eye height, width, and jitter at the DUT with regard to the data rate [10].

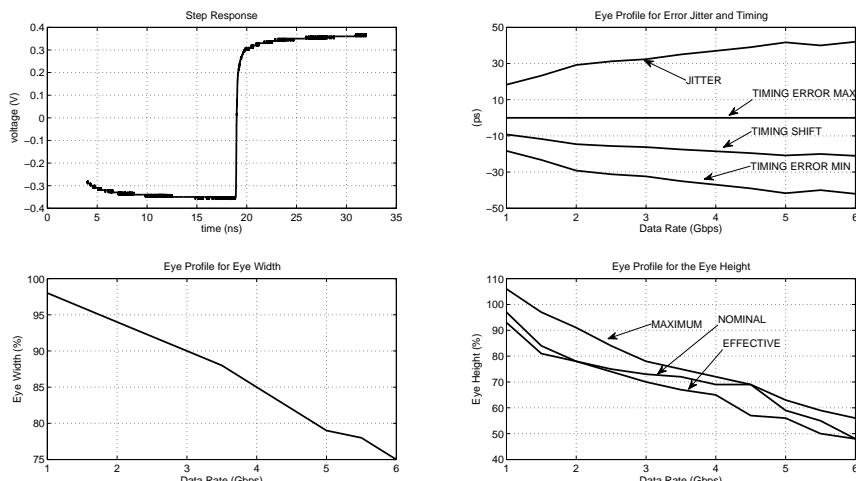


Figure 9.16 Example of a data eye profile obtained through simulation using a measured step response.

The data eye profile cannot only be measured but also derived from simulation based on an ATE and test fixture model or on measured data like a step response. Several options are available to simulate the data eye profile depending on the types of models that are available. One approach that allows for fast simulation time with minimum requirements on the model is to use the measured step response of the ATE system and then generate the data eye profile using techniques like double-edge response (DER) [11]. This was the approach taken in Figure 9.16.

The data eye profile presented in Figure 9.16 takes into account only the DDJ jitter added by the test fixture with the increased data rate based on the step response. It does not include the timing jitter and amplitude noise that is present on the ATE driver. It is possible to include this timing jitter and amplitude noise on the data eye profile generation. This is shown in Figure 9.17 which compares the generated data eye profiles with and without any timing jitter and amplitude noise.

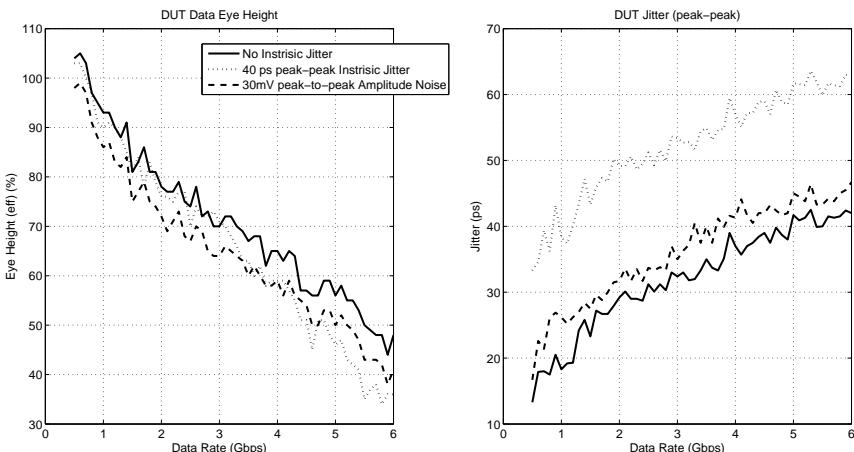


Figure 9.17 Example of a data eye profile computation including the ATE driver timing jitter and amplitude noise for the data eye height (left) and the peak-to-peak jitter (right).

Another extension of the data eye profile methodology is the concept of data eye profile correlation in a multisite application. In this case a data eye profile is generated for each site based on the measured or simulated step response of each site and the difference between the two sites with the most extreme values is computed. In an ideal multisite setup, the difference should be zero for all data eye profile measures. Figure 9.18 shows the step responses and the computed data eye height profiles for four DUT positions in a multi-site test fixture. This data is used to generate the data eye height correlation

plot for these four sites as shown in Figure 9.19. In this example there is a difference in the data eye height between the sites showing that there is some headroom in obtaining a better correlation between sites (e.g., by improving the test fixture).

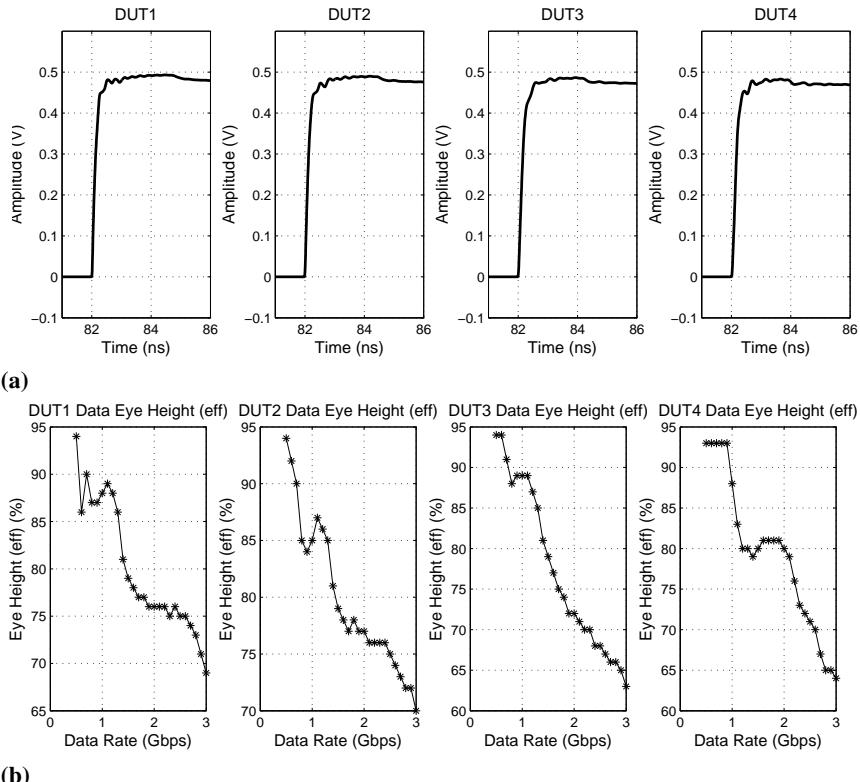


Figure 9.18 Example of (a) the measured step response for a signal pin in each site of a four-site test fixture and (b) the corresponding data eye profile for the data eye height.

9.4 Testing of High-Speed Bidirectional Interfaces with a Dual Transmission Line Approach

To measure bidirectional interfaces with semiduplex data transmission, one requires that the ATE pin electronics is capable of handling semiduplex signals. If this is not the case, one needs to connect the ATE driver and receiver on the test fixture to create a virtual semiduplex interface on the ATE side. The

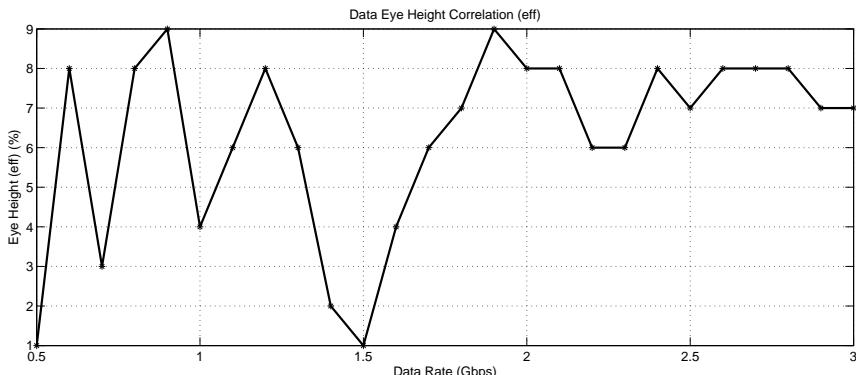


Figure 9.19 Data eye height profile correlation result for a multisite test fixture using the data eye height profiles from Figure 9.18.

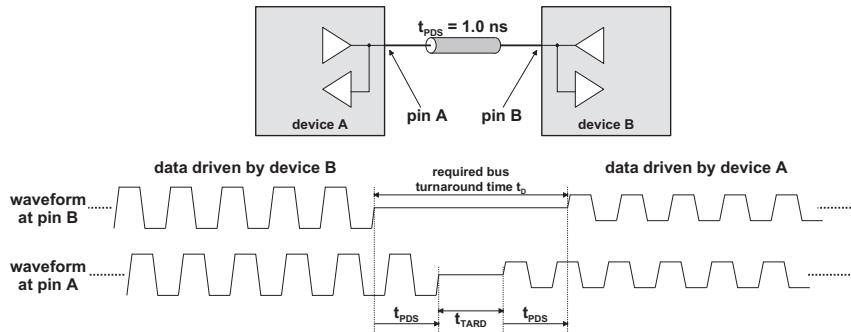
use of separate drive and receive channels in such a configuration is known as dual transmission line or fly-by configuration [12].

The use of DTL configurations is even sometimes required if the pin electronics has native handling of semiduplex signals. The requirement that might drive such a configuration is the test of the minimum bus turnaround time. The bus turnaround time defines the time a device takes to switch the data transmission direction on its bidirectional pin(s). Different connection configurations might be required for the usage of a device in its final system environment and in its ATE test environment because the connection length between the ATE pin electronics front end and the DUT pins is significantly longer than the connections between two devices in the final system. The electrical length or propagation delay of this connection, however, defines the minimum bus turnaround time that can be applied without causing data collisions on the data connection.

The critical case on the ATE in this regard is to test the bus turnaround time of a DUT from its drive to its receive state as shown in Figure 9.20. Let's assume we have a device (device B) in its system environment with a propagation delay of $t_{PDS} = 1.0$ ns to its partner device (device A). The partner device has a receive-to-drive bus turnaround time of $t_{TARD} = 1.0$ ns. In such a configuration, the delay t_D between driving the last bit of data to the partner device and receiving the first data bit from the partner device is $t_D = 2 t_{PDS} + t_{TARD} = 3.0$ ns because the drive data has to travel the connection completely before the partner device starts turning around its bus. After bus turnaround completion on the partner device, the data stemming from the partner device also has to travel the connection before the receiving device sees that data. Thus, the minimum drive-to-receive bus turnaround time

t_{TADR} the device has to support needs to be less than t_D .

SYSTEM ENVIRONMENT



ATE ENVIRONMENT

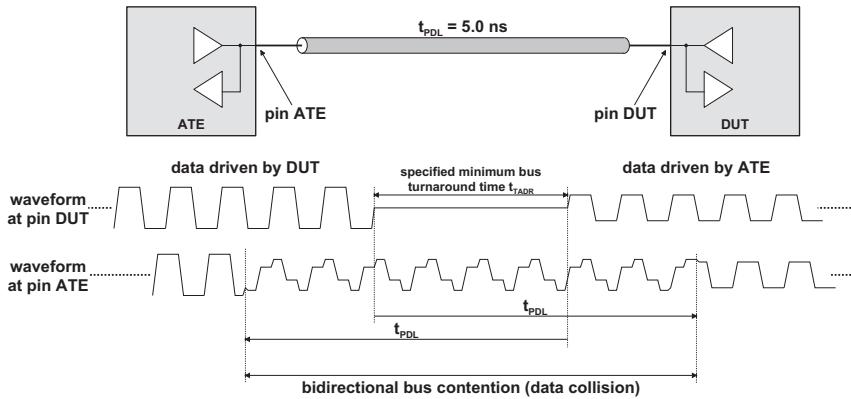


Figure 9.20 Data collision when trying to test a bidirectional application with a long test fixture [13].

As already mentioned, the propagation delay t_{PDL} typically is significantly larger in an ATE environment than in the system environment. ATE systems usually compensate this propagation delay by generating drive data earlier and compare receive data later by the propagation delay between the pin electronics front-end and DUT pin. With this compensation, ATE systems can reference the timing values programmed by the ATE user to the DUT pins which is required for all timing measurements but which also makes it less obvious to the user that such a propagation delay exists.

For the measurement of the drive-to-receive turnaround time of a semiduplex DUT pin, of course the propagation delays of the ATE environment contribute to the overall turnaround delay time t_D that can be achieved

at the DUT pin in the same way as in the system environment. With the longer propagation delay in the ATE environment, the achievable t_D also becomes significantly larger than in the system environment and in most cases exceeds the minimum bus turnaround time at the DUT pins that has to be ensured by the test. If the minimum bus turnaround time for a DUT is programmed on the ATE and this minimum bus turnaround time is less than two times the propagation delay, the ATE driver already will start sending data while there is still data to be received by the ATE on the transmission line between pin electronic frontend and DUT. Thus, the ATE receiver will see a signal that is generated by an overlay of the ATE drive data and the data stemming from the DUT as shown in the comparison between the system environment and the ATE environment in Figure 9.20.

Traditional ATE receivers cannot extract the data sent by the DUT from this overlayed signal and thus cannot do a correct comparison to expected data on the data part that is affected by such a data collision. One solution to such a data collision problem would be to consider the ATE propagation delays during test pattern generation and insert dead-zones that are long enough between the affected signal direction changes. This, however, automatically would prevent the test of minimum bus turnaround times. Another solution is the use of a DTL configuration where the ATE driver and the ATE receiver are connected to the DUT pin via physically separated transmission lines that only have a connection at the DUT pin. This configuration prevents data collisions on the transmission lines that affect the compare capability of the ATE receiver.

Figure 9.21 shows a DTL setup using two ATE channels. On the receiving channel, the channel termination to a termination voltage is enabled to prevent signal reflections to travel back to the DUT. Also the driving channel provides a termination voltage on its driver during pattern sections that contain ATE compare data to absorb the DUT signals that travel down the transmission line connecting the ATE driver to the DUT. Section 8.8 discusses in more detail the impedance discontinuity that arises from connecting two $50\ \Omega$ transmission lines.

Clearly, a major drawback of the DTL configuration is the need to use two ATE channels to test a single DUT I/O channel. The other drawback is that due to the parallel $50\ \Omega$ terminations on the driver and the receiver during compare actions, the signal amplitude at the ATE receiver is reduced. The same is true for the signal amplitude at the DUT during ATE drive actions due to the parallel $50\ \Omega$ terminations in the DUT and the ATE receiver during ATE drive actions. This is shown in the fly-by simulation in Figure 9.22. This is not a problem if the ATE driver has enough margin to compensate for the loss during drive actions and the ATE receiver has sufficient accuracy to also

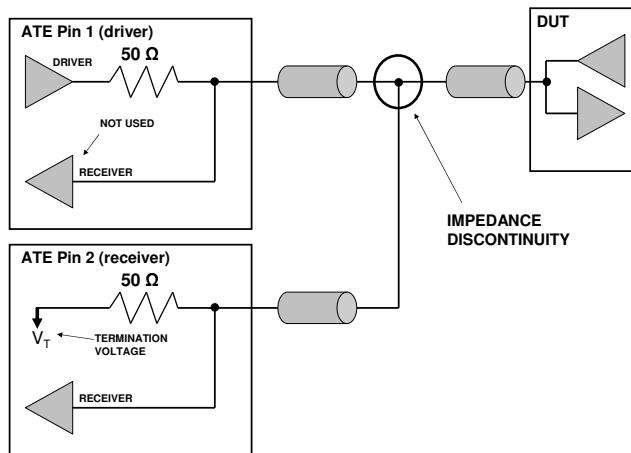


Figure 9.21 Bidirectional application being tested with a DTL (fly-by) configuration.

recognize the lower amplitude signal during ATE receive actions correctly. However, in some situations this might cause test problems. The challenges for test fixture design with a DTL type layout are discussed in Section 8.8.

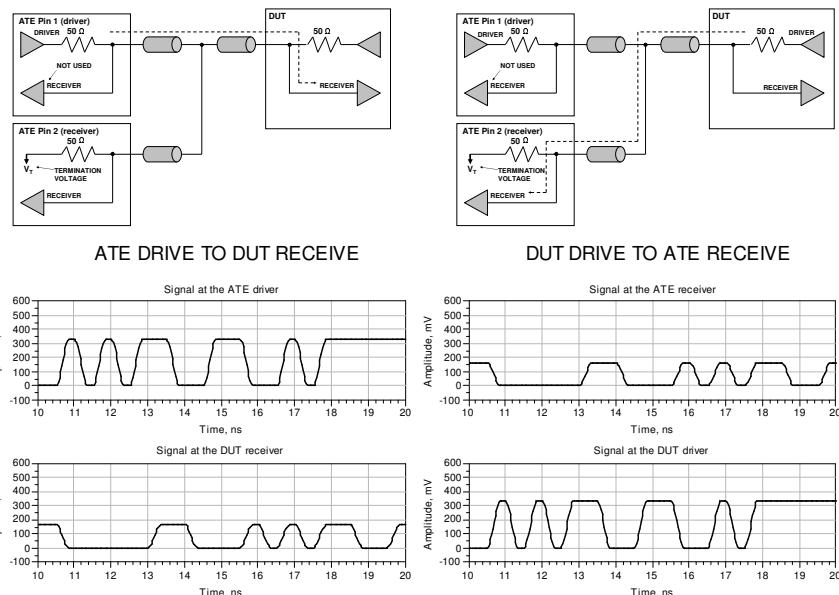


Figure 9.22 Simulation of the loss of the ATE driver signal at the DUT for a bidirectional test setup using fly-by wiring.

Note that simultaneous bidirectional (SBD) pin electronics architectures (see Chapter 4) do not have this drawback and can recover the correct data sent by the DUT on their receivers even in the data collision case. Thus, semiduplex signals can be tested using single transmission line configurations even with long test fixture trace lengths if the pin electronics support SBD operation modes.

9.5 Including the DUT Receiver Data Recovery in Driver Tests

The jitter generation measurements presented in Section 5.5 have the limitation that they do not take into account how the receiver in the link partner works. For some high-speed interfaces, the link partner receiver typically has either a complete clock data recovery (CDR) unit or at least a PLL in its sampling clock path. The jitter seen by the receiver after the CDR or PLL can be different from the jitter sent by the DUT driver as shown in Figure 9.23. A more detailed discussion on the background of this behavior can be found in Section 2.6.3.

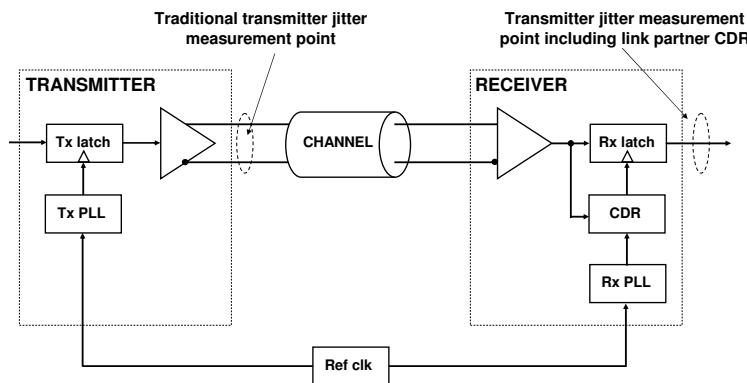


Figure 9.23 Difference of measuring the generated jitter by a DUT driver at the driver output and at the output of the link partner CDR.

Some standards (e.g., PCI Express) require that jitter tests (e.g., jitter generation) are performed using a reference or golden CDR or PLL that is defined by the standard. This allows the jitter generation specification to take into account the capabilities of the link partner receiver CDR/PLL to attenuate certain frequency ranges of the jitter magnitude.

Example Let us assume that a DUT test requires the data eye total jitter (at a given BER) to be less than 0.5 UI. Let's also suppose that the DUT driver has a

design problem that creates a periodic jitter component (e.g., sinusoidal) with 1 UI amplitude at a very low jitter frequency (e.g., 10 kHz). On a standard jitter test without any CDR unit, this part would fail the jitter generation test. Since a typical receiver CDR, however, can handle 10 kHz of sinusoidal jitter easily, this design problem would not have an effect on the real application. Thus, from the IC manufacturer's point of view it makes sense to use a CDR for the jitter measurement with a defined specification so that this DUT would pass the jitter generation test.

Figure 9.24 shows the CDR requirements from the PCI Express standard [14] for any jitter measurements. The figure shows that any jitter components below 10 kHz should be attenuated by a factor of 10^{-3} since these jitter components will be tracked by the CDR.

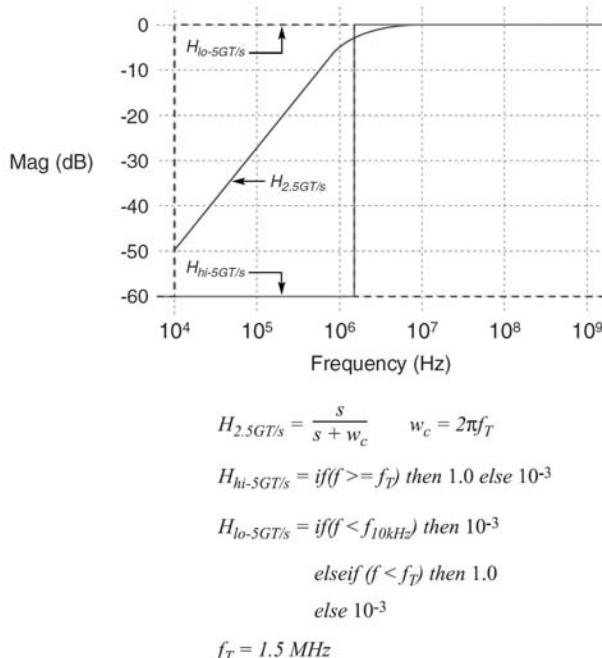


Figure 9.24 CDR requirements for the transmitted jitter measurement as defined by the PCI Express standard. (From: [14]. ©2002–2009 PCI-SIG. Reprinted with permission.)

If a CDR is really required for a jitter measurement, there are two ways to address this requirement. One option is to use a real-time sampling oscilloscope to measure the jitter. Since a real-time oscilloscope can acquire the entire wave form in real time, one could perform a software-based CDR

on the waveform and then compute the transmitted jitter value of the resulting waveform after the CDR operation. Although this is a very elegant solution because it allows the user to define the shape of the CDR response in a software postprocessing step, real-time sampling capabilities are typically not available in standard ATE pin electronics cards.

The second option is to implement the CDR on the measurement equipment hardware. This is currently the typical approach followed for ATE pin electronics cards. Although there is bench test and measurement equipment that provides the capability to define the characteristic of the CDR being used for the jitter measurement, typically what one finds is a fixed CDR unit that is able to track the low frequency jitter and in this way try to be compliant with most standard requirements.

9.6 Protocol Awareness and Protocol-Based Testing

The typical ATE-based functional test of a device applies binary data to the inputs of the DUT and compares the response of the DUT to expected data as shown in Figure 9.25. Traditionally the binary stimulus and expected data is derived from device simulations and the ATE system does not interpret or modify the test data beyond the single bit logical states as, for example, protocol analyzers do.

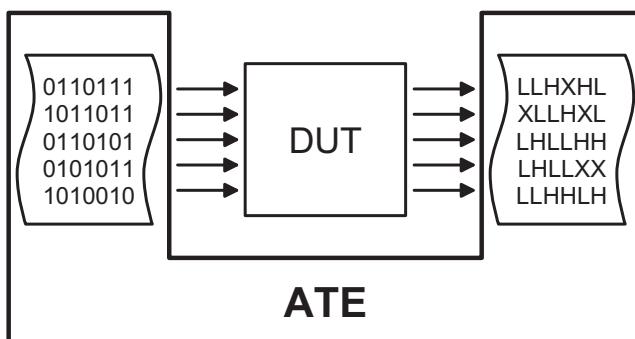


Figure 9.25 Classical view of an ATE functional pattern for a standard digital functional test.

Devices, however, operate based on protocols that define how data needs to be applied to a device and in which sequence this data can be applied. Of course, the devices also follow these protocol rules on the data they generate on their output pins. For low-speed parallel buses or very simple serial protocols as, for example, I2C [15], it was a relatively simple task for a test engineer to interpret the binary data representation available on ATE

systems to understand the functionality behind the test pattern. The ability to interpret this data is especially important during interactive device debugging to allow a test engineer to apply corrective actions without going through the complete simulation and pattern generation process for each debug step. This is of increased importance for the protocols used to program and query the configuration registers of a device.

With the advent of high-speed interfaces and especially the increased application of serial and embedded clock data transmission for these devices, this situation changed significantly. Serial interfaces typically use packet-based protocols that embed their payload like address data or transfer data into complex packets that span over several serial bytes as shown in Figure 9.26 for an example of one of the transaction layer packets (TLP) defined for PCI Express.

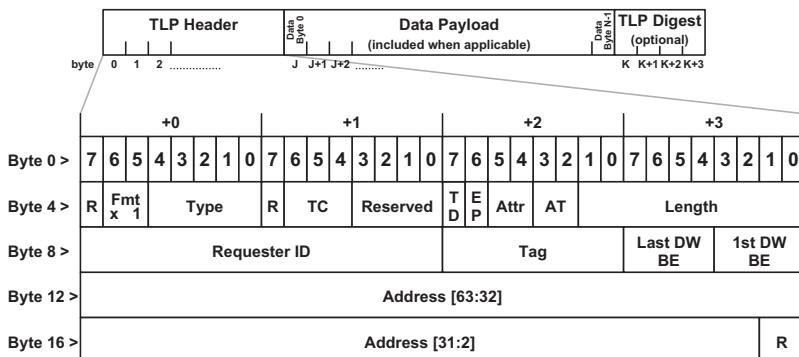


Figure 9.26 Example of a PCI Express transaction layer data packet (TLP) with TLP header for a 64-bit memory access. (After: [14].)

Moreover, such complex high-level packets might be repackaged into several lower level packets that can be distributed over a potentially scalable number of parallel lanes on the physical layer of a serial link. Each of these parallel lanes that form the logical link transmits its data serially.

In addition to the complex packet structure that makes interpretation of the data difficult for a test engineer, embedded clock interfaces usually apply coding like for example 8B/10B and/or data scrambling (see Appendix D). This makes the interpretation of the raw data during interactive debugging nearly impossible without additional support from the ATE system [16]. For production testing with debugged patterns, however, this does not pose a problem as long as the data coding and scrambling generates unique data. For some codes, this is sometimes not necessarily the case. For example, the data generated by 8B/10B coding depends on the starting disparity that is set by the device. If a DUT that uses 8B/10B coding does not set a specific known

starting disparity during initialization, this can lead to varying data response codes of the device which can cause problems when the devices are tested on an ATE that only compares to predefined expected raw data bits.

Another issue for the debug scenario of high-speed I/O interfaces is caused by the nature of embedded clock interfaces. In order to keep the phase lock of a clock data recovery unit, the transmitted data stream needs to contain a sufficient number of data transitions as described in Section 2.6.3. This, however, means that in intervals with no transmitted data, a data connection cannot just be idled and therefore transitions need to be generated by the transmitter driving the transmission line. The relevant high-speed I/O standards usually define the exact data that needs to be transmitted in pauses where there is no payload data available (logical idle). For the SATA standard, this data, for example, is specified as SYNC primitive; for PCI Express, this data is called idle data symbol. The transmission of data-like patterns during these logical idle states makes it difficult for a test engineer to distinguish between payload and idle data during debugging.

The last protocol related test challenge to be mentioned here not only has consequences for device debugging on the ATE, but also for fully automated testing of devices based on predefined test patterns (e.g., in production test runs). With the clocking architectures used for systems that deploy high-speed I/O devices it is possible that two communicating devices are not driven by one central system reference clock, but by two separate reference clocks. Although these separate clocks usually run at the same frequency, there typically is a frequency offset that might be only in the parts-per-million (ppm) range. Since the phase of the reference clocks of high-speed I/O devices directly translates to the phase of the high-speed I/Os, the same offsets are seen on the data connections between the two devices. Thus, devices connected in such an environment see a slow drift of the incoming data relative to their reference clocks. With the data tracking mechanisms of source-synchronous interfaces, this drift is not an issue for sampling the incoming data correctly. However, in the device, usually the clock domains of the device core and the high-speed I/O data section are decoupled via FIFO buffers. On the core side, data is read and written into the FIFO based on the reference clock phase. On the I/O side, the data is read from the FIFO also based on the phase of the reference clock, but the received data is written into the FIFO based on the phase information derived from the high-speed I/O data interface.

If there is a frequency offset between the reference clock and the clock derived from the data interface with the derived clock running at a slightly higher frequency, this decoupling FIFO will fill up more and more over time until it reaches its stall state and does not have any buffer space available

to store received data. This will lead to data loss on the high-speed I/O connection. In order to prevent such a situation, especially high-speed I/O standards that are based on embedded clock data transmission, define so-called skip-data sequences that need to be transmitted regularly and that can be ignored (i.e., not transferred into the FIFO) on the receiving side of the data connection. Examples for skip-data in high-speed I/O standards are skip ordered sets for PCI Express or ALIGN primitives for SATA. The intervals in which skip-data needs to be transmitted are derived from the maximum frequency offsets that are allowed on the reference clocks by these standards.

The issue for ATE-based testing with this kind of frequency offset compensation is that the insertion intervals for skip data are not defined in a deterministic manner. This means devices usually generate skip data in a way that fulfills the specification, but which is not predictable. This means that it might not be possible to know *a priori* how the expected output bit stream from the DUT will look. This is exemplified in Figure 9.27.

IDLE	PAYLOAD A	IDLE	PAYLOAD B	IDLE	IDLE
SKIP	IDLE	PAYLOAD A	IDLE	PAYLOAD B	IDLE
IDLE	PAYLOAD A	SKIP	SKIP	IDLE	PAYLOAD B

Figure 9.27 Example of three possible data patterns from a protocol based I/O cell with the exact same payload.

In the figure there are three different cases where payload A and B are transmitted by the DUT together with command packets called “IDLE” and “SKIP.” The exact function of these control packets is irrelevant for this discussion since from the I/O test perspective we are only interested in verifying if the data content of the payload A and B packets is correct. This means that the three cases in Figure 9.27 would correspond to a functional pass as long as the data on the payload A and B packets are functionally correct. This means that simply defining a compare pattern would not be sufficient because it is not known during pattern generation how the DUT I/O protocol will generate the data packets A and B.

The lowest level of protocol aware testing might simply be to allow the test engineer to look at the data pattern and program the expected pattern without any encoding, scrambling, or error correction required by the protocol (see Appendix D). In this case the ATE system (hardware or software) has to decode and encode the test patterns. The data presentation to the test engineer happens in a way that only the real data patterns that are used without any protocol requirements are visible. At the more complex level, protocol aware ATE systems also might consider the packet structure of the DUT protocol and

only present the payload data to the test engineer. Besides packet composition and decomposition, the ATE also has to take care of correct idle-data and skip-data handling on the high-speed I/O.

There are two options to address the protocol aware test challenge. Instead of performing a functional test by comparing the output data to an expected pattern, the first option is to capture the data from the DUT and post-process it. The post-processing operation uses the protocol knowledge to identify the various packets, extracts the relevant payload data from these packets, and just compares the extracted payload data to the expected payload data. This approach provides the maximum flexibility since any kind of protocol can be implemented with the post-processing software. However, it requires that the ATE system is able to do real-time data capture at the I/O data rate. Another issue is that the time spent for data transfer and post-processing might have a significant impact on the test time (compared to a standard functional test).

The second option is to integrate the protocol capability in the ATE pin electronics hardware (i.e., the pin electronics receiver is able to understand the protocol from the DUT and only compares the payload data). Typically, protocol awareness of ATE systems is referring to such a kind of hardware implementation [17, 18]. This means that the test engineer only needs to specify the expected data for the payload and the pin electronics takes care of the protocol specifics. This approach has the advantage that no overhead or at-speed capture capabilities are required because the pin electronics understands the protocol at-speed. The challenge of this option is that it adds complexity and cost to the pin electronics. Another difficulty is that one cannot expect that a hardware implementation is able to handle any possible protocol scheme (especially the ones that are not defined when the ATE system is designed). One possible approach to mitigate this challenge is to allow some type of programming flexibility on the ATE pin electronics protocol engine either by using a FPGA or other approaches.

In this discussion it is important to note that for characterizing an I/O cell that uses a complex protocol engine it is typically easier and overall more cost efficient if precautions to simplify the test of such a device are already taken during the circuit design phase. As a minimum, it should be ensured that the device generates deterministic and predictable data. This means that, for example, devices that use encoded data offer the possibility to set the encoding circuit to a known state so that predictable codes are generated (e.g., controlled start disparity for 8B/10B code generators). For protocols that generate skip-data, DfT circuitry that allows it to generate the skip-data on a known timing grid helps to simplify the test of such devices without applying time-consuming post-processing operations or dedicated protocol aware ATE

hardware. Very often the main target of ATE-based testing of high-speed I/Os is to ensure their analog specifications such as jitter, data eye height, and so on. In such cases DfT circuitry that helps to generate the required data streams in a way that is decoupled from the device functionality is helpful for the test on an ATE. Examples for such DfT circuitry are pattern generators (e.g., PCI Express compliance pattern generator, PRBS generators), pattern analyzers, or loopback capabilities (e.g., far-end loopback paths, near-end loopback paths) that are implemented for the high-speed I/Os of a DUT.

9.7 Testing Multilevel Interfaces with Standard Digital ATE Pin Electronics

Most digital pin electronics available for today's ATE systems are designed to test and characterize NRZ type interfaces with two voltage levels (see Section 2.1). When facing the challenge of testing a multilevel interface like a PAM-4 that uses four voltage levels, it is necessary to develop an approach to test this type of interface using standard ATE pin electronics.

One possible approach is to use analog ATE instrumentation for testing this type of interface where an arbitrary waveform generator is used to provide the stimulus signal and a sampler is used to characterize/test the DUT output waveform. The other possibility is to use several standard digital channels to generate and receive a multilevel signal. Figure 9.28 shows a diagram of how the setup would look for generating a PAM-4 waveform using two standard ATE channels connected through a power combiner. The idea is to use each standard digital channel to provide two of the four levels as shown in Table 9.1. This requires that an appropriate pattern and timing setup is used. The power combiner on the setup ensures a proper signal integrity when combining the signals on the setup.

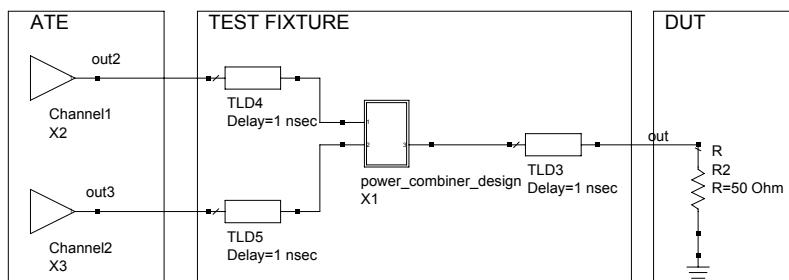


Figure 9.28 Setup for generating a multilevel PAM-4 type waveform with two standard ATE channels.

Table 9.1
Creating a PAM-4 Waveform with Two Standard ATE Channels

Drive Level	Channel 1	Channel 2
Level 1 (0V)	Low (0V)	Low (0V)
Level 2 (0.2V)	Low (0V)	High (1V)
Level 3 (0.4V)	High (2V)	Low (0V)
Level 4 (0.6V)	High (2V)	High (1V)

Figure 9.29 shows the programmed waveform at each standard ATE channel and the resulting PAM-4 waveform at the DUT. Figure 9.30 shows the PAM-4 data eye diagram at the DUT.

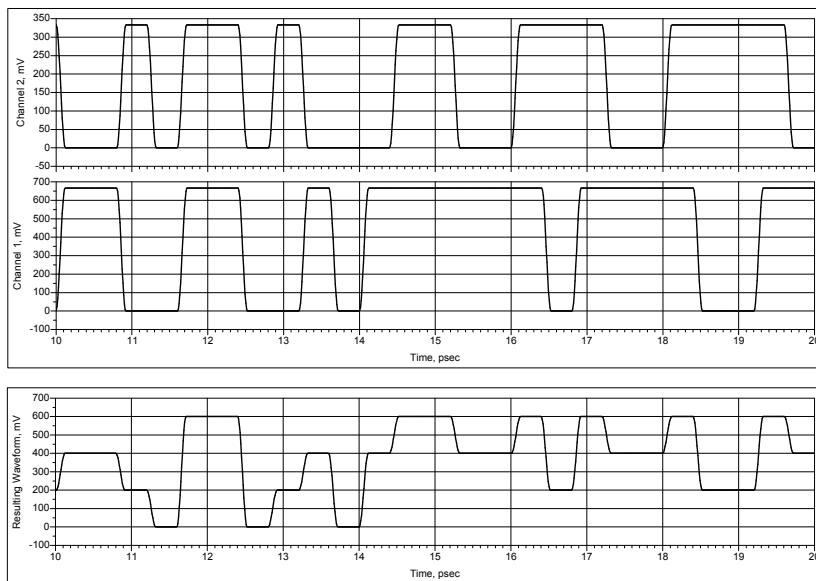


Figure 9.29 Individual channel programmed waveform (top) and combined waveform at the DUT socket (bottom).

Performing a functional test on the PAM-4 data signal stemming from the DUT with two standard ATE channels that only have a single compare threshold value (V_{th}) requires a two-pass test. In the first pass, bits corresponding to two levels are compared and on the second pass the remaining two levels are measured as shown in Table 9.2. In this case the power combiner in Figure 9.28 works as a power divider. Although the two-pass test increases the test time, it allows it to stay with a configuration consisting of two standard ATE channels for a PAM-4 functional test.

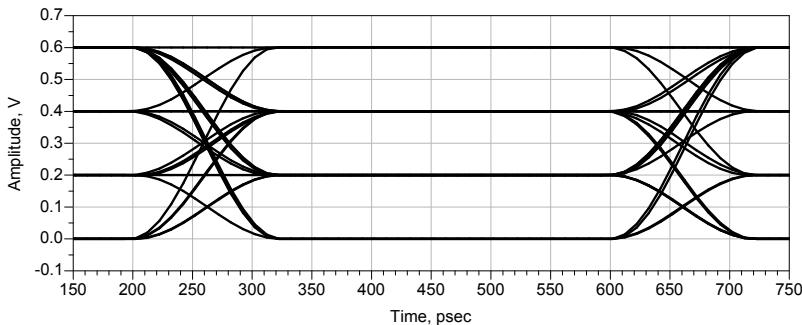


Figure 9.30 Resulting data eye diagram for a PAM-4 waveform.

Table 9.2

Functional Test of a PAM-4 Waveform Using Two Standard ATE Channels with a Single Threshold (V_{th}) (Based on the Waveform in Figure 9.30)

DUT Output Level	Channel 1	Channel 2
Level 1 (0 V) First Pass	Low ($V_{th}=0.1$ V)	-
Level 2 (0.2 V) First pass	High ($V_{th}=0.1$ V)	Low ($V_{th}=0.3$ V)
Level 3 (0.4 V) Second pass	Low ($V_{th}=0.5$ V)	High ($V_{th}=0.4$ V)
Level 4 (0.6 V) Second pass	High ($V_{th}=0.5$ V)	-

In the case of a separate threshold being available for the high and low levels in the standard ATE channels (V_h, V_l), it is possible to perform a functional test in a single pass by utilizing two ATE channels as shown in Table 9.3.

One point that is important to take into consideration when assessing the feasibility of using standard ATE channels for testing a multilevel interface is the voltage attenuation the power combiner will create. Thus, the standard ATE channels need to have a sufficient level range to compensate for that attenuation.

9.8 Signal Path Characterization and Compensation

9.8.1 Signal Path Loss Compensation: De-Embedding

Time domain de-embedding or time domain deconvolution [19] corresponds to a post-measurement step where measurement data is processed to remove the effects of the test fixture loss or the bandwidth limitations of the pin electronics receiver (assuming that those effects are known and have been measured or simulated). The idea is to assume the test fixture and the pin electronics to be a linear time invariant (LTI) system (see, for example, [20]).

Table 9.3

Functional Test of a PAM-4 Waveform with Two Standard ATE Channels with Separate Thresholds for Low and High (V_h, V_l) (e.g., a Single-Ended Interface) (Note That Only Channel 1 Requires Separate Thresholds; Channel 2 Can Use a Standard Single Threshold)

DUT Output Level	Channel 1 ($V_h=0.5\text{ V}, V_l=0.1\text{ V}$)	Channel 2 ($V_{th}=0.3\text{ V}$)
Level 1 (0 V)	Low	-
Level 2 (0.2 V)	High	Low
Level 3 (0.4 V)	Low	High
Level 4 (0.6 V)	High	-

for a proper treatment of LTI systems). One simple model for the loss a signal suffers when traveling from the DUT I/O package pad to the measurement instrument is shown in Figure 9.31 where the DUT output goes through a low-pass filter representing the test fixture (and the DUT socket) and then another one representing the bandwidth limitations of the ATE pin electronics.

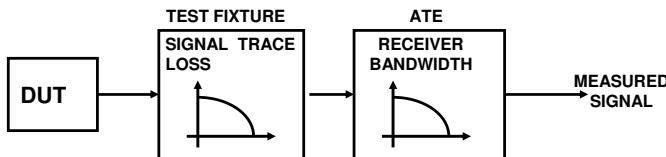


Figure 9.31 Modeling the measurement of the output of a DUT through a test fixture and a bandwidth limited ATE pin electronics.

One straightforward approach to measure the real output waveform of the DUT would be to compensate for the signal degradation by “inverting” the effect of the “low pass filters” on the measurement setup by an appropriately designed “high pass” filter as shown in Figure 9.32. This filter would be applied by means of a software algorithm on the measured data. In order to obtain this de-embedding filter we need to formalize the problem by using a black box approach as shown in Figure 9.33.

The output function (V_{OUT}) can be computed by the following equations in the time and frequency domain where * represents the convolution operator²:

$$V_{OUT}(t) = h(t) * V_{IN}(t) \quad (9.1)$$

²The convolution operator in the time domain between two functions is defined by the expression: $x(t) * y(t) = \int_{-\infty}^{+\infty} x(\delta) \cdot y(t - \delta) d\delta$.

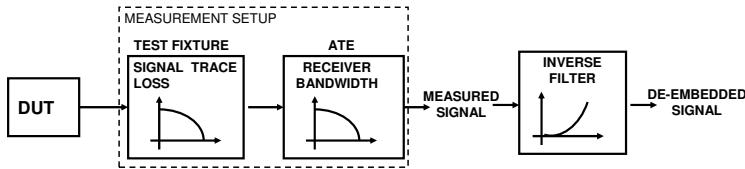


Figure 9.32 Obtaining the original DUT output through an inverse filter applied on the measured data.

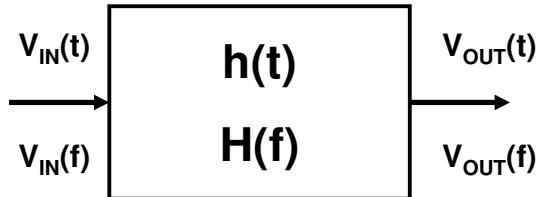


Figure 9.33 Black box model of the de-embedding problem: $h(t)$ is the time domain impulse response of the measurement system and $H(f)$ is the frequency domain transfer function.

$$V_{OUT}(f) = H(f) \cdot V_{IN}(f) \quad (9.2)$$

The de-embedding problem consists of computing the original input waveform function V_{IN} by using the measured output waveform function V_{OUT} and the system transfer function H . In the frequency domain this is represented by the following equation:

$$V_{IN}(f) = \frac{V_{OUT}(f)}{H(f)} \quad (9.3)$$

The frequency domain is used since a convolution in the time domain is simply a multiplication on the frequency domain. The time domain waveform can then be obtained from $V_{IN}(f)$ through an inverse Fourier transform.

One open question is how to obtain the system transfer function $H(f)$ since it is typically unknown. If the test fixture loss is the only item to be de-embedded, then one approach is to obtain the insertion loss of the test fixture. This can be achieved by using the test fixture measurement approaches discussed in Appendix H. The measured insertion loss ($S21$) will then correspond to $H(f)$.

To include the ATE pin electronics receiver in the model, it is typically necessary to obtain the $H(f)$ model first in the time domain since this is how a high-speed digital ATE receiver works. One way to measure the transfer function $H(f)$ is to use a test signal like a step function and measure

the resulting step function with the ATE receiver (i.e., using some of the techniques of Appendix H, a step signal is injected at the DUT socket). The $H(f)$ function can then be computed by dividing the Fourier transform of both step functions:

$$H(f) = \frac{V_{OUT}(f)}{V_{IN}(f)} = \frac{FFT\{V_{OUT}(t)\}}{FFT\{V_{IN}(t)\}} \quad (9.4)$$

The previous discussion shows the basic theory behind time domain de-embedding or deconvolution, but it is important to realize that there are several technical details regarding the Fourier transform of step functions, causality, and stability of the numerical algorithms that are outside the scope of this book. Reference [21] provides an excellent discussion of time domain deconvolution.

Figure 9.34 shows one example of a stimulus step response (generated with a bench pattern generator and measured with a high-bandwidth equivalent-time oscilloscope) and the step response measured by the ATE pin electronics which is degraded by the test fixture loss and the pin electronics bandwidth limitations. Both measured step responses are used to obtain the frequency response of the measurement setup $H(f)$ through the Fourier transform of the step waveforms as shown in Figure 9.35.

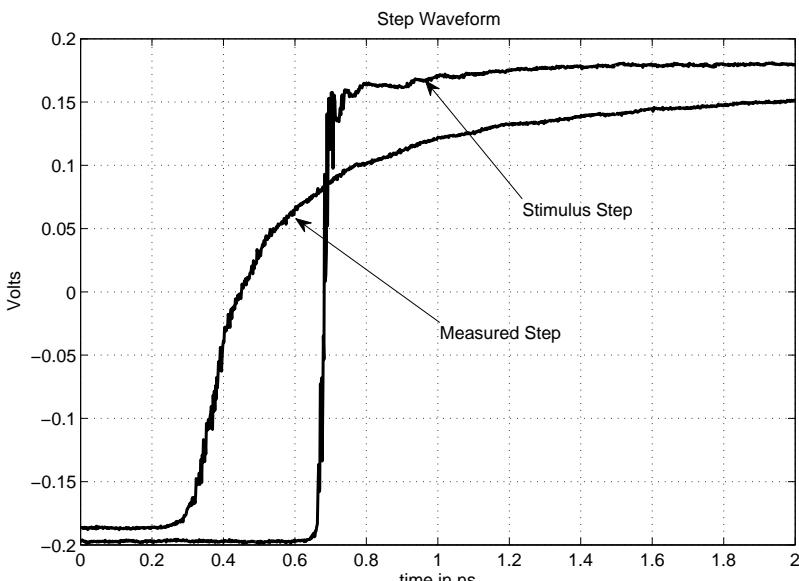


Figure 9.34 Measured step stimulus waveform and the step waveform measured by the ATE pin electronics after a lossy test fixture.

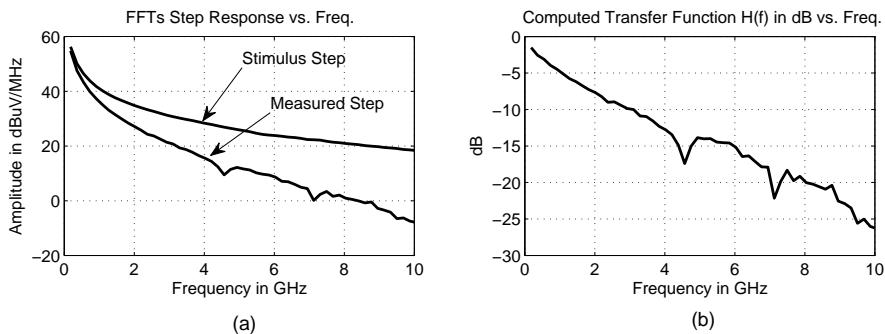


Figure 9.35 (a) Computed FFT of the input and output step responses and (b) computed system transfer function $H(f)$ based on the measured step responses.

With the knowledge of the computed transfer function $H(f)$, it is now possible to de-embed a measured waveform of the DUT. Figure 9.36 shows a comparison of a de-embedded waveform obtained through post-processing of the measured waveform with the transfer function $H(f)$ as described in (9.3).

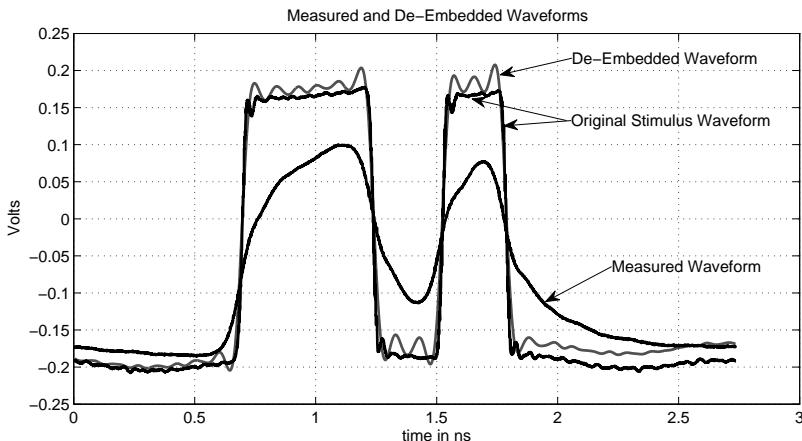


Figure 9.36 Results of applying the de-embedding procedure to a measured 5-Gbps waveform.

This example shows that time domain de-embedding can improve the measured results, but it is also important to be aware of the limitations. The first one is that perfect de-embedding in a real application is not possible due to several factors like the random noise inherent to the pin electronics that might dominate in the frequency region where we still want to de-embed a measured signal or the computational issues associated with obtaining the

transfer function using a step response. Another challenge in ATE applications is the fact that only time-equivalent (undersampled) waveforms are measured either by a sampler ATE card or the digital pin electronics. In the pin electronics case there is the additional challenge that digitizing a waveform can be very time consuming. It is also important to note that this type of post-processing technique cannot be used to improve measurements that depend on at-speed functional testing like a bathtub curve measurement. To address these drawbacks, Section 9.8.3 discusses possible hardware-based techniques in the form of equalization.

9.8.2 Characterization in the Frequency Domain

The previous subsection discussion can also be used as an approach to measure the performance of an ATE receiver channel in the frequency domain with or without the test fixture loss included or even for characterizing the test fixture performance separately from the ATE driver/receiver channel performance.

Figure 9.37 shows an example of measuring the ATE receiver performance in the frequency domain with a reference test fixture using an external pattern generator that provides a step waveform that is measured by the ATE receiver.

The frequency response of the ATE receiver plus the reference test fixture can be computed from the Fourier transform of the measured step responses as described in (9.4) and shown in Figure 9.37.

If no reference test fixture is to be included and only the receiver performance is to be measured, then the pattern generator should be connected directly to the ATE receiver pogo pin.

If the objective is to measure the frequency response of the test fixture, this can be accomplished either for the drive or the receive channels. For the ATE driver channels, one needs to measure the step response at the ATE driver without the test fixture and at the DUT socket with the test fixture docked on the ATE system. The measurement of the driver step response can be done with an external instrument like an equivalent-time oscilloscope. The two measured step responses will then provide the frequency response of the test fixture through (9.4). For the receiver side the process is similar. One measures the step response with the ATE receiver when a pattern generator is connected directly to the ATE receiver pogo pin and when it is connected to the DUT socket on the test fixture. From these two step responses the frequency response of the test fixture can be computed.

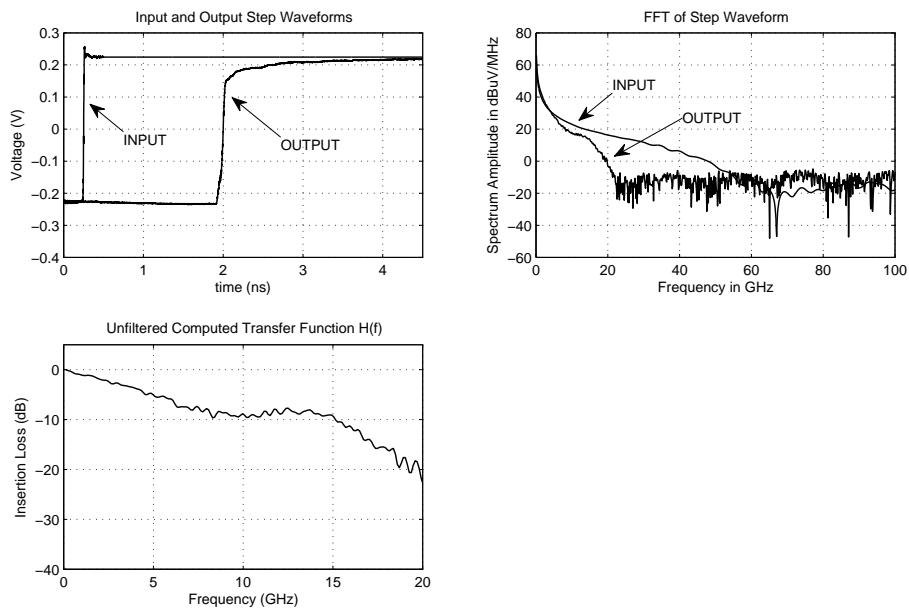
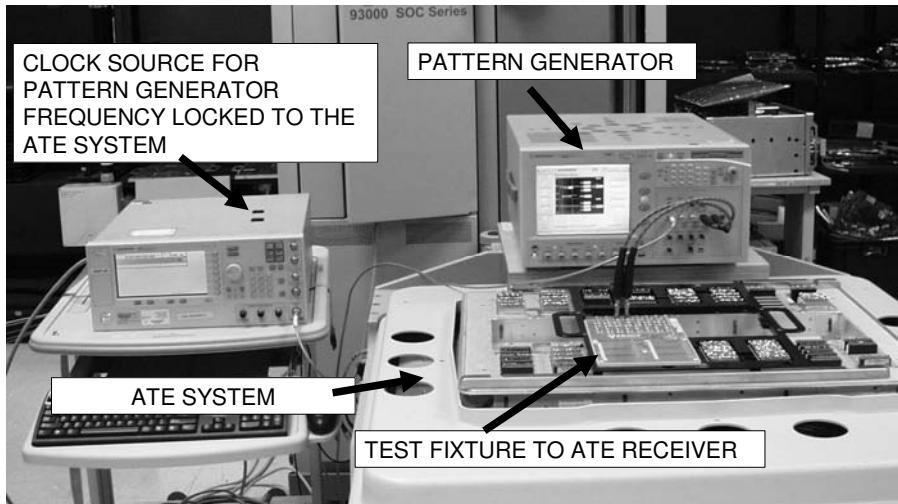


Figure 9.37 Example of the measurement setup for characterizing an ATE receiver in the frequency domain using a step waveform (top) and the measured step response and computed frequency transfer function of the ATE receiver (bottom) (courtesy of Verigty).

9.8.3 Signal Path Loss Compensation: Equalization

The objective of equalization is to substitute the inverse filter function in Figure 9.32 with a hardware component so that the compensation is done before the signal is sampled by the receiver. This means that in contrast to the de-embedding technique presented in Section 9.8.1 it will apply also to at-speed tests like a BER bathtub curve measurement. The other advantage is that equalization can be applied to both the driver and receiver on the ATE pin electronics.

The term “equalization” is used in several different contexts and it is important to first describe how it will be used in the context of this section. By equalization we mean a system that compensates for the degradation of the waveform due to the test fixture loss or the measurement instrumentation. That is, in the case of a waveform from a DUT driver, equalization should ensure that the waveform at the ATE receiver input is exactly the same as at the DUT package pad (independent of whether the waveform is good or bad).

On the stimulus signal to the DUT (i.e., the ATE driver), the term “pre-emphasis” is commonly used, although we will also use the term “equalization” when referring to the ability of the ATE driver to compensate for the loss on the test fixture.

In a modern I/O cell receiver for high-speed digital applications, the objective is to correctly identify the bit stream logic values with the exact shape of the electrical waveform being of lower importance. In this context techniques like decision feedback equalization (DFE) [22] (see also Section 2.6.4) have shown significant advantages since they target the key objective which is to figure out the logic values of the received bits and not on the waveform shape at the link partner driver. But on a test and measurement application like ATE, the electrical waveform is the primary objective together with the bit sequence. Because of this, techniques like DFE are not discussed in the context of equalization on ATE although it is important to point out that testing the DFE circuitry on a DUT is an important topic for ATE (see Section 5.7.4). However, this is not related to the test fixture loss compensation.

The next subsections will present the two main approaches for equalization on ATE: passive and active equalization [23].

9.8.3.1 Passive Equalization

Passive equalization is the longest established approach to compensate for the loss of a test fixture using a hardware approach [24]. The idea is to compensate the test fixture loss, which can be modeled as a low pass filter, by a matching filter similar to a high pass filter that flattens the frequency response. The idea is shown in Figure 9.38 [25, 26].

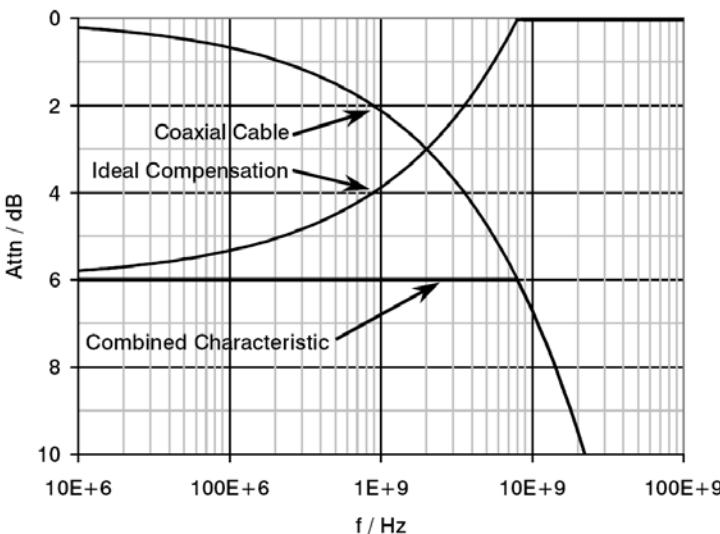


Figure 9.38 Compensating for a coaxial cable signal path loss through a passive equalization filter.

From the figure it is also possible to notice that the combined response does imply a certain constant loss through the entire frequency spectrum of interest. This is the price to be paid for using a passive equalizer. Since the loss is constant over the frequency range of interest, it easily can be calibrated. Figure 9.39 shows one example of a single-ended equalizer topology and the simulation results in the frequency domain of applying a specific implementation of this equalization filter to a lossy signal path. Note that although there is a 8-dB DC loss, the frequency dependent behavior of the signal path is reduced significantly. The DC loss will reduce the data eye height amplitude and can be calibrated. The important fact is that in the time domain, equalization will improve level and timing jitter significantly as will be demonstrated later using a real example.

The challenge with implementing a passive equalizer for high-speed digital applications is to make sure that the manufactured equalizer provides the needed frequency response. Also, for high pin count applications this might require a small form factor [27]. Figure 9.40 shows some examples of passive equalizers for coaxial cables and PCB test fixtures.

Figure 9.41 and Figure 9.42 show a real example of using a passive equalizer to compensate for the test fixture loss. In Figure 9.41 the frequency response of the test fixture without any compensation is shown together with the frequency response of the passive equalization filter that is used and the combined response of the test fixture loss with the passive equalization filter

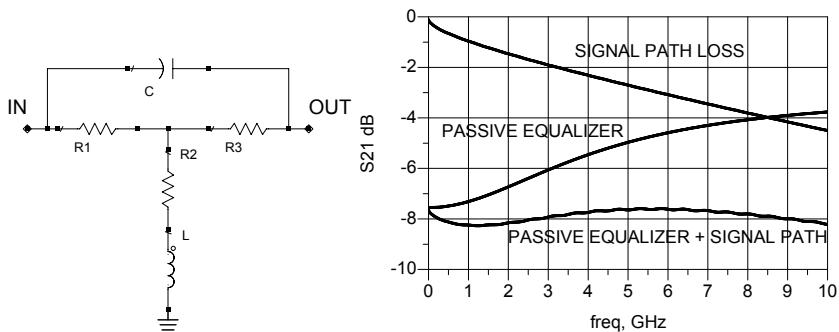


Figure 9.39 Example of a simple passive equalizer circuit and its compensation effect on the signal trace loss.

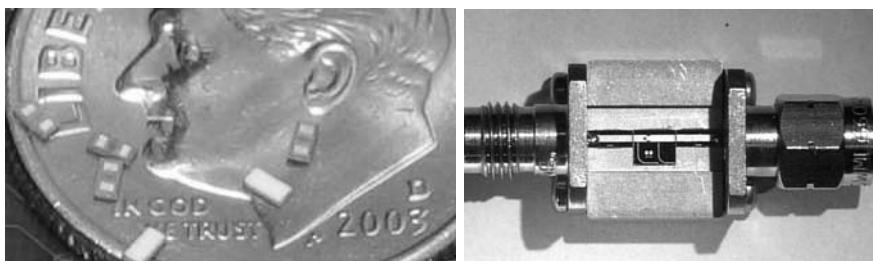


Figure 9.40 Examples of passive equalizers. Left: 0603 type surface mounted passive equalizers (courtesy of Thin Film Technology Corporation [27]); right: passive equalization filter implemented in thin-film technology with wire bonding in a coaxial package (courtesy of Verigy [28]).

(the DC loss is compensated out on the figure). The figure shows that the bandwidth of the signal path between the DUT and the ATE pin electronics is improved from 2.8 GHz to 8.3 GHz, an improvement of 5.5 GHz in terms of bandwidth. Figure 9.42 shows this improvement in the time domain using the data eye diagram. Although the signal amplitude has been reduced due to the passive equalization, the jitter and the data eye opening have been improved.

It is also possible to add a passive equalizer directly into the ATE pin electronics. This has the advantage that it allows the ATE manufacturer to already include all the required calibration factors into the system's standard calibration. The remaining question is if a single passive equalizer could handle all the different possibilities of loss for an ATE test fixture. This depends on the design of the passive equalizer and on the strategy for designing the DUT test fixtures. Figure 9.43 shows both options with an example of an ATE pin electronics channel that contains an integrated passive equalizer and a test fixture with an integrated passive equalizer [27].

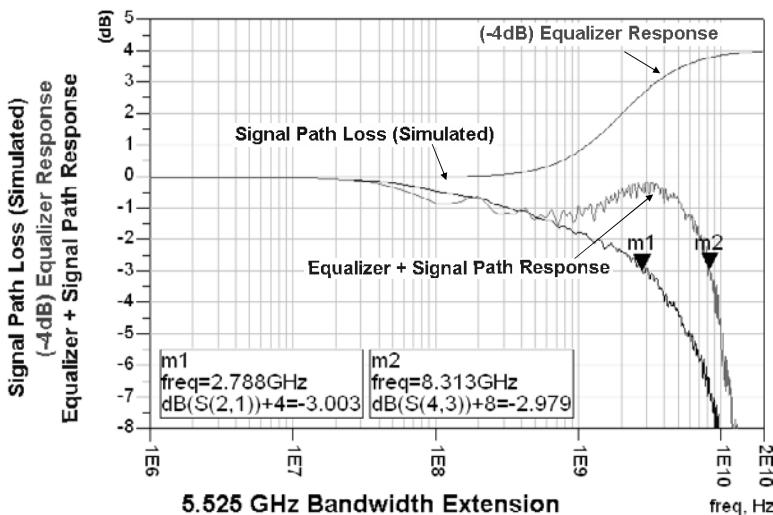


Figure 9.41 Example in the frequency domain of the bandwidth extension through the use of passive equalization (the DC loss of the passive equalizer is calibrated out in the graph).

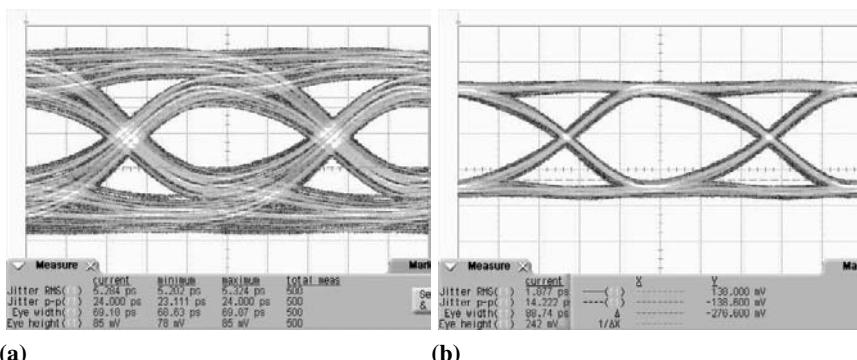


Figure 9.42 Data eye diagram comparison after a lossy signal path (a) without equalization and (b) with equalization.

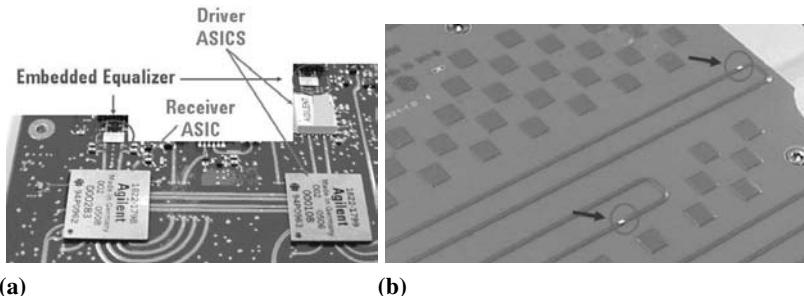


Figure 9.43 Picture of a passive equalizer (a) integrated on an ATE pin electronics card and (b) integrated on the test fixture. Reprinted with permission from [27].

9.8.3.2 Active Equalization³

Another approach to implement equalization on the ATE pin electronics is to use an active circuit. This circuit can be implemented as part of the driver/receiver ASIC on the ATE pin electronics. An active circuit implementation, unlike with passive equalization, allows it to have different types of equalization settings (e.g., for different test fixtures losses).

The important point when discussing an active compensation approach is that it must be a continuous time compensation [22, 26]; that is, the compensation must literally shape the waveform at the DUT and not simply create an open data eye. Figure 9.44 shows a simple implementation of an active equalizer where a copy of the signal is sent through a high pass filter and then added to the original signal resulting in a frequency response that amplifies the higher frequencies and in this way compensates for the frequency dependent loss of a signal path. Note that unlike passive equalization, there is no DC loss with an active equalizer, because the compensated AC loss gets added before the signal is launched into the transmission line.

The possibility of integrating the active equalizer in silicon also allows the option to increase the complexity of the active equalizer with multiple degrees of freedom. Figure 9.45 shows a more realistic approach for an active equalization circuit that would allow programmable equalization on the pin electronics driver. There are two high pass type filters with different time constants that tap into the input signal and then are added to the input signal to generate the output waveform [29]. Each filter has a programmable gain K1 and K2. Figure 9.46 shows the time domain step response of each stage of

³In collaboration with Bernhard Roth.

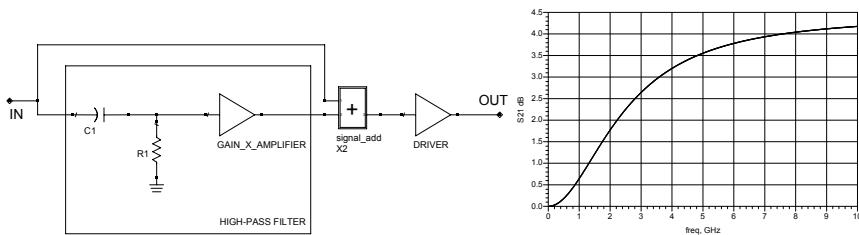


Figure 9.44 Example of a simple active equalizer topology and its frequency response.

the equalizer circuit for a certain value of the K1 and K2 gains and the final combined step response. This approach can be extended by adding additional filter stages.

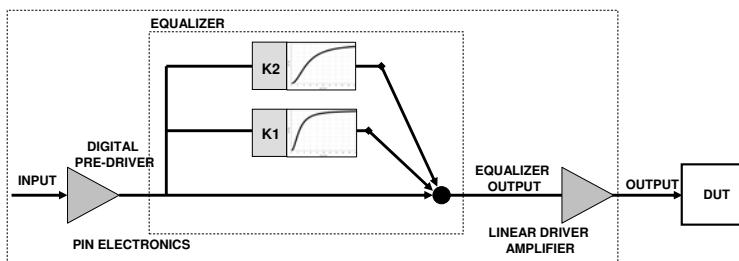


Figure 9.45 High-level architecture of an equalization approach for integration on the pin electronics driver with variable control.

Figure 9.47 shows the data eye diagram at the various stages of an ATE system with active equalization: at the ATE pin electronics before equalization, after the equalization, and at the DUT I/O input. It also presents the data eye at the DUT I/O without equalization. The figure shows how the waveform at the DUT is approximated to the original waveform at the ATE driver with the active equalization in comparison to the waveform at the DUT without equalization.

On the pin electronics receiver side the procedure is exactly the same, although in this case the input signal to the equalizer is the DUT output as shown in Figure 9.48. It is important to note that equalization on the ATE pin electronics is not restricted to either passive or active approaches. There are also some advantages in implementing a hybrid approach combining both passive and active equalization.

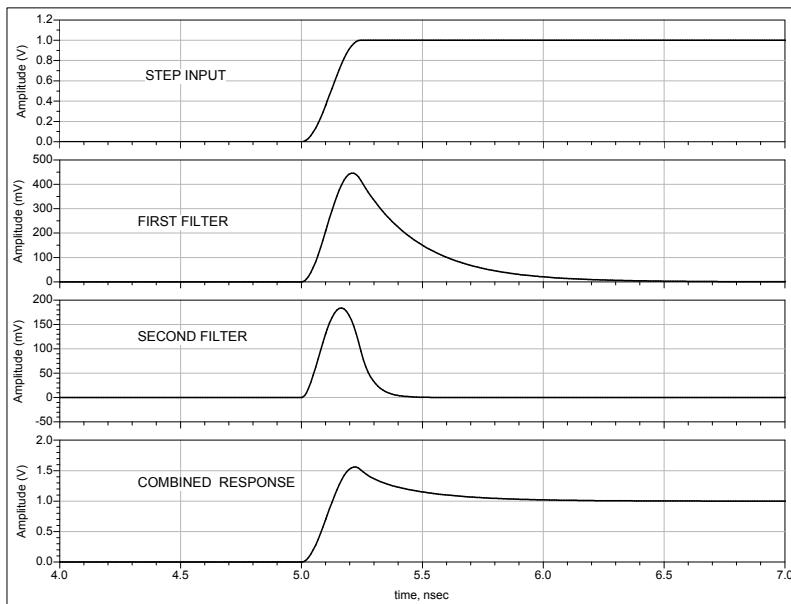


Figure 9.46 Comparison of the initial step response before the equalizer with the individual step responses of each filter on the equalizer and the combined step response of the equalizer.

9.8.3.3 Pitfalls of Overequalization

Equalization is critical for testing high-speed digital applications where the loss of the test fixture can dominate the ATE system performance on the one hand. On the other hand, equalization can have a negative impact on the ability to properly test a DUT if it is not applied correctly. The reason is that equalization should be used to compensate for any loss from the test fixture or pin electronics but not to improve the performance of the DUT (i.e., it should not compensate for any possible deficiencies on the DUT, for example, input capacitance).

Figure 9.49 shows how overequalization can mask a defect in a DUT I/O with a passing functional test when in fact the DUT should fail the functional test. In the figure the DUT driver waveform which consists of a single pulse representing a differential logic high is compared against a fixed threshold to check if the pulse corresponds to a logic high or low. The waveform at the DUT I/O driver is unable to reach this threshold which would represent a bad driver and should result in a failing functional test. The waveform is further degraded by the test fixture during its transmission to the ATE pin electronics. In this case the equalization in the pin electronics was not properly set to

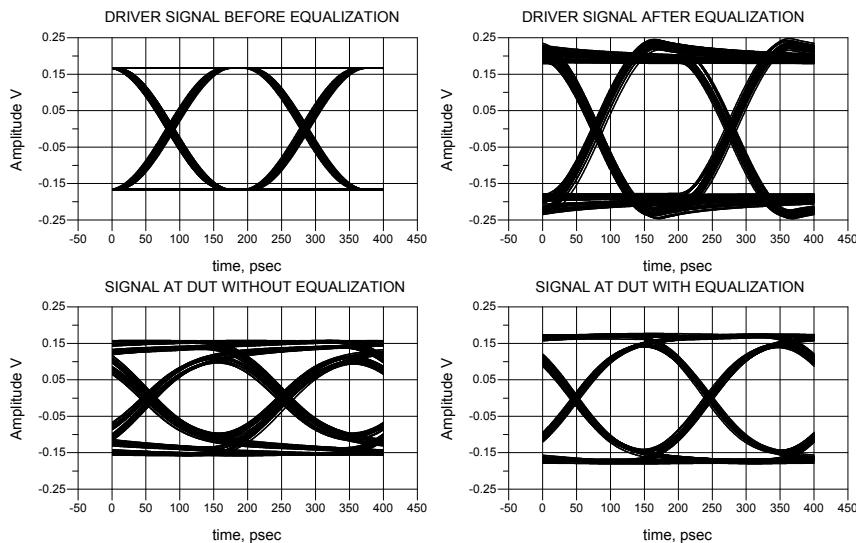


Figure 9.47 Demonstration of the loss compensation capability of the active equalization approach on improving the waveform at the DUT. Data eye diagram before equalization, after equalization, and at the DUT. The data eye diagram at the DUT without equalization is also included for comparison.

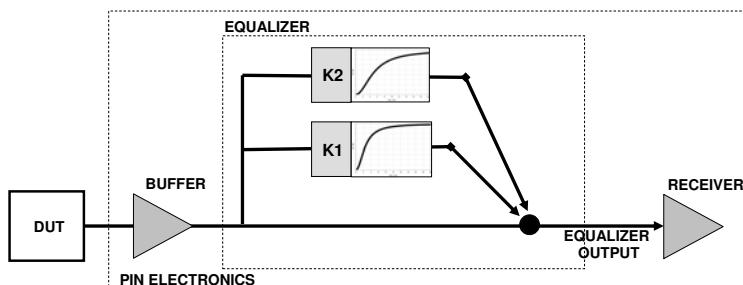


Figure 9.48 High-level architecture of an equalization approach for integration on the pin electronics receiver with variable control.

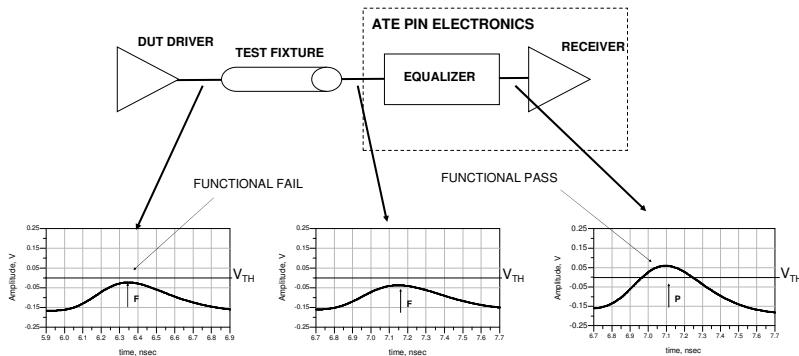


Figure 9.49 Example of how overequalization can mask the problems of a DUT I/O cell.

compensate for the test-fixture loss only but was set to overcompensate the test fixture loss and in this way improve the waveform from the DUT (i.e., compensating for the possible deficiency in the DUT driver). The waveform after the equalizer is now able to achieve the threshold value and would be interpreted by the ATE receiver as a functional pass. The same case can be constructed for the DUT receiver.

The overequalization effect demonstrated in Figure 9.49 can also be shown in the frequency domain. Figure 9.50 shows the insertion loss of the test fixture plus the ATE pin electronics equalizer. We would expect that the equalizer improves the bandwidth of the test fixture but what we observe is that it overcompensates the test fixture loss in the frequency range below 5 GHz. This overcompensation ends up masking the real performance of the DUT.

The danger of overequalization, especially in the case where there is an associated control “knob,” is that the test engineer in the search for the maximum possible yield turns this “knob” to its maximum value. This strategy can in some situations result in an increased yield but at the expense of a reduced fault coverage (i.e., more field returns).

9.8.3.4 Using Active Equalization for DDJ Injection

The ability to have a programmable equalization on the ATE pin electronics driver using an approach like the one presented in Figure 9.45 opens the possibility of using the equalization for DDJ injection. This can be achieved by programming each filter stage of the equalizer to have a low pass behavior instead of the expected high pass behavior from a loss compensation equalizer

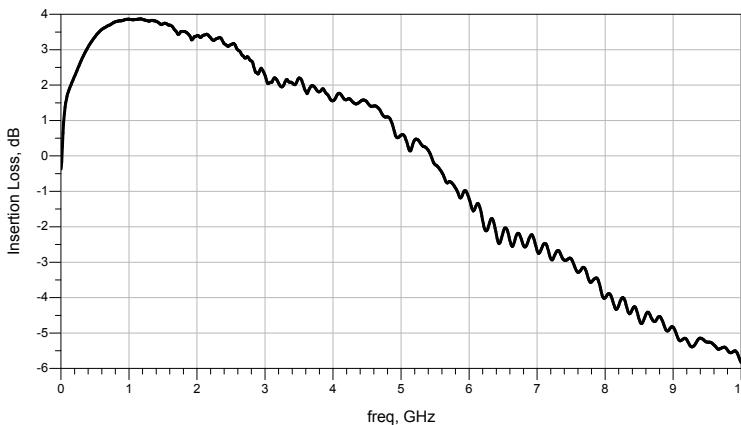


Figure 9.50 Test fixture plus ATE pin electronics insertion loss in the case of overequalization.

as shown in Figure 9.51. This can also be observed on the simulated data eye diagram in Figure 9.52 [23].

9.9 ATE DC Level Adjustments

ATE drivers and receivers designed for testing high-speed I/O circuits typically provide impedances that match the $50\ \Omega$ single-ended or $100\ \Omega$ differential impedances that are predominantly used for the high-speed I/Os of a DUT. In order to achieve the high data rates, proper terminations in the ATE pin electronics and the DUT I/Os are mandatory [30–32]. For a DUT that is to be deployed in a $50\ \Omega$ system environment, the programmed ATE level values can be directly applied from the respective datasheets with potential corrections due to level loss on the signal path as described in Section 9.3.2. Some high-speed I/O standards, though, do not use a $50\ \Omega$ environment. On the other hand, ATE high performance pin electronics cannot be implemented in a cost effective way to support all possible impedance environments. If a device that is designed for a system environment impedance different than the $50\ \Omega/100\ \Omega$ mainstream of the ATE pin electronics needs to be tested according to its specifications, a DC level adjustment of the drive and compare levels used by the ATE is required. The reader should be aware that in this section we only consider the DC effects of such an impedance mismatch. If the impedance difference between ATE pin electronics and DUT I/Os becomes too large, also AC effects as described in Section 8.2.1 have to be taken into consideration and have to be corrected by means of loadboard design.

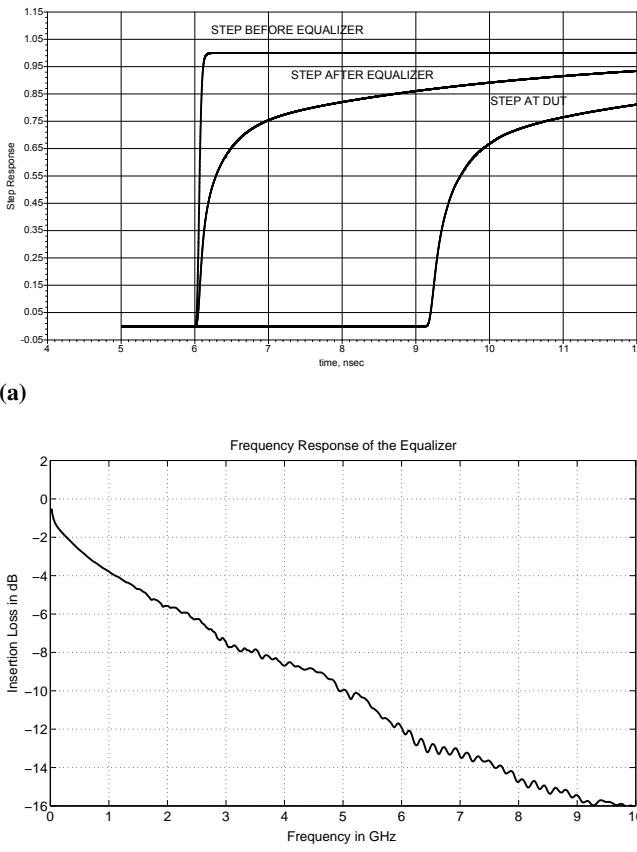


Figure 9.51 DDJ injection through an active equalizer on the ATE pin electronics. (a) Step response before and after the equalizer and at the DUT, and (b) frequency response of the equalizer.

The guiding principles for the correction of the DC levels for non-matching source-impedance and receiver-termination pairs is that for inputs, the DUT should be stimulated at its pin with the same levels as in an impedance matched environment. For output pins, the levels that are measured by the ATE on the DUT pins are based on different load currents I_{OL} and I_{OH} than is the case in a system environment. Thus, the voltages measured at the pins of the DUT need to be corrected to the corresponding values that would be valid in the system environment before comparing them against the specified levels. If the termination scheme in the system environment is symmetrical (i.e., uses identical source and load impedances), the compare threshold level used by the ATE receiver does not need any adjustment

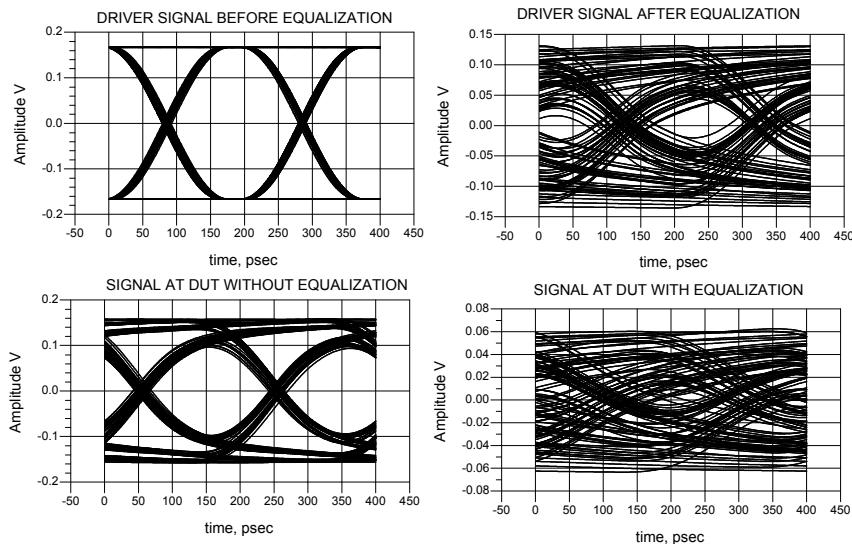


Figure 9.52 DDJ injection through an active equalizer on the ATE pin electronics. Data eye diagram before the equalizer, after the equalizer, and at the DUT. The data eye diagram at the DUT without equalization is also included for comparison.

because the effects on the measured high and low levels also are symmetrical and thus do not have an effect on the threshold that usually is at the center of high and low levels. If the termination scheme, however, is not symmetrical (e.g., as for POD15; see Section 3.4.5), the threshold value that is used to distinguish between logical high and low levels also needs to be adjusted.

If one wants to apply the load currents as seen in a system environment also in the ATE environment, impedance adjustments by means of test fixture design need to be applied or appropriate adjustments of the termination voltages used by the ATE receivers need to be done. The difficulty with the latter of these two is that this would result in two different termination voltages that would need to be used for received high and low levels.

9.9.1 Correction of Force Levels for DUT Input Pins

The general configuration for an input pin of the DUT is shown in Figure 9.53. In this figure, V_{ATE} denotes the voltage that is programmed as force voltage of the ATE driver. V_{DTerm} is the termination voltage used by the DUT, R_{DTerm} is the termination impedance of the DUT receiver, and V_I denotes the resulting voltage at the input pin of the DUT.

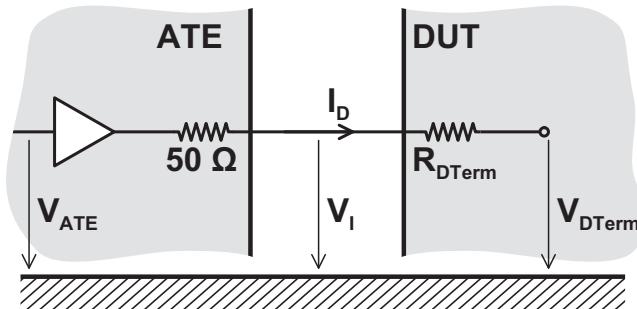


Figure 9.53 General configuration of an ATE to DUT connection for a DUT input pin.

The goal of our calculation is to determine V_{ATE} for a given V_{DTerm} and R_{DTerm} . In order to get there we first calculate the drive current that the ATE driver has to provide according to (9.5) and the resulting voltage at the DUT input pin according to (9.10).

$$I_D = \frac{V_{ATE} - V_{DTerm}}{50\Omega + R_{DTerm}} \quad (9.5)$$

$$V_I = V_{DTerm} + I_D R_{DTerm} \quad (9.6)$$

Substituting I_D in (9.6) with (9.5) yields

$$V_I = V_{DTerm} + \frac{V_{ATE} - V_{DTerm}}{50\Omega + R_{DTerm}} R_{DTerm} \quad (9.7)$$

With this, V_{ATE} can be calculated as

$$V_{ATE} = \left(1 + \frac{50\Omega}{R_{DTerm}} \right) V_I - \frac{50\Omega}{R_{DTerm}} V_{DTerm} \quad (9.8)$$

9.9.2 Correction of Levels for DUT Output Pins

For the correction of the measured DUT output voltage, the source voltage V_{DSrc} used by the DUT to drive the source impedance R_{DSrc} on its outputs to achieve a certain device output voltage V_{OSYS} in the system environment is calculated in a first step. The basis for this calculation is the system configuration with the termination impedance R_{DTerm} and termination voltage V_{DTerm} of the receiving device as shown in Figure 9.54.

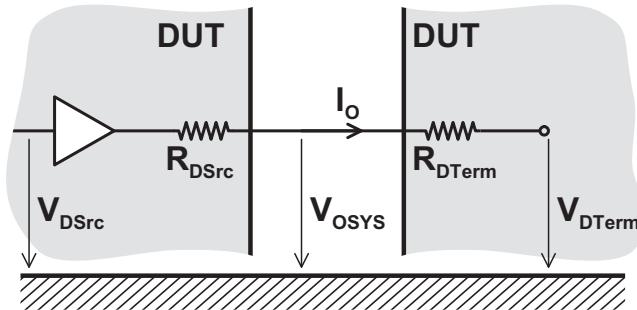


Figure 9.54 General configuration of a DUT system connection for a DUT output pin.

From the figure we get

$$I_O = \frac{V_{DSrc} - V_{DTerm}}{R_{DSrc} + R_{DTerm}} \quad (9.9)$$

$$V_{DSrc} = V_{DTerm} + I_O(R_{DSrc} + R_{DTerm}) \quad (9.10)$$

We also can derive I_O dependent on the voltage V_{OSYS} at the device pin as

$$I_O = \frac{V_{DSrc} - V_{OSYS}}{R_{DSrc}} \quad (9.11)$$

Equation (9.11) in (9.10) yields

$$V_{DSrc} = V_{DTerm} + \frac{V_{DSrc} - V_{OSYS}}{R_{DSrc}} (R_{DSrc} + R_{DTerm}) \quad (9.12)$$

By solving this equation for V_{DSrc} we get

$$V_{DSrc} = \left(1 + \frac{R_{DSrc}}{R_{DTerm}}\right) V_{OSYS} - \frac{R_{DSrc}}{R_{DTerm}} V_{DTerm} \quad (9.13)$$

For the ATE environment as shown in Figure 9.55 we can derive the DUT source voltage required to obtain a certain device output voltage V_{OATE} in the ATE environment in a similar manner.

From Figure 9.55 together with (9.13), we get

$$V_{DSrc} = \left(1 + \frac{R_{DSrc}}{50\Omega}\right) V_{OATE} - \frac{R_{DSrc}}{50\Omega} V_{ATerm} \quad (9.14)$$

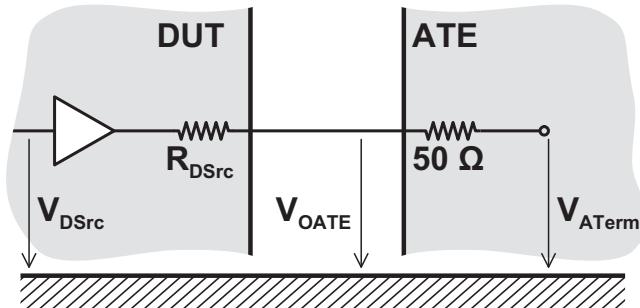


Figure 9.55 General configuration of an ATE to DUT connection for a DUT output pin.

The DUT source voltage V_{DSrc} stays the same, whether we use the device in the impedance matched system environment or in a potentially non-impedance matched ATE environment. Thus, we can link the DUT output voltage V_{OSYS} in a system environment to the DUT output voltage V_{OATE} in a nonimpedance matched ATE environment by equalizing (9.13) with (9.14). This yields

$$\left(1 + \frac{R_{DSrc}}{R_{DTerm}}\right)V_{OSYS} - \frac{R_{DSrc}}{R_{DTerm}}V_{DTerm} = \left(1 + \frac{R_{DSrc}}{50\Omega}\right)V_{OATE} - \frac{R_{DSrc}}{50\Omega}V_{ATerm} \quad (9.15)$$

Solving this equation for V_{OSYS} results in (9.16) that is used to calculate the device output voltage that would be present in a system environment from a measured output voltage value in a nonimpedance matched ATE environment.

$$V_{OSYS} = \frac{R_{DTerm}}{R_{DTerm} + R_{DSrc}} \left(\left(1 + \frac{R_{DSrc}}{50\Omega}\right)V_{OATE} - \frac{R_{DSrc}}{50\Omega}V_{ATerm} + \frac{R_{DSrc}}{R_{DTerm}}V_{DTerm} \right) \quad (9.16)$$

For nonsymmetrical termination schemes, this formula also can be used to calculate the compare threshold that has to be used for an ATE receiver to match the threshold given in the system specification. Equation (9.16) has to

be solved for V_{OATE} in this case. With the resulting (9.17), the ATE threshold for a given system threshold V_{OSYS} can be determined.

$$V_{OATE} = \frac{50\Omega}{50\Omega + R_{DSrc}} \left(\left(1 + \frac{R_{DSrc}}{R_{DTerm}} \right) V_{OSYS} - \frac{R_{DSrc}}{R_{DTerm}} V_{DTerm} + \frac{R_{DSrc}}{50\Omega} V_{ATerm} \right) \quad (9.17)$$

Example POD15 signaling (see Section 3.4.4.2) as used by GDDR5 is an example for a nonsymmetrical termination scheme that does not use 50Ω terminations. Instead it uses a 40Ω drive source impedance R_{DSrc} and a 60Ω termination impedance R_{DTerm} on its receivers (for details see Section 3.4.5). The termination voltage V_{DTerm} that is used by POD15 is the supply voltage $VDDQ$. With this boundary condition and (9.8), the resulting correction equation to calculate the ATE drive levels to be programmed into open for the GDDR5 drive levels is

$$V_{ATE} = \frac{11}{6} V_I - \frac{5}{6} VDDQ \quad (9.18)$$

For the ATE comparators, we can derive the correction rule from (9.16) as

$$V_{OSYS} = \frac{27}{25} V_{OATE} - \frac{2}{25} VDDQ \quad (9.19)$$

Since POD15 is nonsymmetrical, threshold voltages programmed on the ATE also need to be corrected due to the nonmatching impedance environment. For POD15, the standard threshold voltage is $V_{OSYS} = 0.7 VDDQ$. With this and (9.17), we can calculate the corrected threshold voltage to be programmed for the ATE receivers to $V_{OATE} = (13/18) VDDQ$ if a termination voltage $V_{ATerm} = VDDQ$ is used by the ATE receiver.

References

- [1] N. Kularatna, *Digital and Analog Instrumentation: Testing and Measurements*. The Institution of Electrical Engineers, 2003.
- [2] SEMI G80-0200, *Test Method for the Analysis of Overall Digital Timing Accuracy for Automated Test Equipment*, 2000.
- [3] D. C. Keezer, “A Multiplexing Test System Channels for Data Rates Above 1 Gb/s,” *IEEE International Test Conference*, 1990.

- [4] D. C. Keezer, D. Minier, and M.-C. Caron, "Multiplexing ATE Channels for Production Testing at 2.5 Gbps," *IEEE Design and Test of Computers*, 2004.
- [5] D. Keezer, D. Minier, P. Ducharme, and A. Majid, "An Electronic Module for 12.8 Gbps Multiplexing and Loopback Test," *IEEE International Test Conference*, 2008.
- [6] J. Moreira and B. Roth, "Characterization and Focus Calibration of ATE Systems for High-Speed Digital Applications," *IEC DesignCon*, 2009.
- [7] M. Shimanouchi, "Timing Accuracy Enhancement by a New Calibration Scheme for Multi-Gbps ATE," *IEEE International Test Conference*, 2004.
- [8] G. Haensel, "Beating V93000 Timing Specification with Focus Calibration," *Verigy VOICE Users Conference*, 2010.
- [9] J. Moreira, C. McCowan, Z.-Y. Wen, Y. Zhan, Y. He, and W.-M. Zhang, "Calibration of the ATE Driver Data Eye Height for Receiver Sensitivity Measurements," *Verigy VOICE Users Conference*, 2010.
- [10] B. Roth, "Eye-Profile: Generation and Display," *Verigy Technical Note*, 2009.
- [11] J. Ren and K. S. Oh, "Multiple Edge Responses for Fast and Accurate System Simulations," *IEEE Transactions on Advanced Packaging*, vol. 31, Nov. 2008.
- [12] D. C. Keezer and Q. Zhou, "Alternative Interface Methods for Testing High-Speed Bidirectional Signals," *IEEE International Test Conference*, 1998.
- [13] J. Moreira, H. Barnes, and H. Werkmann, "DUT Loadboard Layout for Multi-Gigabit Bi-Directional Interfaces," *Verigy Users Conference*, 2007.
- [14] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [15] NXP Semiconductors, *I2C-Bus Specification and User Manual Rev. 3.0*, June 2007. UM10204.
- [16] J. Rivoir, "Protocol-Aware ATE Enables Cooperative Test Between DUT and ATE for Improved TTM and Test Quality," *IEEE International Test Conference*, 2007.
- [17] E. Larson, "Protocol Aware ATE," *Beijing Advanced Semiconductor Technology Symposium*, 2008.
- [18] A. C. Evans, "The New ATE: Protocol Aware," *IEEE International Test Conference*, 2007.
- [19] S. M. Riad, "The Deconvolution Problem: An Overview," *Proceedings of the IEEE*, vol. 74, Jan. 1986.
- [20] W. J. Rugh, *Linear System Theory*. Prentice-Hall, 1996.
- [21] J. R. Andrews, "Deconvolution of System Impulse Responses and Time Domain Waveforms," *Picosecond Labs Application Note AN-18*, 2004.

- [22] B. Casper, P. Pupalaikis, and J. Zerbe, “Serial Data Equalization,” *IEC DesignCon*, 2007.
- [23] B. Roth and J. Moreira, “Equalization in the HSM6800 Pin-Electronics,” *Verigy VOICE Users Conference*, 2010.
- [24] H. W. Bode, “Attenuation Equalizer,” *U.S. Patent 2,096,027,027*, 1936.
- [25] W. Humann, “Compensation of Transmission Line Loss for Gbit/s Test on ATEs,” *IEEE International Test Conference*, 2002.
- [26] S. H. Hall and H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*. John Wiley and Sons, 2009.
- [27] J. Moreira, M. Howieson, M. Petersen, M. Broman, and J. Kenton, “Passive Equalization of Test Fixtures for High-Speed Digital Measurements with Automated Test Equipment,” *IEEE International Design and Test Workshop*, 2006.
- [28] J. Moreira, G. Haensel, and F. Koban, “Addressing the Challenges of Implementing an At-Speed Production Test-Cell for 10Gb/s Wafer Probing,” *DesignCon*, 2005.
- [29] H. Higashi, “5-6.4Gbps 12 Channel Transceiver with Pre-Emphasis and Equalizer,” *Symposium on VLSI Circuits*, 2004.
- [30] G. Esch and T. Chen, “Design of CMOS IO Drivers with Less Sensitivity to Process, Voltage, and Temperature Variations,” *Electronic Design, Test and Applications, IEEE International Workshop on*, 2004.
- [31] G. Esch and R. B. Manley, “Theory and Design of CMOS HSTL I/O Pads,” *The Hewlett-Packard Journal*, Aug. 1998.
- [32] G. Esch and T. Chen, “Near-Linear CMOS I/O Driver with Less Sensitivity to Process, Voltage, and Temperature Variations,” *IEEE Trans. VLSI Syst.*, vol. 12, no. 11, 2004.

A

Introduction to the Gaussian Distribution and Analytical Computation of the BER

This appendix provides some basic concepts on the Gaussian (or normal) probability distribution that is important for the analysis of high-speed digital signals. It also shows how to apply those concepts to the analytical computation of the bit error rate (BER) of a system. Most of the variables we measure on a typical ATE application exhibit some type of random behavior that normally is described as noise on the measurement. But in other cases like a jitter measurement, the objective is to measure and characterize this random (noisy) behavior. When we have a system that exhibits this type of behavior, we can analyze it using the mathematical arsenal that is available in the form of probability theory, stochastic processes, and so on. For the interested reader [1, 2] provide a good starting point. For the topics addressed in this book, all random variables are modeled as Gaussian distributions. The choice of the Gaussian distribution is not by chance. There is a mathematical principle that brings some insight into this choice; this principle is known as the Central Limit Theorem.

Central Limit Theorem

Given N independent random variables X_i , we can add them in the following way: $X = X_1 + \dots + X_N$

This is a random variable with mean $\mu = \mu_1 + \dots + \mu_N$ and variance $\sigma^2 = \sigma_1^2 + \dots + \sigma_N^2$. The Central Limit Theorem states that under certain general conditions, the distribution $F(x)$ for X approaches a Gaussian distribution with the same mean and variance.

The Central Limit Theorem tells us that if we have a system where the result of a given measurement is due to the contributions of a large number of independent events that are controlled by any kind of random distribution, then the distribution of the sum of these events will be a Gaussian.

As an example, take a crystal used for the reference clock of a system. The crystal is composed of several atoms in a crystalline structure where each atom will have the same response (with an unknown distribution) to external factors like temperature. When looking at the measurement value for this crystal, we are interested in the jitter from the crystal over a certain period of time; this jitter value will be the result of the behavior of all the atoms in the crystal. We can then assume that the resulting measured distribution will follow a Gaussian distribution because of the Central Limit Theorem.

The mathematics of a Gaussian distribution imply that the tails of the curve are infinite which we know is not true because it would not be possible in the real physical world. However, one needs to understand that the tails of the Gaussian distribution reduce to low probability values very quickly which implies that the probabilities of events at the end of the tails will be computed in the “billion of years” type of numbers. This is so far outside our area of interest that it makes little difference to using the Gaussian model on analyzing the worst case jitter scenario on a real data transmission system (e.g., how many failed bits in one day).

Another example on how to look at the Central Limit Theorem and the infinite tails of a Gaussian distribution is the following [Bernd Laquai, personal communication, 2008]:

Take N dice, roll them, and sum up the result. The distribution converges towards a Gaussian distribution. For example, for N=10 it already starts to look Gaussian. But notice that it is bounded because we have only 10 dice (the minimum value one can obtain is 10 and the maximum is 60). To have an unbounded Gaussian distribution we would need an infinite number of dice which is the requirement of the Central Limit Theorem.

With random jitter we have a similar behavior. It looks Gaussian but somehow it is still bounded because the number of contributors is limited.

A.1 The Gaussian Distribution

The Gaussian distribution probability density function¹ (PDF) is described by the following equation:

¹The probability density function is also sometimes referred to as the probability density distribution.

$$p(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \quad (\text{A.1})$$

where μ is the mean value of the Gaussian distribution (also sometimes referred to as the average value, although incorrectly) and σ^2 its variance.

The square root of the variance (σ) is called the standard deviation. The estimation of the standard deviation value based on several measurements is called the root mean square value (RMS). Figure A.1 shows the Gaussian distribution for different σ and μ values. Notice the effect that the standard deviation σ has on the spreading of the Gaussian distribution. If the distribution represents the noise associated with a measurement, clearly we would like to have the smallest possible variance value.

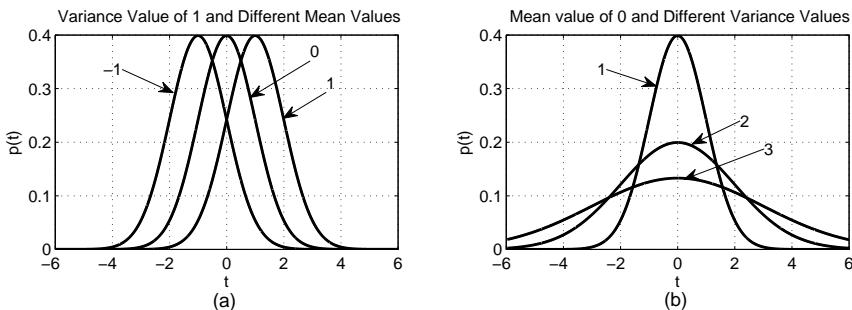


Figure A.1 (a) The Gaussian distribution with $\sigma = 1$ and different μ and (b) the Gaussian distribution with $\mu = 0$ and different standard deviation σ .

One important property of Gaussian distributions is that the sum of two Gaussian distributions X_1, X_2 will also be a Gaussian distribution with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$.

The probability density function is not of much help since typically one is interested in the probability of values smaller or higher than a certain t happening. For computing this probability value one uses the cumulative probability density function (CDF). The CDF function is computed by integrating the PDF function in the interval of interest. For the Gaussian distribution, the CDF is computed using the following equation where the objective is to compute the probability of the measured value to be between $-\infty$ and x :

$$P(t) = \int_{-\infty}^t p(x)dx = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \quad (\text{A.2})$$

In a real application, one typically starts with a set of measurements (sometimes called a time series) and from this set of measurements

a normalized histogram can be computed that will correspond to an approximation of the probability density function of the random process that generated those measured values. Figure A.2 shows one example of this procedure.

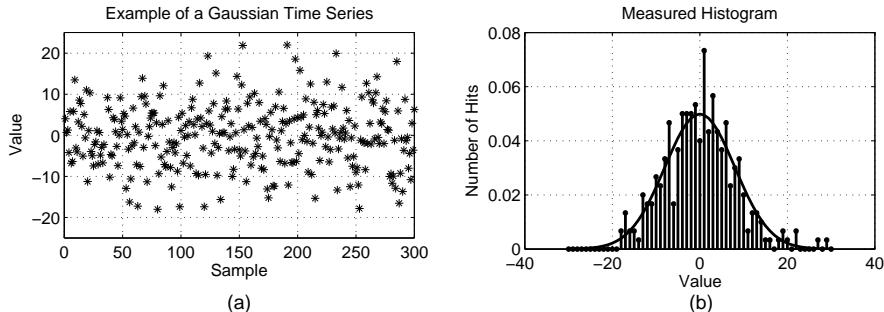


Figure A.2 Example of (a) a time series from a Gaussian distribution and (b) the computed normalized histogram (probability density function) from that time series with the theoretical Gaussian curve that would be obtained with an infinite series of values.

The best estimate for the mean value for the measured time series is:

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^N X_i \quad (\text{A.3})$$

where $\bar{\mu}$ is the estimate of the mean value, X_i is the measured value i of the set of measure values, and N is the total number of samples measured. The best estimate for the standard deviation² is:

$$\bar{\sigma} = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{\mu})^2} \quad (\text{A.4})$$

The important question on the previous estimation equations is the value of the associated error. For the standard deviation estimation the following formula applies [3]:

²There is an alternative definition for the standard deviation which is:

$$\bar{\sigma} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{\mu})^2}$$

This definition is sometimes called population standard deviation or sample standard deviation and it has some advantages in regard to (A.4) [3].

$$(\text{fractional uncertainty in } \bar{\sigma}) = \frac{1}{\sqrt{2(N-1)}} \quad (\text{A.5})$$

This result shows that multiple samples are needed to know with a good certainty the value of the standard deviation. For example, with three samples the uncertainty in the value of $\bar{\sigma}$ is 50%.

For the associated error to the estimate of the mean, the uncertainty is given by the standard deviation of the mean [3]:

$$\sigma_{\bar{\mu}} = \frac{\bar{\sigma}}{\sqrt{N}} \quad (\text{A.6})$$

This means that the error associated with the estimation of the mean $\bar{\mu}$ will be:

$$\bar{\mu} \pm \frac{\bar{\sigma}}{\sqrt{N}} \quad (\text{A.7})$$

Larger sample sizes reduce the error associated with the estimation of the average.

A.2 Computation of the BER for a System with Only Gaussian Random Jitter

In this section we will calculate the BER function for a system where only Gaussian random jitter exists. Figure A.3 shows a graph of the jitter histogram for the right and left edges of the data eye. The horizontal axis corresponds to the time instant used to strobe the data eye (t_S). Ideally to obtain the minimum BER, the strobing point should be on the time instant were the sum of the right and left jitter probability distributions is lowest, which in this case corresponds to the middle of the data eye.

The probability distributions for the jitter on the right and left edges of the data eye are described by the following Gaussian distributions:

$$p_{left}(t) = \frac{1}{\sqrt{2\pi}\sigma_{left}} e^{-\frac{(t-\mu_{left})^2}{2\sigma_{left}^2}} \quad (\text{A.8})$$

$$p_{right}(t) = \frac{1}{\sqrt{2\pi}\sigma_{right}} e^{-\frac{(t-\mu_{right})^2}{2\sigma_{right}^2}} \quad (\text{A.9})$$

The distance between the means of the two Gaussian distributions will correspond to the bit period. σ_{left} and σ_{right} are always assumed to be equal and in the remainder of this section we will then assume $\sigma_{left} = \sigma_{right} = \sigma$.

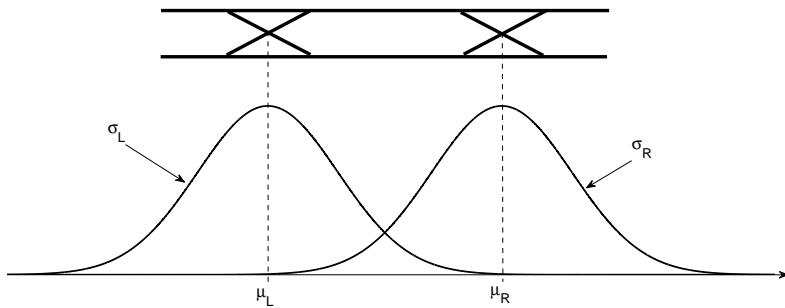


Figure A.3 Jitter probability distribution on the left and right transitions of the data eye.

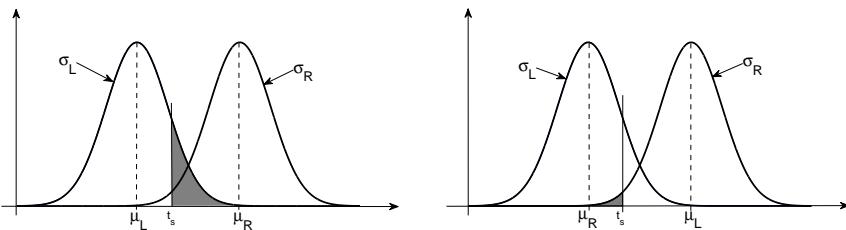
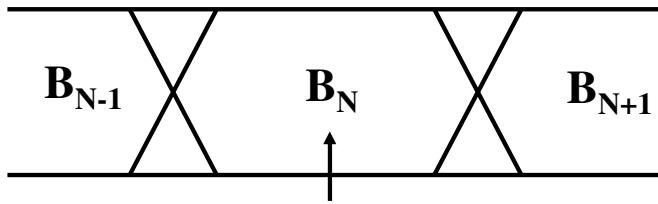


Figure A.4 Computation of the BER function.

To compute the BER value for a given strobe position we need to integrate the contributions from the left and right Gaussian distributions (see Figure A.4) including also the probability that the measured bit is different from the expected bit (see Figure A.5). This factor is usually named transition density and it is important since although we might be strobing the wrong bit (timing-wise) due to the jitter, the bit value that is measured might still be identical to the expected bit value. On a truly random binary sequence the transition density value is 0.5 and on a clock pattern it is 1.0. The BER value for a given strobing time t can then be computed by the following equation:

$$\begin{aligned} BER(t) = & p(b_N \neq b_{N-1}) \int_t^{+\infty} p_{left}(x) dx \\ & + p(b_N \neq b_{N+1}) \int_{-\infty}^t p_{right}(x) dx \end{aligned} \quad (\text{A.10})$$

where p_{left} and p_{right} represent the probability distributions for the left and right edges of the data eye. $p(b_N \neq b_{N+1})$ and $p(b_N \neq b_{N-1})$ are the probabilities of the n th bit being different from bit $(n+1)$ th or bit $(n-1)$ th (i.e., the transition density).



Clock Pattern: $P(b_N \neq b_{N+1}) = P(b_N \neq b_{N-1}) = 1$

True Random Pattern: $P(b_N \neq b_{N+1}) = P(b_N \neq b_{N-1}) = 0.5$

Figure A.5 Probability of adjacent bits being different.

If we suppose a Gaussian distribution for the random jitter, and assuming the same standard deviation for the right and left Gaussian ($\sigma = \sigma_{right} = \sigma_{left}$) and the same value for the transition density ($T_D = p(b_N \neq b_{N+1}) = p(b_N \neq b_{N-1})$), we obtain the following result:

$$\begin{aligned} BER(t) = & T_D \frac{1}{\sqrt{2\pi}\sigma} \int_t^{+\infty} e^{-\frac{(t-\mu_{left})^2}{2\sigma^2}} dt \\ & + T_D \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^t e^{-\frac{(t-\mu_{right})^2}{2\sigma^2}} dt \end{aligned} \quad (\text{A.11})$$

We will now use the following variable change:

$$x_1 = \left(\frac{t - \mu_{left}}{\sqrt{2}\sigma} \right); \quad x_2 = \left(\frac{t - \mu_{right}}{\sqrt{2}\sigma} \right) \quad (\text{A.12})$$

to obtain the following expression for the BER:

$$\begin{aligned} BER(t) = & \frac{T_D}{\sqrt{\pi}} \int_{\frac{t-\mu_{left}}{2\sigma}}^{+\infty} e^{-x_1^2} dx_1 \\ & + \frac{T_D}{\sqrt{\pi}} \int_{\frac{\mu_{right}-t}{2\sigma}}^{+\infty} e^{-x_2^2} dx_2 \end{aligned} \quad (\text{A.13})$$

Using the complementary error function ($erfc$) defined in Section A.4, the BER function can then be described by the following equation:

$$\begin{aligned} BER(t) = & \frac{T_D}{2} erfc \left(\frac{t - \mu_{left}}{\sqrt{2}\sigma} \right) \\ & + \frac{T_D}{2} erfc \left(\frac{\mu_{right} - t}{\sqrt{2}\sigma} \right) \end{aligned} \quad (\text{A.14})$$

Figure A.6 shows the BER function, also known as the BER bathtub curve, for the following parameters: $\mu_{right} = 600$, $\mu_{left} = 200$, and $\sigma = 8$ using a clock pattern ($T_D = 1$).

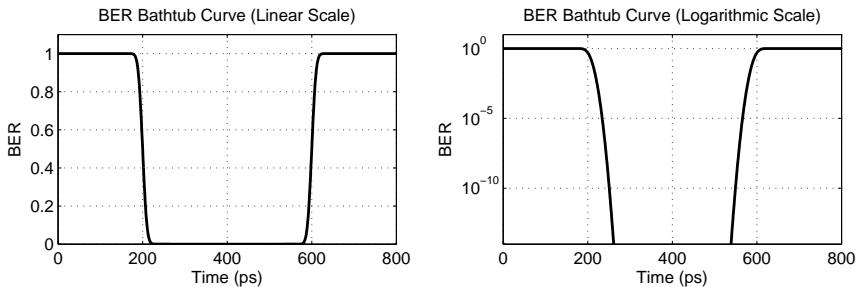


Figure A.6 BER bathtub curve for $\mu_{right} = 600$ ps, $\mu_{left} = 200$ ps, and $\sigma = 8$ ps using a clock pattern ($p(b_N \neq b_{N+1}) = p(b_N \neq b_{N-1}) = 1$) plotted in a linear BER scale (left) and logarithmic BER scale (right).

A.3 Computation of the $\alpha(BER)$ Value

In a system with Gaussian random jitter and deterministic jitter, the total jitter value at a given BER is usually computed through the following equation [4, 5]:

$$TJ = DJ + \alpha(BER) RJ \quad (\text{A.15})$$

where DJ is the peak-to-peak value of the deterministic jitter and RJ is the standard deviation (σ) or RMS jitter value of the random jitter Gaussian distribution.

The $\alpha(BER)$ factor is a value that is used to compute a “peak-to-peak” value of the random jitter for a given BER. This is exemplified in Figure A.7 where we define the peak-to-peak value for a Gaussian distribution as the interval $[-t, t]$ where $1 - BER$ of the jitter samples reside.

From Figure A.7 we can extract the following equation:

$$\frac{1}{2}BER = T_D \frac{1}{\sqrt{2\pi}\cdot\sigma} \int_t^{+\infty} e^{-\frac{x^2}{2\sigma^2}} dx \quad (\text{A.16})$$

Note that we need to include the transition density factor T_D which is the probability of the expected value of the bit under measurement, which may or may not be different from the previous or next bit. Equation (A.16) can be simplified as the following equation:

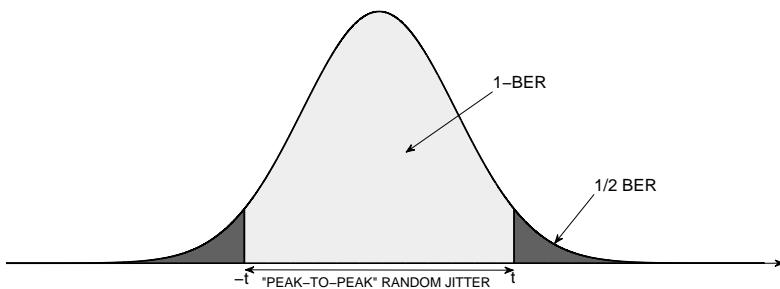


Figure A.7 Computing the $\alpha(BER)$ value.

$$BER = T_D \operatorname{erfc}\left(\frac{t}{\sqrt{2}\sigma}\right) \quad (\text{A.17})$$

$$t = \sqrt{2} \operatorname{erfc}^{-1}\left(\frac{BER}{T_D}\right)\sigma \quad (\text{A.18})$$

The “peak-to-peak” value of the random jitter can now be computed:

$$RJ_{PK-PK} = 2\sqrt{2} \operatorname{erfc}^{-1}\left(\frac{BER}{T_D}\right)\sigma \quad (\text{A.19})$$

From (A.15) the $\alpha(BER)$ is:

$$\alpha(BER) = 2\sqrt{2} \operatorname{erfc}^{-1}\left(\frac{BER}{T_D}\right) \quad (\text{A.20})$$

For example, if we assume a true random pattern ($T_D = 0.5$), for a BER of 10^{-12} the $\alpha(BER)$ value is 14.069. Table A.1 shows some of the typical used values for $\alpha(BER)$ for a transition density of 0.5 and 1. In most applications a transition density value of 0.5 is explicitly assumed.

Table A.1
Some Typical Used Values for $\alpha(BER)$

BER	$\alpha(BER)$, $T_D = 0.5$	$\alpha(BER)$, $T_D = 1$
10^{-9}	11.9956	12.2188
10^{-12}	14.0690	14.2610
10^{-16}	16.4442	16.6096

Figure A.8 shows the link between a specific desired BER value, the percentage of jitter “events” that need to be contained on the interval that guarantees that BER value, and the corresponding $\alpha(BER)$ factor that

multiplied by the standard deviation σ provides the random jitter “peak-to-peak” value for a specific BER.

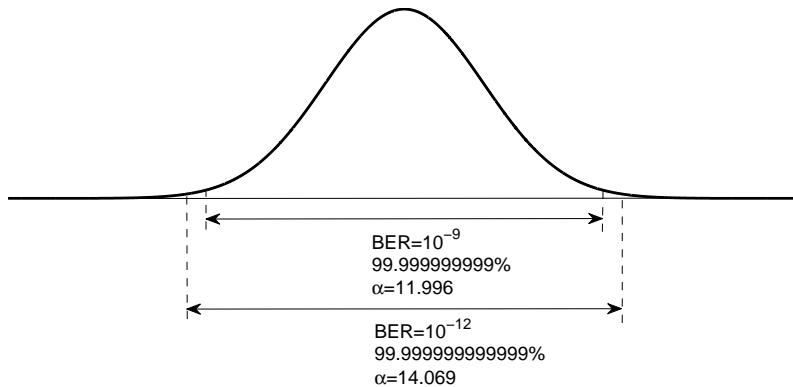


Figure A.8 Relationship between the BER value and the percentage of occurrences that need to be included for the computation of the $\alpha(BER)$.

A.4 Properties of the Error Function $erf(x)$ and Complementary Error Function $erfc(x)$

The error function $erf(x)$ is defined by the following equation:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy \quad (\text{A.21})$$

It has the following properties:

$$\begin{aligned} erf(0) &= 0 \\ erf(+\infty) &= 1 \\ erf(-\infty) &= -1 \end{aligned} \quad (\text{A.22})$$

The complementary error function $erfc(x)$ is defined by the following equation:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-y^2} dy \quad (\text{A.23})$$

And it has the following property:

$$\int_0^x erfc(y) dy = x erfc(x) + \frac{1}{\sqrt{\pi}} (1 - e^{-x^2}) \quad (\text{A.24})$$

The error function and the complementary error function are related by the following expression:

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) \quad (\text{A.25})$$

The inverse of the complementary error function is also important.

$$\operatorname{erfc}^{-1}(\operatorname{erfc}(x)) = x \quad (\text{A.26})$$

For values of $x \gg 1$ the complementary error function can be approximated by the following expression:

$$\operatorname{erfc}(x) \approx \frac{2}{\sqrt{\pi}} \frac{e^{-x^2}}{x} \text{ for } x \gg 1 \quad (\text{A.27})$$

Note that the used definitions of the erf and erfc functions are the same as the ones used by software packages like MATLAB and Mathematica. Note that some authors use slightly different definitions.

References

- [1] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.
- [2] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. John Wiley and Sons, 1987.
- [3] J. R. Taylor, *An Introduction to Error Analysis*. University Science Books, 1997.
- [4] Maxim, “Converting Between RMS and Peak-to-Peak Jitter at a Specified BER,” *Application Note HFAN-4.0.2*, 2000.
- [5] INCITS, *Fiber Channel – Methodology for Jitter and Signal Quality Specification – MJSQ*, 2004. Technical Report for Information Technology.

B

The Dual Dirac Model and RJ/DJ Separation

This appendix presents in detail the dual Dirac jitter model and the random and deterministic jitter separation algorithm based on this model which is used in several standards [1–3].

B.1 The Dual Dirac Jitter Model

The dual Dirac jitter model was developed to model the random and deterministic jitter components of a digital signal. The reasoning comes from the fact that the deterministic components of the jitter will be bounded. This means that the simplest model for deterministic jitter is a kind of ping-pong jitter that jumps between the two boundaries of the deterministic jitter range as shown in Figure B.1.

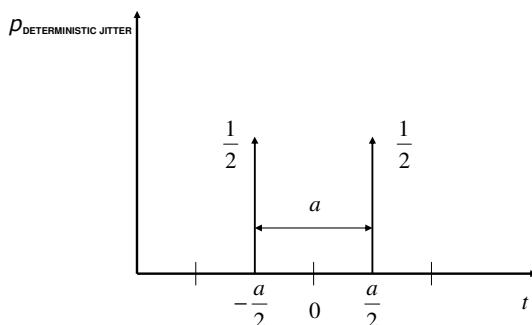


Figure B.1 The dual Dirac deterministic jitter model for a peak-to-peak jitter value of a .

This behavior can be modeled by the use of the Dirac delta function. A Dirac delta function is part of what is called generalized functions [4]. A Dirac delta function can be visualized as a zero width pulse at a specific point in time. Mathematically, the deterministic jitter model will have the following form:

$$DJ = \frac{1}{2}\delta\left(t - \frac{a}{2}\right) - \frac{1}{2}\delta\left(t + \frac{a}{2}\right) \quad (\text{B.1})$$

where $\delta(t)$ is the Dirac delta function and a is the peak-to-peak value of the deterministic jitter. This model is referred to as the dual Dirac jitter model.

We can now complete the model by convolving the random jitter in the form of a Gaussian distribution as shown in Figure B.2.

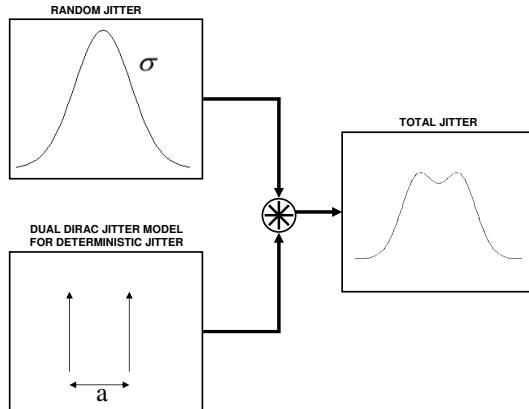


Figure B.2 The dual Dirac jitter model.

To obtain an analytical expression for this model one needs to perform the convolution of the dual Dirac model for the deterministic jitter with the Gaussian distribution model for the random jitter (A.1) with a mean value of zero ($\mu = 0$). Mathematically, the model is now:

$$\begin{aligned} p_{\text{jitter}}(t) &= p_{\text{random}}(t) * p_{\text{dual_dirac}}(t) \\ &= \int_{-\infty}^{+\infty} p_{\text{random}}(\tau)p_{\text{dual_dirac}}(t - \tau)d\tau \end{aligned} \quad (\text{B.2})$$

Substituting the probability distributions we obtain:

$$p(t) = \int_{-\infty}^{+\infty} \left[\frac{1}{2} \delta\left(\tau - \frac{a}{2}\right) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-\tau)^2}{2\sigma^2}} + \frac{1}{2} \delta\left(\tau + \frac{a}{2}\right) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-\tau)^2}{2\sigma^2}} \right] d\tau \quad (\text{B.3})$$

Solving this integral we obtain the following expression for the total jitter distribution of the dual Dirac jitter model:

$$p(t) = \frac{1}{2\sqrt{2\pi}\sigma} \left[e^{-\frac{(t+\frac{a}{2})^2}{2\sigma^2}} + e^{-\frac{(t-\frac{a}{2})^2}{2\sigma^2}} \right] \quad (\text{B.4})$$

Figure B.3 presents the total jitter probability distribution for the parameters $\sigma = 8$ and $a = 20$.

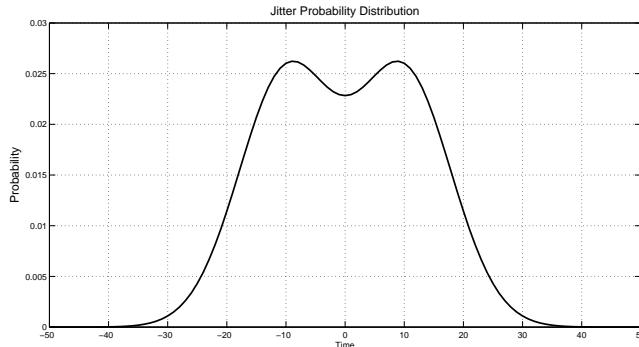


Figure B.3 Total jitter probability distribution for a dual Dirac jitter model ($\sigma = 8$ and $a = 20$).

Using the computed distribution it is now possible to write the jitter distributions for the right and left edges of the data eye.

$$p_{right}(t) = \frac{1}{2\sqrt{2\pi}\sigma} \left[e^{-\frac{(t+\frac{a}{2}-E_{right})^2}{2\sigma^2}} + e^{-\frac{(t-\frac{a}{2}-E_{right})^2}{2\sigma^2}} \right]$$

$$p_{left}(t) = \frac{1}{2\sqrt{2\pi}\sigma} \left[e^{-\frac{(t+\frac{a}{2}-E_{left})^2}{2\sigma^2}} + e^{-\frac{(t-\frac{a}{2}-E_{left})^2}{2\sigma^2}} \right] \quad (\text{B.5})$$

where E_{left} and E_{right} correspond to the left and right crossing points of the data eye. $E_{left} - E_{right}$ should correspond to the bit period. The BER for a sampling point t is then computed by the following expression where T_D is the transition density (see Appendix A).

$$BER(t) = T_D \int_t^{+\infty} p_{left}(t) dt + T_D \int_{-\infty}^t p_{right}(t) dt \quad (B.6)$$

Substituting by the respective probability distributions we obtain:

$$BER(t) = \frac{T_D}{2\sqrt{2\pi}\sigma} \int_t^{+\infty} \left[e^{\frac{(t-\frac{a}{2}-E_{left})^2}{2\sigma^2}} + e^{\frac{(t+\frac{a}{2}-E_{left})^2}{2\sigma^2}} \right] dt + \frac{T_D}{2\sqrt{2\pi}\sigma} \int_{-\infty}^t \left[e^{\frac{(t-\frac{a}{2}-E_{right})^2}{2\sigma^2}} + e^{\frac{(t+\frac{a}{2}-E_{right})^2}{2\sigma^2}} \right] dt \quad (B.7)$$

Applying the same techniques used in Section A.2, we then obtain:

$$BER(t) = \frac{T_D}{4} \left[erfc\left(\frac{t - \frac{a}{2} - E_{left}}{\sqrt{2}\sigma}\right) + erfc\left(\frac{t + \frac{a}{2} - E_{left}}{\sqrt{2}\sigma}\right) \right] + \frac{T_D}{4} \left[erfc\left(\frac{E_{right} + \frac{a}{2} - t}{\sqrt{2}\sigma}\right) + erfc\left(\frac{E_{right} - \frac{a}{2} - t}{\sqrt{2}\sigma}\right) \right] \quad (B.8)$$

Figure B.4 shows the BER bathtub curve for the following parameters: $E_{right}=600$ ps, $E_{left}=200$ ps, $\sigma=8$ ps, and $a = 100$ ps using a clock data pattern $T_D = 1$. The bit period is 400 ps.

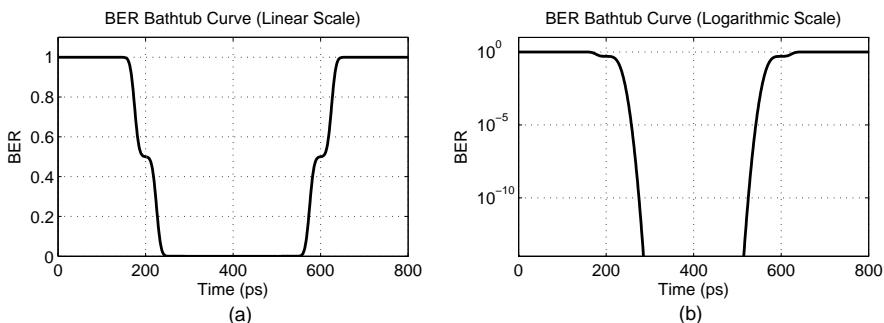


Figure B.4 BER bathtub curve for the dual Dirac jitter model in (a) a linear BER scale and (b) a logarithmic BER scale.

B.2 RJ/DJ Separation with the Q-Factor Algorithm

The idea behind RJ/DJ separation using the BER bathtub curve is that below a certain BER value, the BER will be dominated by the random jitter which follows a Gaussian distribution. In this region the BER can be approximated by a normalized complementary error function:

$$\begin{aligned} BER_{RJ}^{left}(t) &= N_{left}erfc\left(\frac{t - \mu_{left}}{\sqrt{2}\sigma_{left}}\right) \\ BER_{RJ}^{right}(t) &= N_{right}erfc\left(\frac{\mu_{right} - t}{\sqrt{2}\sigma_{right}}\right) \end{aligned} \quad (\text{B.9})$$

We will now assume a normalization factor corresponding to the dual Dirac jitter model in the previous expressions ($N_{left} = T_D/4$ and $N_{right} = T_D/4$).

Let us now define the Q-factor¹ [5] for each one of the right and left sides of the BER bathtub curve using the following equations:

$$Q_{right} = \frac{\mu_{right} - t}{\sigma_{right}}; Q_{left} = \frac{t - \mu_{left}}{\sigma_{left}} \quad (\text{B.10})$$

We obtain the following relation between the BER and the Q-factor:

$$BER_{RJ}(t) = \frac{T_D}{4}erfc\left(\frac{Q}{\sqrt{2}}\right) \quad (\text{B.11})$$

By re-arranging the terms we obtain:

$$\frac{4}{T_D}BER_{RJ} = 1 - erf\left(\frac{Q}{\sqrt{2}}\right) \Leftrightarrow Q = \sqrt{2} erf^{-1}\left(1 - \frac{4}{T_D}BER_{RJ}\right) \quad (\text{B.12})$$

We can look at this expression as a mapping of the BER bathtub curve into the Q-factor space:

$$\begin{aligned} Q &= F(BER_{RJ}) \\ F(x) &= \sqrt{2} erf^{-1}\left(1 - \frac{4}{T_D}x\right) \end{aligned} \quad (\text{B.13})$$

Figure B.5 presents a graphic example of this mapping. The important factor is that an $erfc$ function (i.e., a Gaussian jitter distribution) is a straight line when represented in the Q-factor space (i.e., when we are on the

¹Also referred to as the Q-scale.

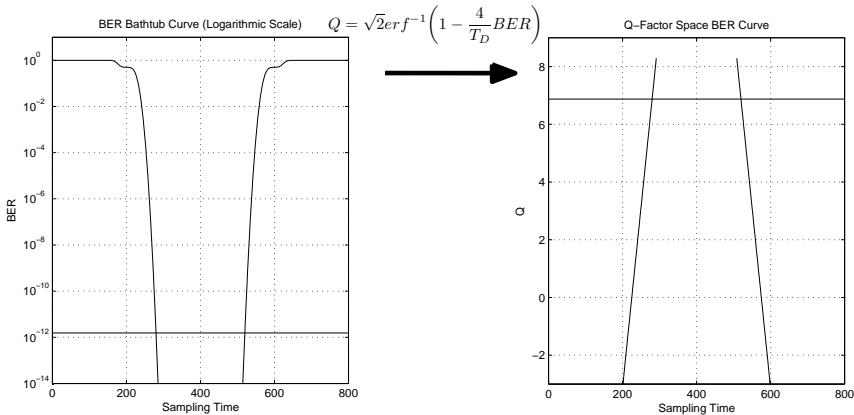


Figure B.5 Mapping the BER bathtub curve into the Q-factor space.

“Gaussian” part of the BER bathtub curve). Given the definition of the Q-factor, it is now easy to obtain the random and deterministic jitter values. First it is necessary to perform a linear fitting to the left and right curve obtained after the transformation (see [6]), but only to the part where the random jitter is the dominant factor (below a certain BER). Let us now suppose we compute the fitting parameters (A_{right} , A_{left} , B_{right} , B_{left}) shown on (B.14).

$$\begin{aligned} Q_{right}(t) &= t A_{right} + B_{right} \\ Q_{left}(t) &= t A_{left} + B_{left} \end{aligned} \quad (\text{B.14})$$

Given the previous definitions for the right and left Q-factor it is then possible to create an equality between the fitting parameters and the Q-factor of each curve.

$$\begin{aligned} \frac{\mu_{right} - t}{\sigma_{right}} &= t A_{right} + B_{right} \\ -\frac{t}{\sigma_{right}} + \frac{\mu_{right}}{\sigma_{right}} &= t A_{right} + B_{right} \end{aligned} \quad (\text{B.15})$$

$$\begin{aligned} \frac{t - \mu_{left}}{\sigma_{left}} &= t A_{left} + B_{left} \\ \frac{t}{\sigma_{left}} - \frac{\mu_{left}}{\sigma_{left}} &= t A_{left} + B_{left} \end{aligned} \quad (\text{B.16})$$

From the above equation it is possible to infer the values of the random jitter Gaussian distribution standard deviation and mean of each curve from the obtained fitting parameters:

$$\begin{aligned}\sigma_{right} &= -\frac{1}{A_{right}} \\ \sigma_{left} &= \frac{1}{A_{left}} \\ \mu_{right} &= B_{right} \sigma_{right} \\ \mu_{left} &= -B_{left} \sigma_{left}\end{aligned}\tag{B.17}$$

Then, the random jitter RMS value will correspond to the mean of the left and right standard deviations:

$$RJ = \frac{\sigma_{right} + \sigma_{left}}{2}\tag{B.18}$$

To compute the deterministic jitter it is necessary to compute the spacing between the two Gaussian distributions. But given that each Gaussian will reside on one of the sides of the BER bathtub curve, the following expression will provide the deterministic jitter value:

$$DJ = 1.0 \text{ UI} - (\mu_{right} - \mu_{left})\tag{B.19}$$

The advantage of this approach is that by transforming the Gaussian fitting into a linear fitting, it allows the computation of the random and deterministic jitter separation to be done in a very fast way given that there is an easy and fast solution for the linear fitting of a set of points [6].

As a final point, it is important to note that this jitter separation algorithm is exact as long as the model it is based on is exact. In a real application the double Dirac jitter model is an approximation which impacts the accuracy of the obtained jitter separation values [7–10].

References

- [1] M. P. Li, J. Wilstrup, R. Jessen, and D. Petrich, “A New Method for Jitter Decomposition Through Its Distribution Tail Fitting,” *IEEE International Test Conference*, 1999.
- [2] G. Haensel, K. Stieglbauer, G. Schulze, and J. Moreira, “Implementation of an Economic Jitter Compliance Test for Multi-Gigabit Device on ATE,” *IEEE International Test Conference*, 2004.
- [3] Agilent Technologies, “Jitter Analysis: The Dual-Dirac Model, RJ/DJ, and Q-Scale,” 2005. White Paper.
- [4] R. P. Kanwal, *Generalized Functions: Theory and Applications*. Birkhauser, 2004.
- [5] N. S. Bergano, F. W. Kerfoot, and C. R. Davidson, “Margin Measurements in Optical Amplifier Systems,” *IEEE Photonics Technology Letters*, vol. 5, no. 3, 1993.

- [6] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, 2003.
- [7] R. Stephens, "What the Dual-Dirac Model Is and What It Is Not," *Tektronix 360 Knowledge Series*, 2006.
- [8] M. Miller and M. Schnecker, "A Comparison of Methods for Estimating Total Jitter Concerning Precision, Accuracy and Robustness," *IEC DesignCon*, 2007.
- [9] T. J. Yamaguchi, K. Ichiyama, H. X. Hou, and M. Ishida, "A Robust Method for Identifying a Deterministic Jitter Model in a Total Jitter Distribution," *IEEE International Test Conference*, 2009.
- [10] T. J. Yamaguchi, H. X. Hou, K. Takayama, D. Armstrong, M. Ishida, and M. Soma, "An FFT-Based Jitter Separation Method for High-Frequency Jitter Testing with a 10X Reduction in Time," *IEEE International Test Conference*, 2007.

C

Pseudo-Random Bit Sequences and Other Data Patterns

This appendix provides a background on pseudo-random bit sequences (PRBS) and a few of the most common data patterns used for testing and characterization of high-speed digital interfaces.

C.1 Pseudo-Random Bit Sequences

A pseudo-random bit sequence data pattern is frequently used for the characterization of high-speed digital I/O interfaces. A PRBS data pattern generates all possible bit sequences for a given length. This means that for a given word with a set length of 7 bits, for example, the corresponding PRBS sequence will contain all the possible permutations of 1s and 0s for a 7-bit word. A PRBS sequence is easily implemented in hardware or software using a linear feedback shift register (LFSR) as shown in Figure C.1 [1, 2].

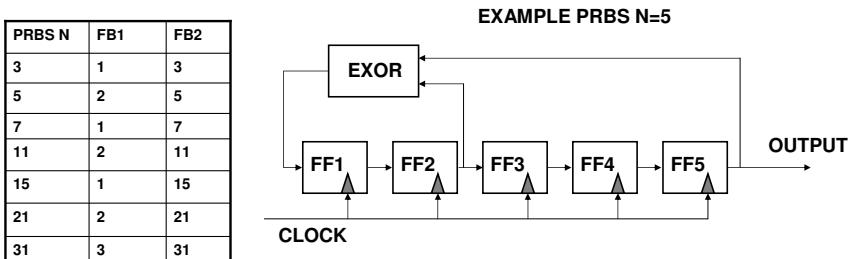


Figure C.1 Basic block diagram example of a PRBS implementation with a feedback shift register and the design parameters for different word length PRBS data patterns.

PRBS sequences are based on the mathematical theory of “irreducible polynom of length N .” Each PRBS $2^N - 1$ data sequence will contain all the N -bit long sequences (except $N \times 0$) and its cycle length will be $2^N - 1$. After this length the sequence will start again. Note that although the data word length is fixed for each N , the exact full length PRBS sequence will depend on the initial seed (initial values set in the storage elements of the LFSR) provided to the algorithm. Sometimes a PRBS $2^N - 1$ sequence is referred to as a PRBS N sequence for a compact notation. But it is important not to mistake it for a PRBS 2^N data pattern which is different from a PRBS $2^N - 1$ pattern.

Given its simple hardware implementation, PRBS sequences are used extensively in test and measurement equipment. But it is important to understand their properties when choosing a PRBS sequence to stimulate a DUT. Figure C.2 shows a comparison of the power spectrum for two different PRBS sequences (PRBS $2^{31} - 1$ and PRBS $2^7 - 1$) at 5 Gbps. One notes that the spectrum for both PRBS sequences has the same bandwidth (the bandwidth is defined by the signal rise time which is the same for both patterns). The spectrum may look similar for both patterns with the expected sinc function shape¹; however, on closer inspection of the details, it is possible to observe that the PRBS $2^{31} - 1$ sequence has additional spectrum lines since it can generate a higher number of symbol combinations.

The fact that a PRBS $2^{31} - 1$ pattern has a higher number of spectral lines that are spread through the frequency band has consequences on the measured performance of a DUT or a signal path that is stimulated with a PRBS pattern. Figure C.3 shows the data eye diagrams obtained from a lossy signal path with a PRBS $2^{31} - 1$ and a PRBS $2^7 - 1$ data pattern. Although the stimulus driver performance is the same for both patterns (i.e., intrinsic jitter, rise-time, and so forth), the PRBS $2^{31} - 1$ data pattern shows worse performance than the PRBS $2^7 - 1$ pattern since its power spectrum “stresses” the DUT more.

C.2 Pseudo-Random Word Sequences

Pseudo-random word sequences (PRWS) are a special application of PRBS sequences. One way to understand PRWS is to look at their typical usage which is the test of applications like a multiplexer. Figure C.4 shows the need to define a proper bit sequence for the 2-bit word that is an input to the multiplexer in a way that guarantees that the pattern at the output of the multiplexer is a PRBS sequence [3].

¹More details in Section 2.2.

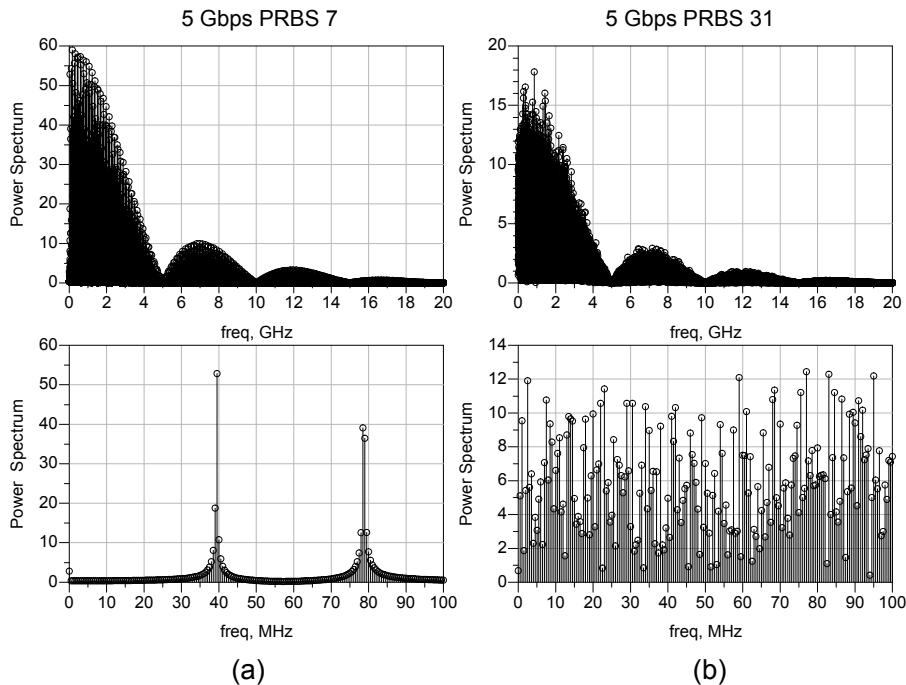


Figure C.2 Comparison of the power spectrum of (a) a PRBS $2^7 - 1$ pattern with (b) a PRBS $2^{31} - 1$ pattern.

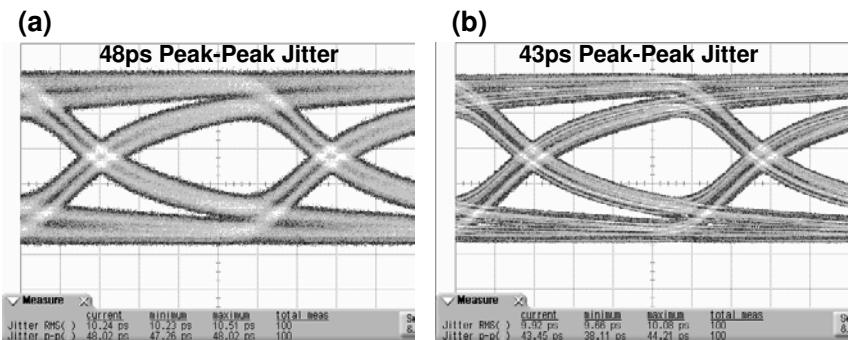


Figure C.3 Comparison of the measured data eye diagram of a lossy signal path with (a) a PRBS $2^{31} - 1$ and (b) a PRBS $2^7 - 1$ pattern.

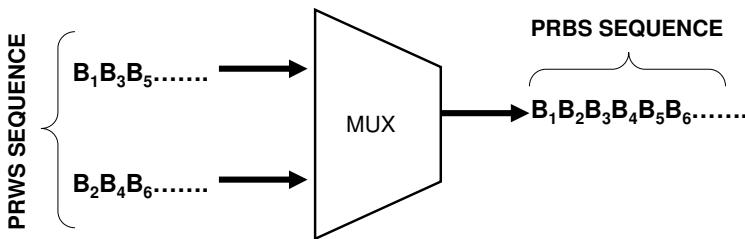


Figure C.4 The need to use a PRWS data pattern when testing a multiplexer.

Likewise for the PRBS case, PRWS generators are easily implemented into hardware through shift-registers [4].

C.3 Other Important Patterns

Several patterns are important enough that their names are used instead of the corresponding bit sequence. One is the PCI Express compliance pattern shown in Figure C.5.



Figure C.5 The PCI Express compliance pattern (001111101010101010101100000010101010101) [5].

Another pattern is the K28.5 pattern shown in Figure C.6 that is generated by two consecutive K28.5 characters defined in the 8B/10B code space (for more details see Appendix D).

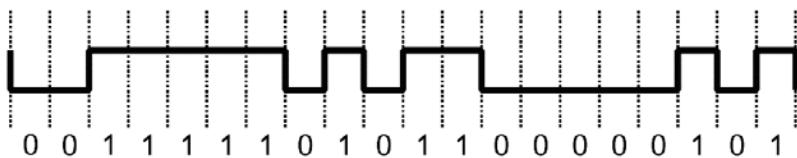


Figure C.6 The K28.5 pattern.

Other important patterns are the continuous random test pattern (CRPAT) and the continuous jitter test pattern (CJPAT). Both patterns are used in several standards (e.g., 10 Gigabit Ethernet). These patterns are not simple and their definitions can be found in [6]. The CRPAT has a broad and relatively flat spectral content that is targeted to provide high frequency jitter components

like ISI but with little content at low frequencies. The CJPAT in contrast was designed to have spectral content at low frequencies where the DUT PLL bandwidth will reside while also containing high frequency components.

References

- [1] S. W. Golomb, *Shift Register Sequences*. Aegean Park Press, 1982.
- [2] D. Derickson and M. Mueller, *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization*. Prentice-Hall, 2007.
- [3] Agilent Technologies, “Agilent Technologies ParBERT 81250 Mux/Demux Application,” *Application Note 5968-9695*, 2004.
- [4] W. J. McFarland, K. H. Springer, and C.-S. Yen, “1-Gword/s Pseudorandom Word Generator,” *IEEE Journal of Solid State Circuits*, vol. 24, June 1989.
- [5] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [6] IEEE, *802.3 Part3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, 2005.

D

Coding, Scrambling, Disparity, and CRC

Coding or scrambling of bit sequences has the purpose to adapt the properties of a data stream as much as possible to limitations and requirements of the signal path and the receiver that has to interpret the signal it receives. In real systems, signal paths have certain transmission characteristics that are mainly determined by the materials and geometries used to implement the signal path. Since system designers are usually not free in the selection of the materials and geometries they use for signal path implementation, they are also limited in their influence on the transmission characteristic of the signal path. In such cases, coding and scrambling can help to modify the data stream properties in a way that the characteristic of the signal path has less influence on the transmitted signal.

One example of this is to spread out the spectral energy of a data stream over a wider frequency range using scrambling to minimize EMI caused by peaks in the spectral power distribution of a data stream due to repetitive data that is transmitted. Another example to minimize the influence of the signal path on the transmitted signal is coding to achieve DC-balance in the data stream. Each signal path has an inevitable capacitance that is charged and discharged depending on the data stream which is transmitted. The baseline voltage observed on the transmission path will be modulated with a lowpass filter characteristic according to the charging and discharging of the signal path.

If a pattern contains long sequences of charging bits (ones) and/or long sequences of discharging bits (zeros), there is the possibility that the baseline voltage of the signal path wanders away too far from the desired voltage level. This can cause errors in the data transmission because bits with a value other

than the ones that caused this baseline wander are not recognized correctly anymore. One way to avoid this behavior is to code the transmitted data stream in a way that limits the number of consecutive bits having the same logical value and to make sure that the number of zeros and ones in the transmitted data stream are balanced over a certain range of bits. A measure for this balance over a certain bit range is the so-called disparity which will be explained in more detail in Section D.1. The amount to which subsequent bits of the same logical value are limited and the observation range for the disparity calculation a transmission path requires depends on its lowpass filter characteristics and thus on its capacitance.

As mentioned already, besides optimizing data stream properties to match signal path characteristics, coding and scrambling also are used to guarantee signal conditions that device receivers require to be able to correctly interpret the received signals. An example for this use scenario are embedded clock receivers that have to extract the clock phase information for the received signal from this signal itself. The phase information from the received data stream is used to align the phase of the device internal sampling clock to the actual bit positions in the received data.

If such a receiver cannot readjust its sampling clock phase regularly to the incoming data, there is the potential to miss incoming data bits or to sample more bits than there actually are in the data stream due to phase offsets between the incoming data stream and the device's sampling clock. In order to be able to extract the required clock phase information, embedded clock receivers rely on transitions in the received data stream. Thus, these receivers typically require that the data streams they receive contain transitions that are not too far apart from each other. Scrambling or coding usually is applied to ensure that there is a sufficient amount of transitions in the data stream and that the distance between the transitions is not exceeding the limits required by the receiver.

D.1 Disparity

Disparity is a term used to define the relation between the number of zeros and ones over a sequence of binary data (e.g., a byte or a data word). The disparity for a given bit sequence is calculated by increasing a counter for each one in the analyzed data and by decreasing this counter for each zero as shown in Figure D.1. With this calculation rule, the disparity number that is obtained represents a measure for how many more ones are contained in an analyzed data stream section than there are zeros.

If the disparity calculation as described above with a counter increment for each one and a counter decrement for each zero is applied to a continuously

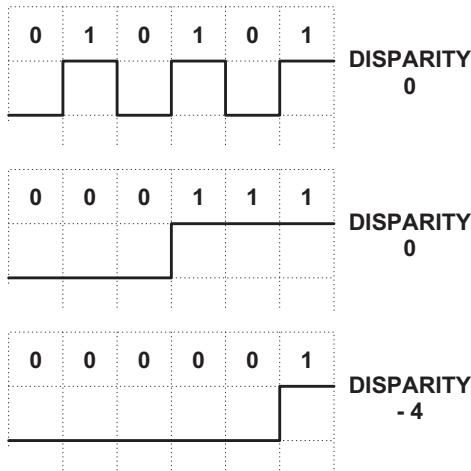


Figure D.1 Example of computing the disparity in a 6-bit symbol.

running data stream, a disparity number is obtained after each single bit. This disparity information is also referred to as running disparity. The running disparity indicates at each instance during transmission of a data stream the difference between transmitted ones and zeros over the whole data transmitted so far.

As already discussed before, it is important for some applications to keep the average numbers of zeros and ones over a transmitted data stream balanced. In order to keep such a balanced or neutral disparity, the easiest approach would be to use coding that transfers disparity-unbalanced data words into disparity-balanced codes. However, it is obvious that with such a truly balanced code space, the amount of bits required per code word would be significantly larger than the number of bits in the original data word because only code words having the same amount of zeros and ones would be allowed. Since the effects such as baseline wander that result from unbalanced disparity are usually found in the DC space and the data rates we are talking about are in the Gbps-range, it is typically not required that the running disparity is kept neutral for each and every transmitted bit to achieve DC balance over the data stream. Usually it is sufficient to ensure by suitable code selection that the running disparity is kept at a minimum after transmission of a code word. One example for a code that follows this methodology to create a DC balanced code space is the 8B/10B code which was developed by IBM in 1983. In the following section we will give an overview of this 8B/10B code. More detailed information can be obtained in the original paper describing the 8B/10B coding [1].

D.2 8B/10B Coding

8B/10B is a line code that maps 8-bit symbols to 10-bit codes to achieve DC balance, bounded disparity, and provide enough transitions to allow for reasonable clock recovery. As a result of coding, any combination of two code words has neutral disparity and the longest sequence of bits with the same logical value (runlength) is limited to 5 consecutive bits.

The 8B/10B coding scheme transmits one byte of data as a 10-bit code which also is called symbol or character. The lower 5 bits of the transmitted byte are encoded into a 6-bit group and the upper 3 bits of the byte are encoded into a 4-bit group. Both of these code groups are concatenated together to form the 10-bit symbol that is transmitted. The data symbols are often referred to as D_{xx}.y where the number xx ranges from 0 to 31 and represents the value of the lower 5 bits before conversion. The number y is in the range from 0 to 7 and represents the value of the upper 3 bits of the transmitted byte before conversion. Thus, the original byte value can easily be determined from the symbol name.

Standards using the 8B/10B encoding also define special symbols (or control characters) that can be sent in place of a data symbol. They are often used to indicate end-of-frame, link idle, skip, and similar link-level conditions. These control characters are referred to as K_{xx}.y and have different encoding from any of the D_{xx}.y symbols. As a result of the differences in encoding data and control symbols, the maximum runlength of five for 8B/10B codes can only occur within control symbols whereas data symbols have a maximum runlength of four. This distinct property of some control symbols allows easy identification of these in a data stream and typically is used for synchronization to the data stream by the standards that apply to 8B/10B coding.

One goal when developing the 8B/10B code was to achieve DC balance for the coded data streams. This is achieved by tracking the running disparity for each code segment used and selecting one of up to two code alternatives for the next code segment based on the current status of the running disparity. Two code alternatives are usually available for all 3- or 5-bit portions of the original byte that translate into a code segment which does not have neutral disparity or only contains a single bit transition. A single bit transition within the code automatically results in a code that contains only zeros in one half of the code word and only ones in the other half.

One characteristic of these two alternative code segments is that one is the inverse of the other. Since the disparity of any code segment is +2, 0, or -2, it is obvious that in all the cases where a choice exists between two code segments due to nonneutral disparity, one of these increases the running

disparity by two and the other one reduces it by two. In cases of alternative code segments with neutral disparity, there is the potential of a temporary increase or decrease of the running disparity within the code section by three because all zeros and ones in the code segment are cumulated. If there are alternative code segments, the code segment that keeps the running disparity closer to zero is selected based on the current value of the running disparity. Since 8B/10B coding requires setting the start value for the running disparity to +1 or to -1, the running disparity after each code segment also will be +1 (rd+) or -1 (rd-).

The coding tables for the 5B/6B coding (Table D.1) and the 3B/4B coding (Table D.2) that are the basis for 8B/10B codes show the mapping between an original byte HGFEDCBA to the 5B/6B code abcdei derived from EDCBA and the 3B/4B code fghj derived from HGF. In the header columns of the tables, the current status of the running disparity (rd+ or rd-) is listed to allow the selection of the required code segment for code words that offer this alternative. The column rd' in the tables indicates the status of the running disparity after the code segment is applied. The entry -rd in this column means that the running disparity will be reversed by the applied code segment, and the entry rd means that the running disparity will keep its state.

Table D.2 specifies a special handling for the coding of Dxx.7 symbols. There is a primary coding (Dxx.P7) and an alternative coding (Dxx.A7) for the 3B/4B code segment. The two alternatives are required to avoid a runlength of five in combination with the previous 5B/6B code segment for the data symbols. Thus, the alternative Dxx.A7 code is used for D17, D18, and D20 if the running disparity is rd- and for D11, D13, and D14 if the running disparity is rd+.

Example Let's assume we want to transmit the data bytes 0xAA and 0x17 in this sequence. First we separate these bytes into the 5B and 3B components that will then be mapped to their 5B/6B and 3B/4B codes. Since the bytes follow the bit sequence HGFEDCBA, the lower 5 bits EDCBA will be the source for the 5B/6B code section and the upper 3 bits HGF are the source for the 3B/4B code section. Thus, byte 0xAA = 10101010 is separated into the 5B section 01010 = 10 and the 3B section 101 = 5. Byte 0x17 = 00010111 is separated into 10111 = 23 and 000 = 0. With the binary representation of this separation, 0xAA will be coded by D10.5 and 0x17 will be coded by D23.0. Using Tables D.1 and D.2 these codes are translated into their 5B/6B and 3B/4B codes. Since the coding can vary depending on the running disparity, a start value for the running disparity has to be assumed. With a negative running disparity (rd-) as starting disparity, D10.y is coded to 010101 and the running disparity keeps its state. The code for Dxx.5 is 1010 independently

Table D.1
5B/6B Data Character Coding

Inputs			abcdei outputs		rd'	Inputs			abcdei outputs		rd'
Dx	EDCBA	rd+	rd-	rd'		Dx	EDCBA	rd+	rd-	rd'	
D0	00000	011000	100111	-rd		D16	10000	100100	011011	-rd	
D1	00001	100010	011101	-rd		D17	10001		100011	rd	
D2	00010	010010	101101	-rd		D18	10010		010011	rd	
D3	00011		110001	rd		D19	10011		110010	rd	
D4	00100	001010	110101	-rd		D20	10100		001011	rd	
D5	00101		101001	rd		D21	10101		101010	rd	
D6	00110		011001	rd		D22	10110		011010	rd	
D7	00111	000111	111000	rd		D23	10111	000101	111010	-rd	
D8	01000	000110	111001	-rd		D24	11000	001100	110011	-rd	
D9	01001		100101	rd		D25	11001		100110	rd	
D10	01010		010101	rd		D26	11010		010110	rd	
D11	01011		110100	rd		D27	11011	001001	110110	-rd	
D12	01100		001101	rd		D28	11100		001110	rd	
D13	01101		101100	rd		D29	11101	010001	101110	-rd	
D14	01110		011100	rd		D30	11110	100001	011110	-rd	
D15	01111	101000	010111	-rd		D31	11111	010100	101011	-rd	

Table D.2
3B/4B Data Character Coding

Inputs			fghj outputs		rd'
Dx	HGF	rd+	rd-	rd'	
Dxx.0	000	0100	1011	-rd	
Dxx.1	001		1001	rd	
Dxx.2	010		0101	rd	
Dxx.3	011	0011	1100	rd	
Dxx.4	100	0010	1101	-rd	
Dxx.5	101		1010	rd	
Dxx.6	110		0110	rd	
Dxx.P7	111	0001	1110	-rd	
Dxx.A7	111	1000	0111	-rd	

of the running disparity and the disparity is kept. Thus, D10.5 with a starting negative running disparity is coded to 0101011010 and the running disparity is kept negative. This negative running disparity selects 111010 as coding for D23.y. The code for D23.y reverses the running disparity from negative to positive. Thus, Dxx.0 is coded with the rd+ representative to 0100 and the running disparity again is reversed from positive to negative. With this, the overall 10B code for D23.0 in our case is 1110100100 and the running disparity overall stays negative.

As mentioned previously, the 8B/10B code space contains distinct control characters Kxx.y that follow a different code mapping than the data characters. Overall, the 8B/10B coding scheme defines 12 control characters K28.0–K28.7, K23.7, K27.7, K29.7, and K30.7. Three of these special symbols (K28.1, K28.5, and K28.7) have a runlength of five. This unique property makes them suitable for synchronization to the data stream because they can be identified easily. Characters that allow such an easy identification and synchronization are also called comma characters or comma symbols.

The use of the K28.7 comma character has to follow some special rules in order to avoid ambiguities during comma character identification. For example, it can only be transmitted isolated from other K28.7 characters and there are restrictions in regard to data character combinations because these could create comma characters consisting of two adjacent sections of two characters. For completeness, Tables D.3 and D.4 show the code mapping for all control characters defined in the 5B/6B and 3B/4B coding schemes.

Table D.3
5B/6B Command Character Coding

Inputs		abcdei outputs		rd'
Kx	EDCBA	rd+	rd-	rd'
K23	10111	000101	111010	-rd
K27	11011	001001	110110	-rd
K29	11101	010001	101110	-rd
K30	11110	100001	011110	-rd
K28	11100	001111	110000	-rd

D.3 Scrambling

Scrambling is a methodology to create a random distribution of zeros and ones in the transmitted data stream. When scrambling is applied, the transmitter performs a scrambling operation before the data is transmitted and the receiver

Table D.4
3B/4B Command Character Coding

Inputs		fghj outputs		rd'
Dx	HGF	rd+	rd-	rd'
Kxx.0	000	0100	1011	-rd
Kxx.1	001	1001	0110	rd
Kxx.2	010	0101	1010	rd
Kxx.3	011	0011	1100	rd
Kxx.4	100	0010	1101	-rd
Kxx.5	101	1010	0101	rd
Kxx.6	110	0110	1001	rd
Kxx.7	111	1000	0111	-rd

reverses this operation in a descrambler before the data is handed over to the following operational device blocks for further processing. The purpose of such a randomization of the data stream is to reduce the likelihood for long runlengths to occur and to reduce repetitions of data sequences within the transmitted data stream.

With scrambling, average runlength reduction can be achieved without generating data traffic overhead as it is the case with 8B/10B coding, for example, where a byte that is to be transmitted needs to be expanded to a 10-bit word causing a data transmission overhead of 25%. Scrambling however has other challenges (for more detailed information, refer to [2]).

The reduction of data repetitions due to scrambling has the desired effect to spread out the spectrum of the energy transmitted with a data stream and thus reduces the radiated EMI energy at single distinct frequencies. One extreme example for this would be the unscrambled transmission of a data pattern that is a bit clock. This would create a peak in the frequency spectrum of the transmitted data at the frequency representing the data rate of the transmitted data. With scrambling of the clock pattern such a peak is reduced and the energy originally contained in the peak is transferred to other frequency bins as shown in Figures D.2 and D.3.

The functionality of scramblers and descramblers can be modeled best by a linear feedback shift register (LFSR) as shown in Figure D.4 for a very simple scrambling/descrambling operation. It has to be noted that this block diagram is very basic and in a real system additional blocks are needed to avoid desynchronization.

The similarities of the scrambler/descrambler structure to an LFSR-based PRBS generator are obvious and should be no surprise to the reader since both circuits have the goal to generate random data. The only difference

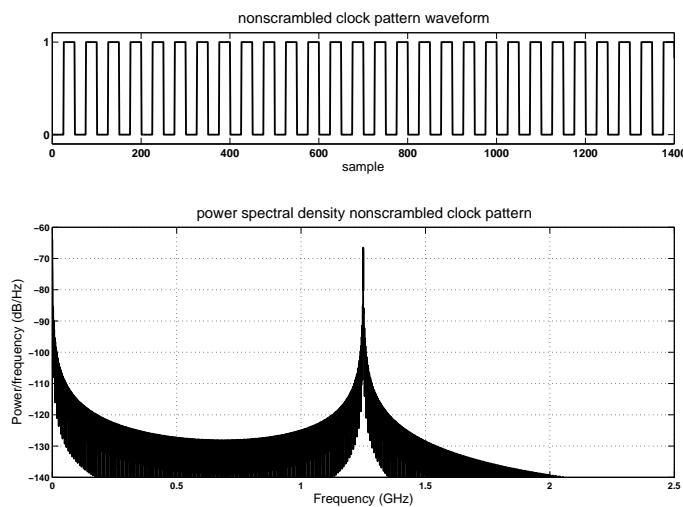


Figure D.2 Nonscrambled bit clock pattern waveform and associated power density spectrum.

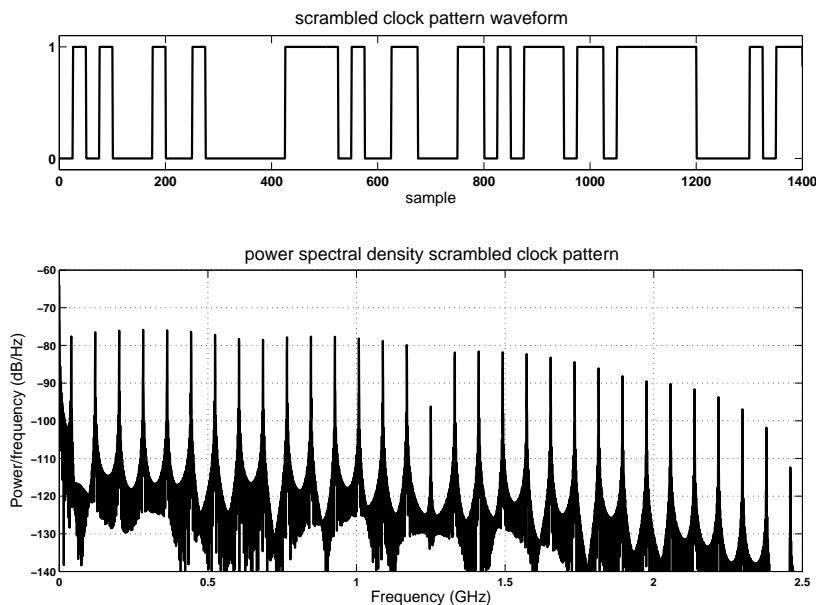


Figure D.3 Scrambled bit clock pattern waveform and associated power density spectrum.

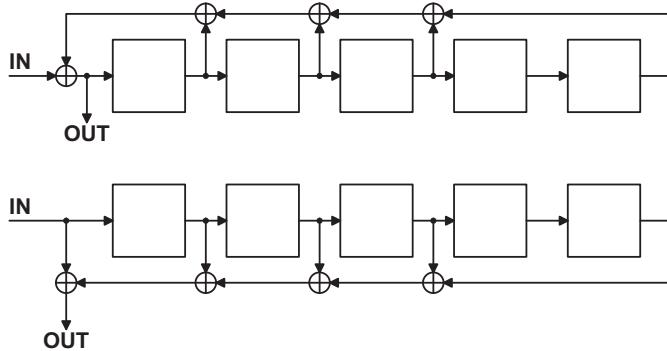


Figure D.4 Basic block diagram example of a scrambler (top) and descrambler (bottom) implementation.

is that for the scrambler/descrambler, the generated data depends on data fed into the LFSR and the seed of the LFSR, while for a PRBS generator, the output data only depends on the seed.

Real scrambler/descrambler implementations usually do not use LFSRs since they would have to run at the speed of the transmitted data which is a challenge for high-speed interfaces. Instead, the LFSR functionality is transferred into a parallel architecture consisting of a combinatorial network of exclusive OR gates (XOR, operator symbol \oplus) [3].

D.4 Error Detection

With increasing data rates and shrinking parameter margins for the transmission of high-speed signals, methodologies to detect errors that occur during the data transmission start to gain traction in some of the high-speed I/O standards. For high-speed I/O interfaces that are designed for in-system or local communication, deploying error detection usually is sufficient because retransmission of corrupted data is affordable. For long haul communication standards, error correction schemes are applied because the latencies caused by retransmissions are not tolerable. In this appendix we will restrict the discussion to error detection methodologies that are required to get a better understanding of the applications discussed in this book. If the reader is interested in more detailed information on error correction it can be found in [4], for example.

D.4.1 Parity Bits

The most simple form of an error detection mechanism probably is the use of parity bits that are attached to each entity of bits they are based on and are then transmitted together with this entity. The calculation of a parity bit is simply a bitwise recursive XOR operation on a set of bits of the data stream (e.g., one data word, a byte, and so forth) as shown in Figure D.5. Since the XOR function represents a modulo-2 operation on binary data, the result of this operation indicates whether the analyzed set of bits contains an even or odd number of ones. If the receiver of the data also builds a parity bit over the same set of bits including the received parity bit itself, it can detect an error if the result of that operation is not zero. With this parity bit based error detection all errors that affect an odd number of bits can be detected and data retransmission can be requested. It is obvious that the error detection rate of parity bit based error detection is limited and that it does not work if an even number of bits are affected by errors. While this is sufficient for applications where single bit errors are expected as the main failure type (like in early times on PC main memory components), it is not necessarily suited for state-of-the-art high-speed I/O interfaces where distortions often affect multiple bits due to the short bit times used. Thus, usually more sophisticated checksum-based error detection mechanisms are applied for this type of interface.

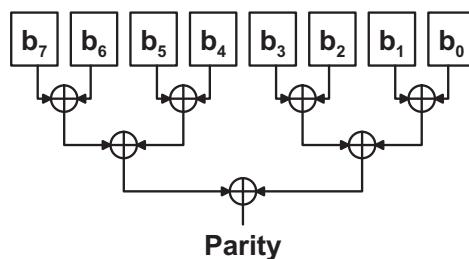


Figure D.5 Parity bit generation with recursive XOR gates.

D.4.2 Checksums

A logical step for a checksum based error detection scheme is to extend the parity bit methodology to a parity word. An example of such an approach is the bit interleaved parity N (BIP-N) error detection scheme [5]. Here, a section (e.g., one packet) of the data stream is split into N-bit words and the parity bit calculation is done over corresponding bits within these words for all data words. This operation will result in a N-bit parity checksum.

Of course, the BIP-N error detection scheme has the same limitations as the standard parity bit approach regarding an even number of errors on the bits forming the basis for one parity bit within the parity checksum. However, the advantage of building one parity bit over several words instead of one parity bit for each word is that the bits that are used for the parity calculation are timely separated from each other. Thus the likelihood that distortions during data transmission affect multiple bits that are used for the calculation of a single parity bit is lower with the BIP-N approach than with the classical parity bit error detection.

Besides the nonsensitivity for an even number of errors, parity based approaches also have the issue that they are not sensitive to changes in the sequence of the data they calculate the parity on. Thus, checksum error detection methodologies that are used in real applications predominantly make use of position dependent checksum approaches that do not have the disadvantages of the previously discussed parity based algorithms. While there are quite a few possible implementations of position dependent error detection methodologies, we want to discuss the most prominent representative of this group, which is the cyclic redundancy check (CRC) [6–8].

The CRC calculates a checksum based on a polynomial division on subsets of the data to be transmitted by a generator polynomial. The remainder of this division serves as checksum that is transmitted together with the data subsets. The receiver also calculates the division remainder between received data and generator polynomial and compares its local remainder with the received CRC checksum. If these are not the same an error is present on the received data or checksum. The identification of a transmission error by the receiver can be simplified for an n -bit CRC checksum by appending n zeros to the original dividend before the polynomial division by the sender. In this case, the receiver can append the received checksum to the received data before doing its division. If the remainder of the polynomial division in the receiver is not zero, there was a data transmission error.

The selection of the generator polynomial is a critical step for CRC error detection because the polynomial defines what types of errors can be detected. Applications that use CRC checksums usually have polynomials defined in their standards that are selected to best fit to the most likely errors that can occur in the respective environment of that application. A by far not complete selection of CRC generator polynomials is shown in Table D.5. As a side note on this table, we want to highlight that the previously discussed parity bit error detection approach also can be implemented as a CRC checksum of the length one using the polynomial $x + 1$.

In order to explain how a polynomial division can be done easily in a real

Table D.5
Selection of CRC Generator Polynomials

Name	Polynomial	Application
CRC-1	$x + 1$	Parity bit
CRC-5-USB	$x^5 + x^2 + 1$	USB
CRC-8-ATM	$x^8 + x^2 + x + 1$	ATM HEC, GDDR5
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$	CDMA, Bluetooth, IrDA, MMC, SD

implementation, we represent both the payload data as well as the generator polynomial in a binary form. The binary representation of a polynomial is done in a way that the single bits of the binary data form the coefficients of the polynomial according to the following example:

$$\begin{aligned} M &= 10100101 \\ \Leftrightarrow M &= 1x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 0x^1 + 1x^0 \\ &= x^7 + x^5 + x^2 + 1 \end{aligned}$$

The division of the two numbers then works exactly as a polynomial division. An important fact to notice is that the modulo-2 subtraction we have to apply during the division process is a simple bitwise exclusive OR operation. An example for such a division is shown in Figure D.6. Since we are not interested in the division result, but only the remainder of the division, we do not keep track of the actual result in this example.

From this example it becomes obvious that the polynomial division just consists of XOR operations with shifts of the generator polynomial versus the data polynomial. From a hardware point of view this functionality can be implemented using a linear feedback shift register with a seed value of zero. This linear feedback shift register represents the CRC generator polynomial which is fed serially by the data the CRC checksum is to be built for. After all data bits are consumed, the shift register contains the CRC checksum. The LFSR representation of the CRC generator polynomial places its XOR gates after the CRC bits that have a coefficient of one in the generator polynomial as shown for the CRC-8-ATM generator polynomial in Figure D.7.

As the reader can imagine, the LFSR implementation to calculate CRC checksums in hardware raises some significant challenges for high-speed interfaces due to the raw speed the shift operations and XOR operations between shift stages have to operate at. If the LFSR operation is analyzed in detail, it becomes obvious that the CRC checksum is generated by repeated XOR operations on the data bits and the seed initially present in the shift

generator polynomial : $x^8+x^2+1 \Leftrightarrow M=1001011$
 data : 1111011010110011



Figure D.6 Polynomial division example.

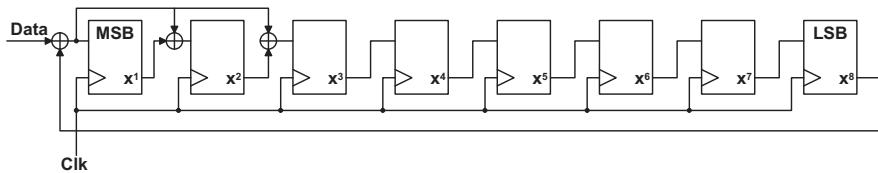


Figure D.7 LFSR representation of CRC-8-ATM polynomial.

register. Thus, the functionality of the LFSR can be mapped to a combinatorial XOR network to which the bits of the data word can be applied in parallel. Since the seed of the LFSR is set to zero and the rule $A \oplus 0 = A$ applies, the seed values do not have to be considered as inputs for the combinatorial network. A methodology to transfer the LFSR representation of a CRC generator polynomial to a combinatorial XOR network is given in [9]. The same transformation from an LFSR representation to pure XOR terms also can be applied for scrambling LFSRs.

References

- [1] A. Widmer and P. Franaszek, "A DC-Balanced, Partitioned-Block, 8b/10B Transition Code," *IBM Journal of Research and Development*, vol. 17, Sept. 1983.
- [2] Y. Takasaki, *Digital Transmission Design and Jitter Analysis*. Artech House, 1991.
- [3] D. R. Stauffer, J. T. Mechler, M. A. Sorna, K. Dramstad, C. R. Ogilvie, A. Mohammad, and J. D. Rockrohr, *High Speed Serdes Devices and Applications*. Springer, 2008.

- [4] H. S. Warren, “Hacker’s Delight – Error Correcting Codes,” 2003. <http://www.hackersdelight.org/ecc.pdf>.
- [5] American National Standards Institute, *American National Standard for Telecommunications - Digital Hierarchy Optical Interface Rates and Formats Specification*, 1988. ANSI T1.105-1988.
- [6] W. W. Peterson and D. T. Brown, “Cyclic Codes for Error Detection,” *Proceedings of the IRE*, Jan. 1961.
- [7] A. S. Tanenbaum, *Computer Networks*. Simon & Schuster, 1983.
- [8] H. S. Warren, “Hacker’s Delight – Cyclic Redundancy Check,” 2003. <http://www.hackersdelight.org/crc.pdf>.
- [9] Cypress Semiconductor Corporation, *Parallel Cyclic Redundancy Check (CRC) for HOTLinkTM*, Mar. 1999. AN1089.

E

Time Domain Reflectometry and Time Domain Transmission (TDR/TDT)¹

This appendix presents some important concepts regarding time domain reflectometry (TDR) and time domain transmission (TDT).

TDR/TDT is a tool that has been critical in electrical engineering from its beginning and continues to be of significant importance today [1]. Introductory references on TDR/TDT for electrical characterization are [2–7]. The evaluation of a printed circuit board (PCB) test fixture benefits from a number of TDR and TDT measurements to validate and determine performance with the following being the most common:

- Measuring the characteristic impedance of a PCB signal trace (TDR);
- Measuring and identifying discontinuities on a signal path (TDR);
- Measuring the step response of a signal path (TDT);
- Obtaining a model of a transmission line (TDR/TDT);
- Determining the location and magnitude of near-end and far-end (NEXT/FEXT) crosstalk (TDT).

Figure E.1 shows a very simple block diagram of a TDR/TDT instrument. The TDR/TDT timing resolution depends mainly on the step generator rise time and the sampler bandwidth. The TDR/TDT information can also be obtained with the conversion to the time domain of the S-parameters measured with a vector network analyzer (VNA). A typical TDR instrument has the step generator and sampler connected together inside the instrument for ease of use, however, a high performance instrument with very

¹In collaboration with Heidi Barnes.

fast rise times may require separate specialized modules as shown in Figure E.2 [1].

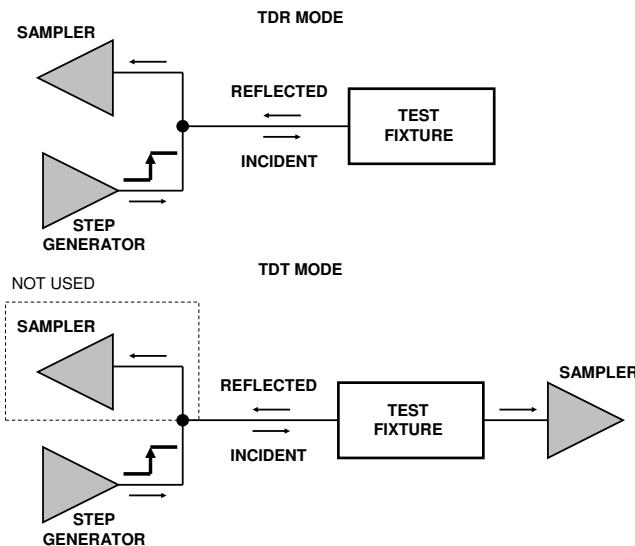


Figure E.1 Block diagram of a TDR/TDT instrument.

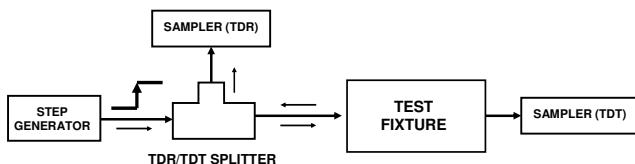


Figure E.2 Block diagram of a TDR/TDT instrument with independent step generator and sampler modules.

E.1 TDR

When performing a TDR measurement, a step waveform is sent into the transmission line path of interest, such as a test fixture. This step edge will travel the length of the path and reflect at the end if there is an open (e.g., at the empty DUT socket) or a short (DUT pins including ground pins shorted together), and return to the beginning of the path where the TDR sampler can then measure the reflected waveform. Reflections will also occur at any changes in impedance (e.g., a change on the PCB trace width or a discontinuity like a via) as shown in Figure E.3.

The polarity of the reflection provides additional information on the type of discontinuity that was encountered. Higher series inductance than the characteristic impedance or an open provides a positive voltage reflection. A parallel capacitance or a short provides a negative voltage reflection as shown in Figure E.3. The equivalent excess inductance or capacitance of a discontinuity can be calculated from the step response by integrating the difference between the reflected response waveform and a reference waveform without the excess inductance or capacitance using (E.2) and (E.3). This approach is more accurate than comparing peak reflections [8]. This is not to be confused with the transmission line equation used for calculating total capacitance or inductance of a signal path as shown in (E.4) and (E.5).

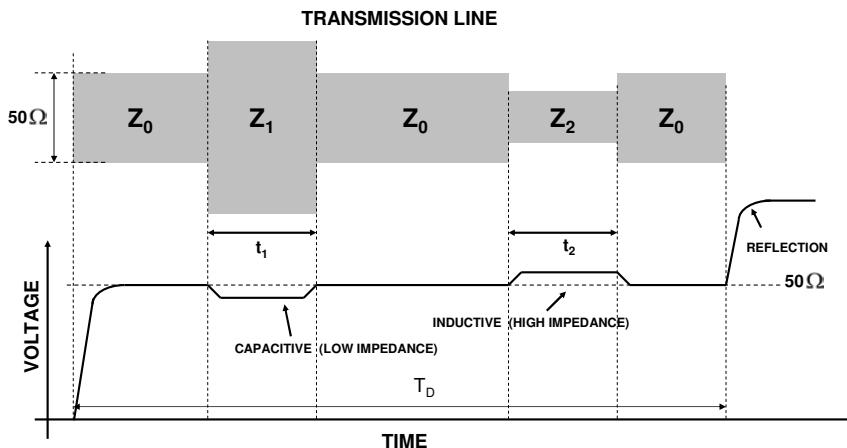


Figure E.3 Diagram showing the effect of discontinuities on the transmission line on the response to a step waveform (TDR).

E.1.1 Measuring the Impedance of a Trace with a TDR

One of the first measurements that should be done on the high-speed signal paths of a PCB test fixture is to verify the signal trace impedance with TDR. It is important with this measurement to use the built-in calibration support of the TDR instrument to set the 50Ω reference of the TDR measurement. Figure E.4 presents an example of an impedance measurement with a TDR. Transmission line losses and discontinuity reflections will mask the true impedance of the transmission line the further the TDR travels into a structure and a “peeling algorithm” may be required to analyze the impedance as the TDR reflection gets further from the input port [9].

Computing the Excess Capacitance or Inductance

Define τ to be the integral of the portion of the TDR reflection that contains the discontinuity of interest, and $\rho(t)$ is the reflected TDR voltage waveform:

$$\tau = \int_{t_0}^{t_1} \rho(t) dt \quad (\text{E.1})$$

If $\tau > 0$ there is excess L :

$$L = 2Z_o\tau \frac{(Z_0 + Z_N)^2}{4Z_0Z_N} \quad (\text{E.2})$$

If $\tau < 0$ there is excess C :

$$C = -\frac{2\tau}{Z_0} \frac{(Z_0 + Z_N)^2}{4Z_0Z_N} \quad (\text{E.3})$$

Z_0 is the impedance of the non-50 Ω system where the integral starts and is used to normalize the integral to the height of the step incident on the discontinuity. Z_N is the nominal impedance of the TDR system (usually 50 Ω).

Computing the Total Capacitance and Inductance

The total capacitance and inductance for an ideal (lossless) transmission line can be found from the TDR measured delay and impedance (Z) using the following equations:

$$C(\text{Total}) = \frac{t(\text{delay})}{Z} = \frac{t(\text{TDR round-trip delay})}{2} \frac{1}{Z} \quad (\text{E.4})$$

$$L(\text{Total}) = t(\text{delay}) Z = \frac{t(\text{TDR round-trip delay})}{2} Z \quad (\text{E.5})$$

E.1.2 Measuring the Round-Trip Delay of a Signal Trace

Another important measurement that can be done with a TDR on a test fixture is to determine the round-trip delay between the DUT socket and the test fixture pogo via connection. This measurement is especially important when evaluating the phase matching or time skew between different signal traces. Figure E.5 presents an example of a round-trip delay measurement with a

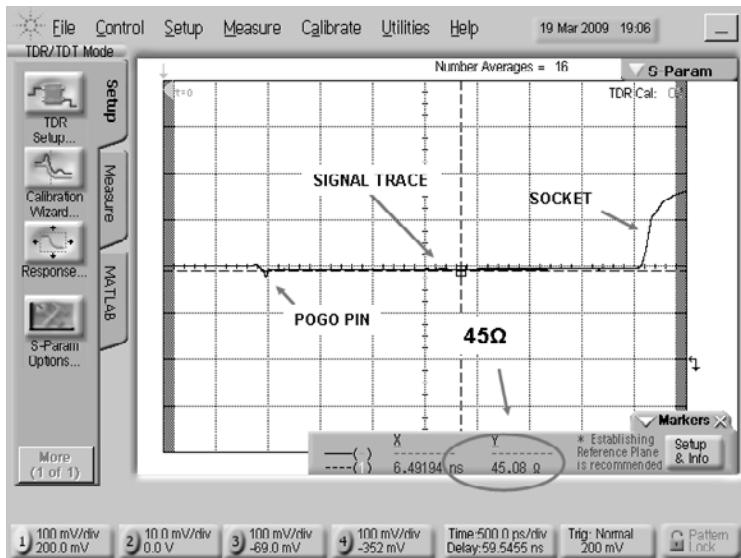


Figure E.4 Example of an impedance measurement on an ATE test fixture using a TDR.

TDR. It is important to remember that the TDR will always display the round-trip delay. A simple approximation for the signal path delay is to divide the round-trip delay by two. This approximation is only exact for ideal (lossless) signal paths [10].

E.1.3 Measuring Discontinuities on a Signal Path with a TDR

A TDR is extremely useful in evaluating the discontinuities on a test fixture and is one of the few methods that enables one to identify locations for signal integrity improvement. These discontinuities can be interlayer vias, relays, SMT footprints, and so on. The ability to see the details of a discontinuity is directly related to the rise time of the pulse generator with faster rise times providing finer time resolution and enabling the separation and measurement of closely spaced impedance discontinuities. Figure E.6 shows an example of measuring test fixture discontinuities using a TDR.

E.1.4 Measuring the Return Loss with a TDR

With the appropriate calibration techniques and software it is possible to obtain a frequency domain return loss measurement (S_{11}) through a TDR measurement [11]. The magnitude of the spectral content of a TDR step falls

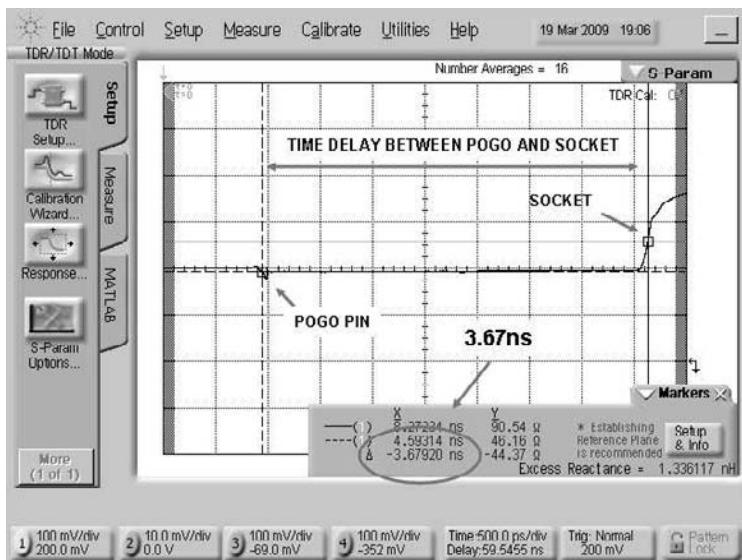


Figure E.5 Example of a round-trip delay measurement on an ATE test fixture with a TDR.

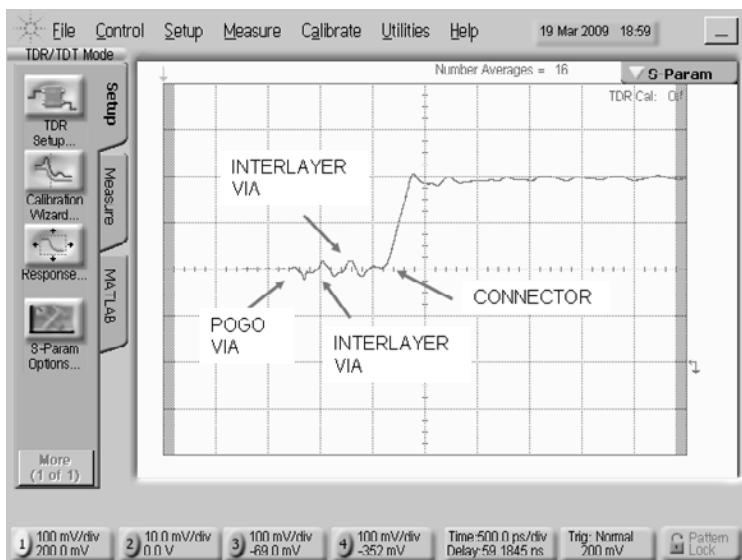


Figure E.6 Example of discontinuities measurements using a TDR.

off as $1/f$ so these types of measurements often require a significant amount of averaging to improve the S/N ratio at the higher frequencies. Most TDR instruments have this algorithm integrated into the available software tools. References [6, 12] provide detailed discussions of the differences between a TDR and a VNA approach for signal path characterization measurements.

E.2 TDT

The TDR provides valuable information on the test fixture, but what really matters for signal integrity is the signal that is received or transmitted by the DUT, which can be investigated with a TDT measurement. When performing a TDT measurement, a step waveform is sent through the test fixture signal trace and is measured at the other end of the signal trace with a sampler. The loss of the signal trace, impedance changes, and discontinuities will have an effect on the shape of the step after traveling through the signal trace as shown in Figure E.7. Note that unlike the TDR, for a TDT measurement it is necessary to have access to both ends of the PCB signal trace being measured, which is not always easy in the case of a test fixture. See Appendix H for an in-depth discussion of this topic.



Figure E.7 Diagram showing the effect of discontinuities on the transmission line on the response to a pulse waveform (TDT).

E.2.1 Measuring the Step Response

The key result from a TDT measurement is the measured step response of the test fixture signal path. If one assumes the signal trace transmission line to be a time invariant linear system, then the system response to a known stimulus step will provide a complete description of the test fixture signal path performance [13]. Figure E.8 shows an example of the measured step response for a test fixture signal trace.

The rise time of the measured step response is related to the bandwidth of the test fixture signal trace being measured. Short rise times (faster slew rates) contain higher frequencies and a wider bandwidth. Accurate measurement of the step response relies on the intrinsic rise time of the TDR module step being much smaller than the expected rise time at the end of the signal. This ensures

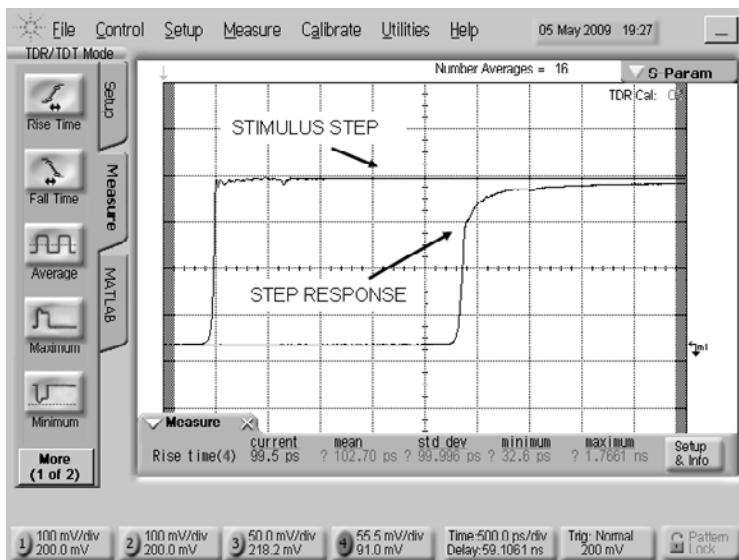


Figure E.8 Measuring the step response of a test fixture transmission line.

that the measured rise time value is dominated by the bandwidth limitations of the signal trace under measurement and not the input step.

It is also possible to take the step response and use it to obtain a “synthesized” data eye diagram as shown in Figure E.9 [14]; it is synthesized because it is not directly measured but instead it is generated from the measured step response or frequency domain S-parameters.

E.2.2 Measuring the Insertion Loss with a TDT

Similar to obtaining frequency dependent reflection information (S_{11}) from the TDR measurement, it is also possible to obtain the insertion loss measurement (S_{21}) through a TDT measurement with the appropriate calibration techniques and software [11]. Observing the performance in the frequency domain can provide insight into signal path resonances and bandwidth limitations.

E.2.3 Measuring Crosstalk Using a TDT and an Extra Sampler

Another interesting use for TDT measurements is the ability to measure aggressor to victim crosstalk. Figure E.10 shows a block diagram of a measurement scenario that would require only an extra sampler to measure

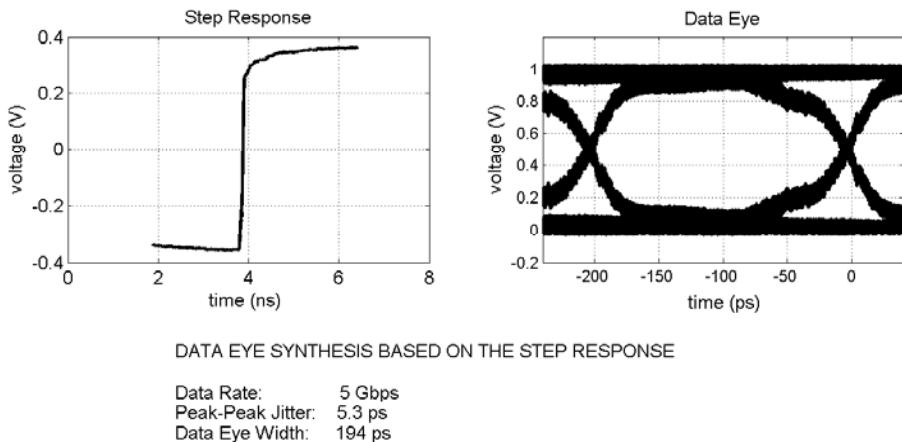


Figure E.9 Example of a synthesized data eye diagram from the step response of a test fixture measured with the ATE pin electronics (courtesy of Verigy).

both the far-end crosstalk (FEXT) and the near-end crosstalk (NEXT) on a victim trace from an adjacent aggressor trace [2, 15].

Figure E.11 shows an example of a crosstalk measurement of a pogo via on an ATE system using the setup described in Figure E.10 [16].

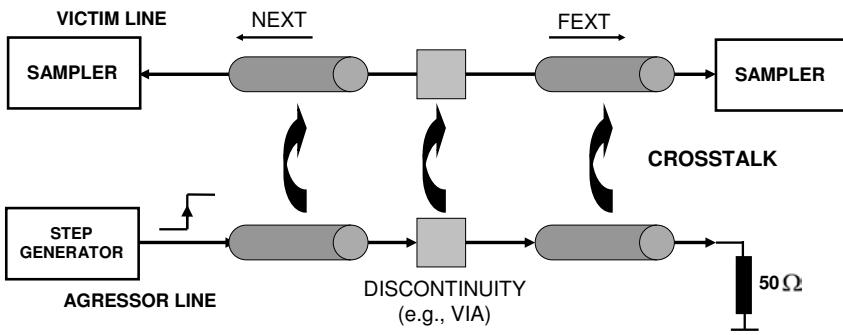


Figure E.10 Diagram showing a crosstalk measurement using a TDT signal source and two extra samplers.

E.3 Differential TDR/TDT Measurements

As the name implies a differential TDR or TDT measurement requires the measurement of the difference of two signals and can be obtained by using two step sources to stimulate a given differential structure to determine its

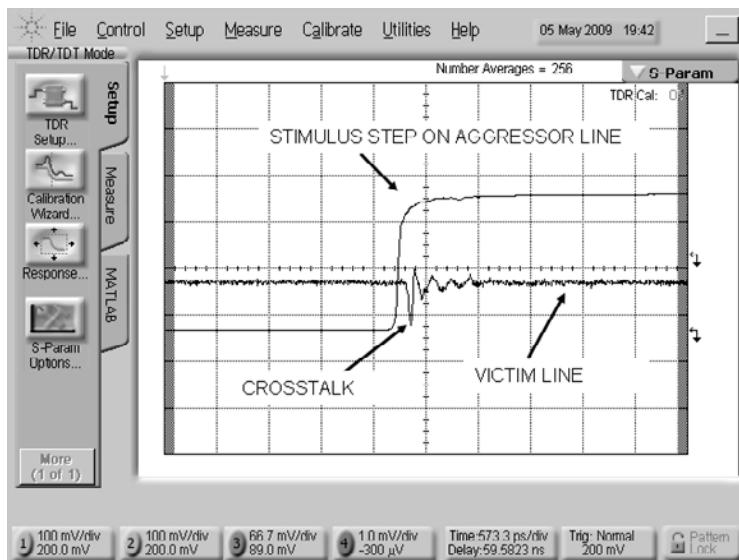


Figure E.11 Example of measuring the crosstalk between two adjacent pogo vias using a TDT instrument.

differential response. The differential stimulus is important for accurately capturing the effects of coupling throughout the signal path and the effects of common mode to differential and differential to common mode conversions. The TDR/TDT instrument as shown in Figure E.12 can drive a pair of traces in the even mode with a common signal that has the same polarity step on each trace or in the odd mode with a differential signal that has opposite polarity steps on each trace. This is needed for obtaining both the common and differential modes of the step response [6, 17].

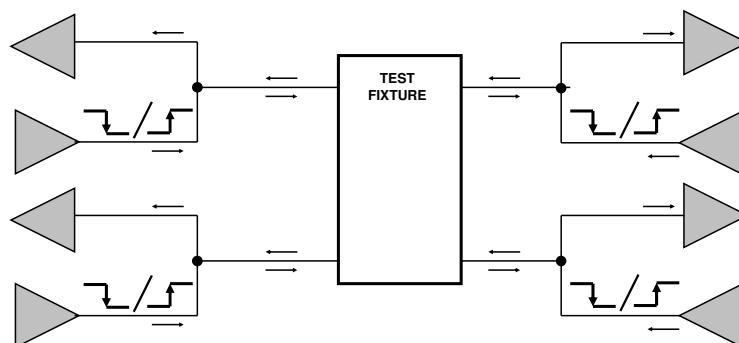


Figure E.12 A differential TDR/TDT measurement setup.

Like a four-port S-Parameter measurement (see Appendix F), a differential TDR/TDT measurement is also a four-port measurement that will result in 16 separate measurements [3] as shown in Figure E.13. Measurements in the frequency domain with a VNA can also provide differential TDR and TDT information with the use of a multiport network analyzer.

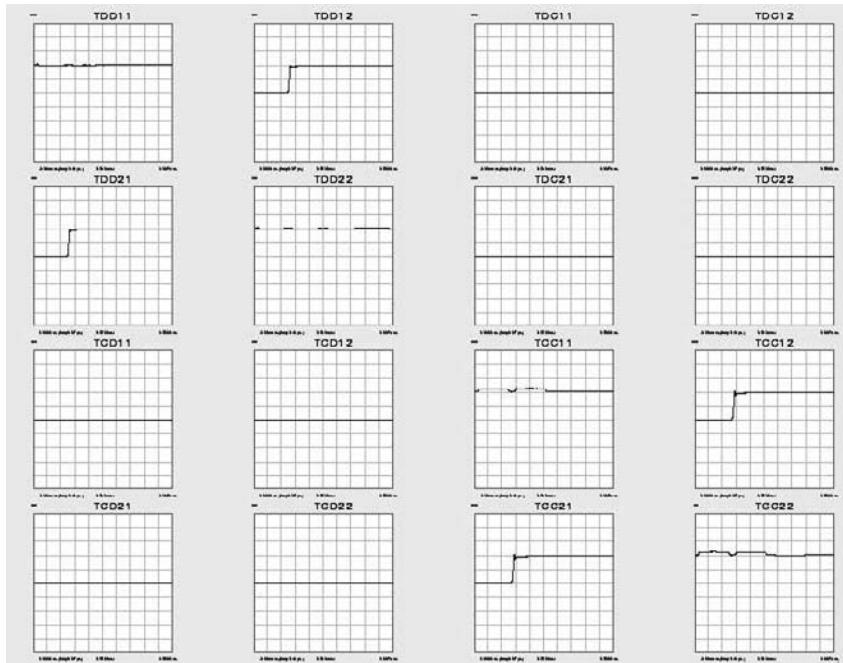


Figure E.13 Example of a four-port TDR/TDT measurement on a coupled differential line.

References

- [1] D.-H. Han, B.-S. Xu, M. J. Choi, J. He, S. Gardiner, and C. Lee, “Realization of Ultra-Wideband High-Resolution TDR for Chip-Carrier Packages,” *Proceedings of the ASME InterPACK*, 2005.
- [2] Agilent Technologies, “Agilent Signal Integrity Analysis Series, Part 1: Single-Port TDR, TDR/TDT, and 2-Port TDR,” *Application Note 5989-5763EN*, 2007.
- [3] Agilent Technologies, “Agilent Signal Integrity Analysis Series, Part 2: 4-Port TDR/VNA/PLTS,” *Application Note 5989-5764EN*, 2007.
- [4] Agilent Technologies, “Time Domain Reflectometry Theory,” *Application Note 1304-2*.

- [5] D. I. Geoff Lawday and G. Edlund, *A Signal Integrity Engineer's Companion*. Prentice-Hall, 2008.
- [6] M. Resso and E. Bogatin, *Signal Integrity Characterization Techniques*. IEC, 2009.
- [7] J. R. Andrews, "Time Domain Reflectometry (TDR) and Time Domain Transmission (TDT) Measurement Fundamentals," *Picosecond Labs Application Note AN-15*, 2004.
- [8] D. J. Dascher, "Measuring Parasitic Capacitance and Inductance Using TDR," *Hewlett-Packard Journal*, 1996.
- [9] D. Miller, *Designing High-Speed Interconnect Circuits*. Intel Press, 2004.
- [10] W. Maichen, *Digital Timing Measurements*. Springer, 2006.
- [11] J. R. Andrews, "Time Domain Spectrum Analyzer and 'S' Parameter Vector Network Analyzer," *Picosecond Labs Application Note AN-16a*, 2004.
- [12] Agilent Technologies, "Limitations and Accuracies of Time and Frequency Analysis of Physical Layer Devices," *White Paper 5989-2421*, 2005.
- [13] J. R. Andrews, "Deconvolution of System Impulse Responses and Time Domain Waveforms," *Picosecond Labs Application Note AN-18*, 2004.
- [14] J. Ren and K. S. Oh, "Multiple Edge Responses for Fast and Accurate System Simulations," *IEEE Transactions on Advanced Packaging*, vol. 31, Nov. 2008.
- [15] Tektronix, "Time Domain Methods for Measuring Crosstalk for PCB Quality Verification," *Application Note*.
- [16] B. B. Szendrenyi, H. Barnes, J. Moreira, M. Wollitzer, T. Schmid, and M. Tsai, "Addressing the Broadband Crosstalk Challenges of Pogo Pin Type Interfaces for High-Density High-Speed Digital Applications," *International Microwave Symposium*, 2007.
- [17] M. M. McTigue and C. P. Duff, "Differential Time-Domain Reflectometry Module for a Digital Oscilloscope and Communications Analyzer," *Hewlett-Packard Journal*, 1996.

F

S-Parameters¹

Scattering parameters or S-parameters provide a powerful approach to analyze the performance of a test fixture and its individual components. S-parameters work by creating a black-box model that relates the voltage waves incident at each port of the model with the voltage waves that exit from the same ports. This appendix provides an overview of S-parameters and for more detailed information on the definitions and conversions, [1–3] are good starting points for further reading.

For a two-port system like a single-ended transmission line, the S-parameters are described by the following equation:

$$\begin{bmatrix} V_1^-(f) \\ V_2^-(f) \end{bmatrix} = \begin{bmatrix} S_{11}(f) & S_{12}(f) \\ S_{21}(f) & S_{22}(f) \end{bmatrix} \begin{bmatrix} V_1^+(f) \\ V_2^+(f) \end{bmatrix} \quad (\text{F.1})$$

where V_1^+ and V_2^+ are the waves traveling into the two-port network at ports 1 and 2 and V_1^- and V_2^- are the waves traveling out of the two-port network at ports 1 and 2. Note that the S-parameters matrix is defined at multiple frequency values, since in most applications there is a frequency band of interest and not a single value. That is the reason for the term (f) in the matrix.

The S-parameter equation is analyzed in more detail in the flow diagram in Figure F.1. The diagram represents the four different signal paths that a two-port system has and the corresponding S_{ij} terminology used to describe them. The “i” term is the network port being measured and the “j” term is where the input signal is located. The S_{21} insertion gain for a two-port network is the signal measured at port 2 when a voltage wave is input at port 1 and is also known as insertion loss for a passive system. The S_{11} term is the return loss at port 1 which corresponds to the signal reflected at port 1 when a voltage

¹In collaboration with Heidi Barnes.

wave is input at port 1. The S-parameter measurement captures both phase and magnitude of the voltage waves to provide a complete electrical description of this two-port black-box network for the mathematical case of a linear time invariant (LTI) system. This information can then be used to determine the performance of a system such as that shown in Figure F.1 where the ATE pin electronics connects to port 1, the DUT connects to port 2, and the two-port network would be the passive ATE test fixture with its LTI properties.

2-Port S-Parameter Interconnect Model

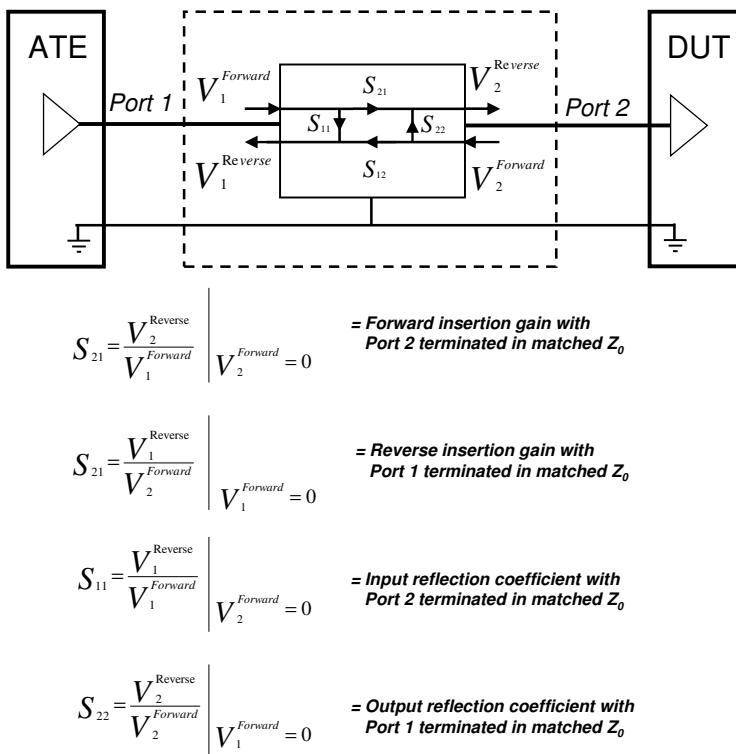


Figure F.1 Block diagram of a two-port configuration that could correspond to a single-ended transmission line. Z_0 is the characteristic impedance of the measurement setup, which in most situations will correspond to 50 Ω .

It is possible to move from the frequency domain representation of the S-parameters to a time domain representation through Fourier transform techniques. Figure F.2 is an example of a two-port S-parameter measurement for a PCB test fixture (see Appendix H on techniques to make this type of measurement). In the figure, the S-parameter data is also transformed to

the time domain for analysis of transition discontinuities, impedance, and step response. Note that the four S-parameter signal paths ($S_{11}, S_{12}, S_{21}, S_{22}$) for the two-port network also have a corresponding time domain path ($T_{11}, T_{12}, T_{21}, T_{22}$). The S-parameter magnitude data is plotted here without the phase; however, the conversion to the time domain is dependent on both the magnitude and phase information.

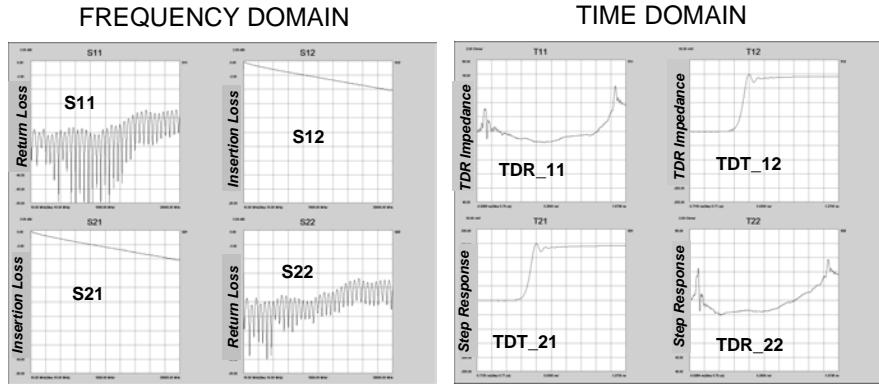


Figure F.2 Example of two-port S-parameter measurement and the corresponding conversion to time domain information.

The simple two-port S-parameter network is a good starting point for measurements and analysis, but it is limited to a single mode of propagation. Most high-speed digital applications utilize differential signaling with multiple modes of propagation. The S-parameter network can easily be expanded to accommodate a four-port network as shown in Figure F.3. In this case there are 16 possible signal paths that show up in the four-port S-parameter matrix that is shown in (F.2). Figure F.4 shows an example of the measured S-parameters for a differential pair in an ATE test fixture. The same matrix would also describe the interaction between a victim and an aggressor signal path including the crosstalk between the two.

$$\begin{bmatrix} V_1^-(f) \\ V_2^-(f) \\ V_3^-(f) \\ V_4^-(f) \end{bmatrix} = \begin{pmatrix} S_{11}(f) & S_{12}(f) & S_{13}(f) & S_{14}(f) \\ S_{21}(f) & S_{22}(f) & S_{23}(f) & S_{24}(f) \\ S_{31}(f) & S_{32}(f) & S_{33}(f) & S_{34}(f) \\ S_{41}(f) & S_{42}(f) & S_{43}(f) & S_{44}(f) \end{pmatrix} \begin{bmatrix} V_1^+(f) \\ V_2^+(f) \\ V_3^+(f) \\ V_4^+(f) \end{bmatrix} \quad (F.2)$$

If one groups the through paths so that they are in reference to each other, then one can rewrite the S-parameter matrix in terms of two-port networks

4 PORT NETWORK

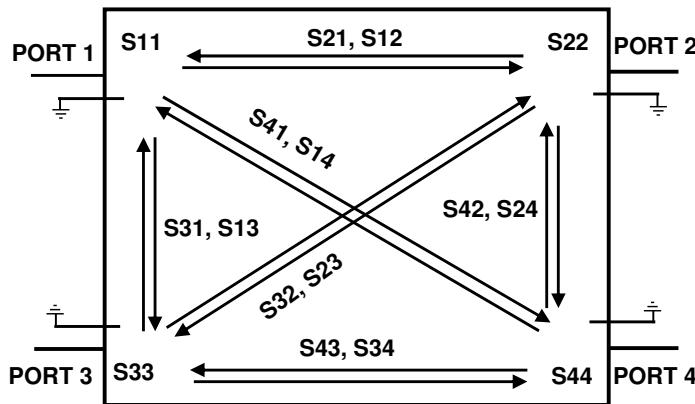


Figure F.3 Block diagram of a four-port system that could correspond to a differential transmission line.

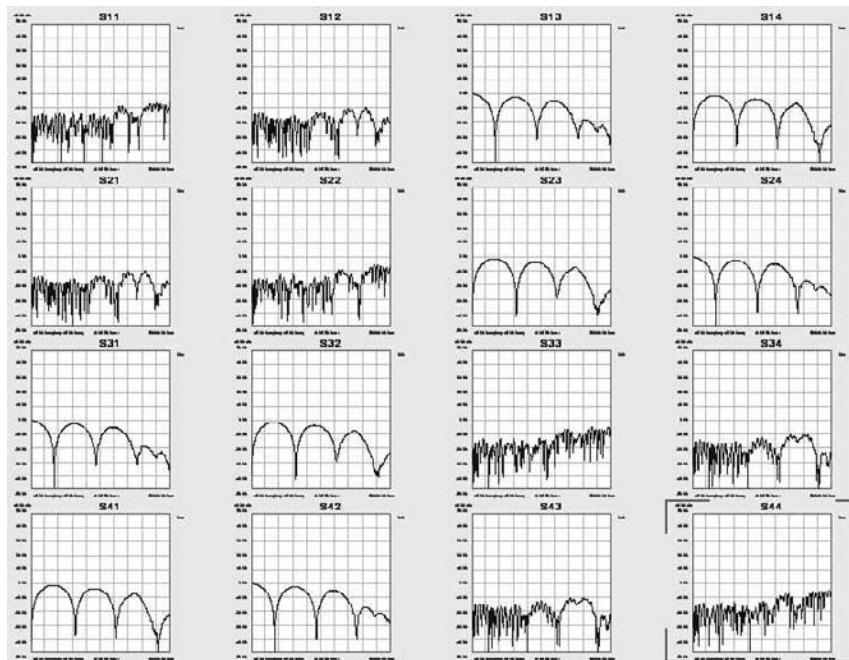


Figure F.4 Measured S-parameters from a four-port system (a differential pair on a test fixture).

representing the differential and common types of excitation for both the through path and the crosstalk paths. In this way the differential S-parameters can be regrouped in a way that allows quick analysis of the signal integrity of a serial I/O link and of any issues with crosstalk or even mode EMI (e.g., S_{13} is the near-end crosstalk between ports 1 and 3 and S_{14} is the far-end crosstalk between ports 1 and 4).

This mixed mode S-parameter data set is now in reference to the excitation as well as the ports and utilizes a new set of descriptors. The signal paths can be excited or measured as a differential signal (D) or a common signal (C). It is possible to identify on the four-port S-parameter matrix (F.3) the submatrices that correspond to the differential response to the differential signal (S_{DD}), the common mode response to a differential signal (S_{CD}), the differential response to the common mode signal (S_{DC}), and finally the common mode response to a common mode signal (S_{CC}). The port 1' notation refers to the signal across the single mode ports 1 and 3, and likewise port 2' refers to the signal across the single mode ports 2 and 4 using the convention that the through path always connects 1 to 2, 3 to 4, and so on, for commonality in port names as the number of ports is increased. Again, the output port is listed first and then the input port. We can then write the S-parameter matrix for the four-port system in the following form:

$$\begin{bmatrix} V_{D1'} \\ V_{D2'} \\ V_{C1'} \\ V_{C2'} \end{bmatrix} = \begin{pmatrix} S_{DD1'1'} & S_{DD1'2'} & S_{DC1'1'} & S_{DC1'2'} \\ S_{DD2'1'} & S_{DD2'2'} & S_{DC2'1'} & S_{DC2'2'} \\ S_{CD1'1'} & S_{CD1'2'} & S_{CC1'1'} & S_{CC1'2'} \\ S_{CD2'1'} & S_{CD2'2'} & S_{CC2'1'} & S_{CC2'2'} \end{pmatrix} \begin{bmatrix} V_{D1'} \\ V_{D2'} \\ V_{C1'} \\ V_{C2'} \end{bmatrix} \quad (\text{F.3})$$

References [4, 5] provide a more detailed discussion on the four-port S-parameter representation and its properties. Figure F.5 shows the measured S-parameters in the format of (F.3) from a four-port VNA measurement of a differential signal trace of a test fixture.

Mathematically it is easy to extend the S-parameter concept to any number of ports but realistically for a high-speed digital I/O the next step would be a 12-port network. This would allow the measurement of three adjacent differential pairs to capture not just the differential transmission of the center signal path, but also the crosstalk from the two adjacent differential transmission lines as shown in Figure F.6. An example of a 12-port network analyzer system is shown in Figure F.7. Note that calibration can become a significant challenge with so many ports if it is done manually.

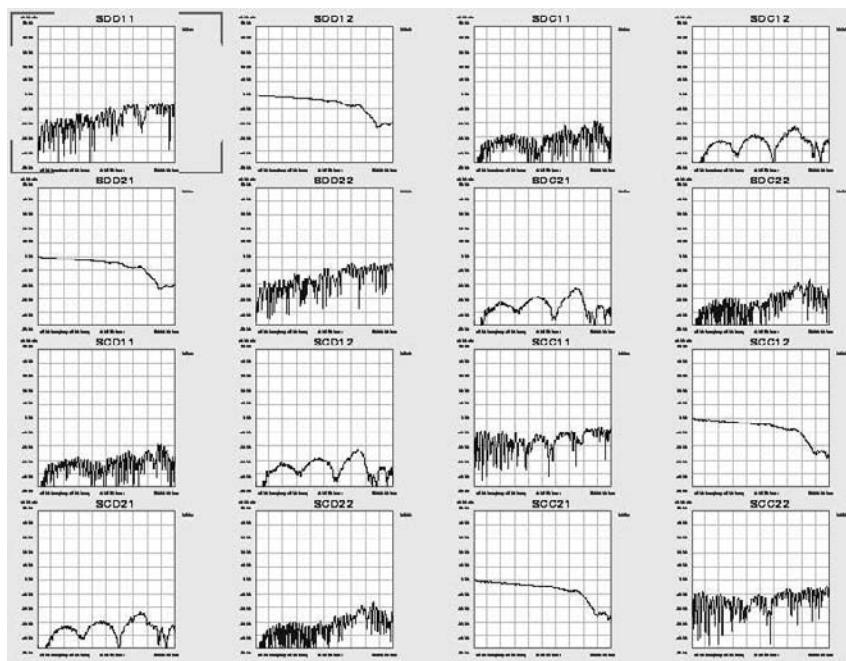


Figure F.5 Measured differential S-parameters from a four-port DUT (differential pair).

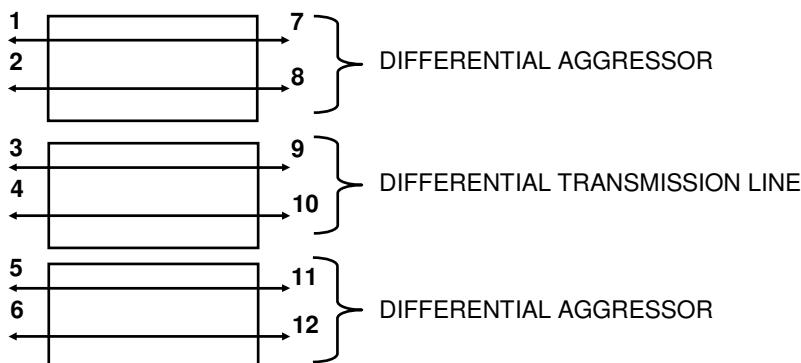


Figure F.6 Block diagram of a 12-port system.

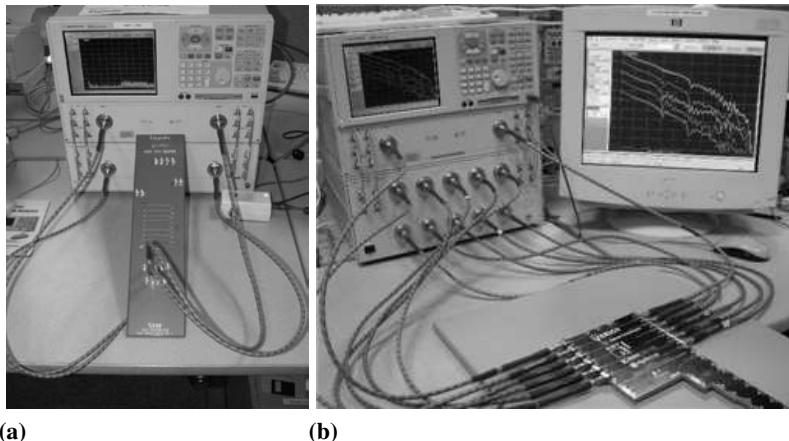


Figure F.7 Picture of (a) a four-port VNA measurement setup and (b) a 12-port VNA measurement setup.

F.1 Simulating and Synthesizing Time-Domain Responses from S-Parameters

As mentioned before, S-parameters are not only useful for visualizing the response of a system in the frequency domain but also for analyzing the system in the time domain. Several software packages can take the measured S-parameters and compute either the time-domain response in the form of a TDR/TDT or provide a data eye diagram for a given data rate and rise time as shown in Figure F.8.

The other important usage of S-parameters is in a simulation where the ultimate goal is to have the simulation predict the performance of a measurement. Just like with the measured S-parameters, the simulated S-parameters of a test fixture performance can be integrated into a more complex simulation that can include models of the ATE pin electronics and of the DUT (Figure F.9). Ideally the transformation between time domain and frequency domain results in no loss of data or corruption of the signal; however, real data both simulated and measured can have nonreal effects at the edges or unwanted noise relative to the desired DUT electrical data. With measured data from a VNA the DC point is extrapolated to zero and in some cases a poor calibration can even result in gain on a passive system. It is important when using band limited S-parameter data to understand the challenges with obtaining passive and causal data and one must always be watchful of data “features” caused by mathematical transformations on nonideal data sets

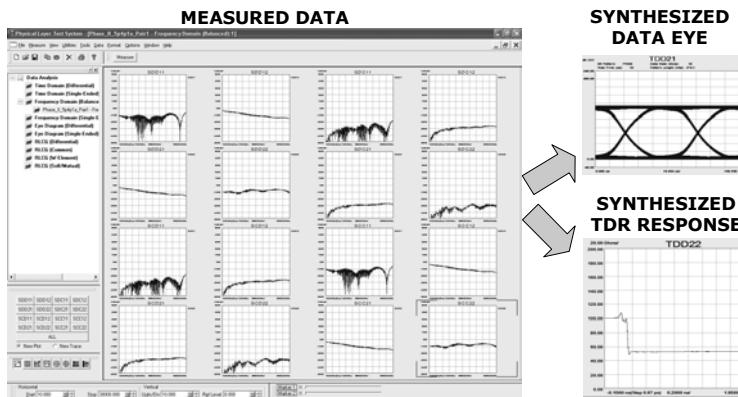


Figure F.8 Example of synthesizing the time-domain response from the S-parameters.

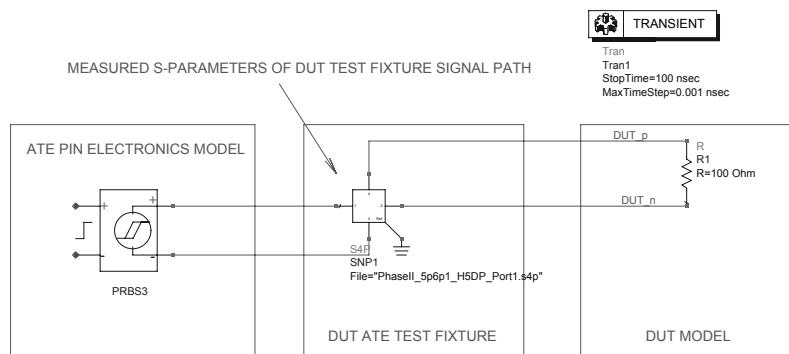


Figure F.9 Example of using S-parameters measured on a test fixture on a time domain simulation of the ATE stimulus waveform at the DUT.

[6, 7].

It is also possible to cascade S-parameters to obtain the simulation of a signal path that consists of multiple elements (connectors, vias, signal traces), each described by an S-parameter data block. Although a CAD simulation tool will show this cascade of S-parameters, it should be noted that mathematically the S-parameters must be converted to a scattering transfer or T-matrix format to properly handle the reflections between two networks. In other words, one cannot simply matrix-multiply the S-parameter matrices of the single path components together to get the total signal path response. The T-matrix has the property that:

$$T_{Total}(f) = T_A(f) T_{DUT}(f) T_B(f) \quad (F.4)$$

where:

$$[T] = \begin{bmatrix} T_{11}(f) & T_{12}(f) \\ T_{21}(f) & T_{22}(f) \end{bmatrix} \quad (\text{F.5})$$

and:

$$\begin{bmatrix} V_1^{Reverse}(f) \\ V_1^{Forward}(f) \end{bmatrix} = \begin{bmatrix} T_{11}(f) & T_{12}(f) \\ T_{21}(f) & T_{22}(f) \end{bmatrix} \begin{bmatrix} V_2^{Forward}(f) \\ V_2^{Reverse}(f) \end{bmatrix} \quad (\text{F.6})$$

Once cascaded together the T-matrix terms are converted back to the S-matrix for direct analysis of reflection and transmission performance. The T-matrix can also be used to go the other direction and actually de-embed a known network element connected to the input or output of a system [8]. In the case of a DUT with test fixture connections on either side:

$$T_{DUT}(f) = T_A(f)^{-1} T_{Total}(f) T_B(f)^{-1} \quad (\text{F.7})$$

This frequency domain technique of de-embedding the test fixture effects (a.k.a. “test fixture adapter”) from the measured performance of a DUT is an effective tool for obtaining a library of components for designing and simulating a full-path system. The accuracy of the de-embedded DUT performance will depend on the calibration techniques used to obtain the “test fixture adapter” data and typically requires the design of custom calibration structures in the form of transmission through lines, reflects, shorts, and loads [9].

F.2 S-Parameters of Coupled Differential Pairs and Structures

The significance of being able to look at differential S-parameters becomes apparent when one starts using differential transmission lines with increased coupling between the two single transmission lines that make up the differential pair. The impedance of the differential system is 100Ω , or 50Ω in series with 50Ω when there is no coupling between the two transmission lines, but as the traces come together to utilize coupled routing for dense routing applications or to minimize sensitivity to ground discontinuities the issues increase in complexity. The single-ended performance with the changes in impedance due to coupling will no longer represent the same electrical performance as that of the differential path with the matched system impedance so it is important to display and simulate the S-parameters differentially [3, 10]. For example, Figure F.10 compares the single-ended

and differential insertion loss of a coupled differential pair, while Figure F.11 shows the synthesized single-ended and differential data eye diagrams.

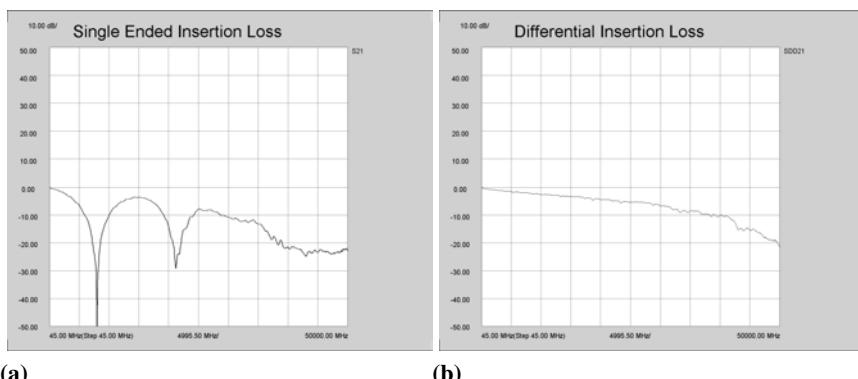


Figure F.10 Comparing (a) the single-ended insertion loss with (b) the differential insertion loss.

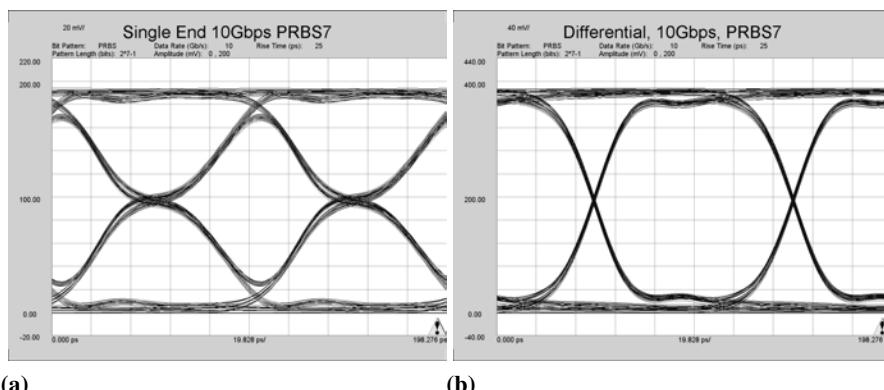


Figure F.11 Comparison of (a) the synthesized single-ended data eye with (b) the synthesized differential data eye.

From the figures it is easy to see that in this case, relying on the single-ended measurements to extrapolate the differential performance of the differential signal path with coupling would result in a significant error.

References

- [1] D. M. Pozar, *Microwave Engineering*. John Wiley and Sons, 2005.
- [2] M. Resso and E. Bogatin, *Signal Integrity Characterization Techniques*. IEC, 2009.
- [3] S. H. Hall and H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*. John Wiley and Sons, 2009.
- [4] D. Derickson and M. Mueller, *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization*. Prentice-Hall, 2007.
- [5] L. Besser and R. Gilmore, *Practical RF Circuit Design for Modern Wireless Systems, Volume I: Passive Circuits and Systems*. Artech House, 2003.
- [6] D. Kaller, C. Schuster, Y. Kwark, D. Altabella, B. Truong, Z. Chen, A. Haridass, and E. Klink, “Using S-Parameters Successfully in Time Domain Link Simulations,” *IEEE 14th Topical Meeting on Electrical Performance of Electronic Packaging*, 2005.
- [7] 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,” *IEC DesignCon*, 2008.
- [8] Agilent Technologies, “De-Embedding and Embedding S-Parameter Networks Using a Vector Network Analyzer,” *Application Note 1364-1*, 2004.
- [9] V. Duperron, D. Dunham, and M. Resso, “Practical Design and Implementation of Stripline TRL Calibration Fixtures for 10-Gigabit Interconnect Analysis,” *IEC DesignCon*, 2006.
- [10] E. Bogatin, *Signal and Power Integrity Simplified*. Prentice-Hall, 2010.

G

Engineering CAD Tools¹

This appendix presents additional detail on several engineering CAD tools that can be important to the test engineer working on high-speed digital applications with ATE. The tools can be divided into model based circuit simulators, finite element electromagnetic (EM) field solvers, data transformation tools, and physical layout tools. There is a significant amount of overlap between the tools and the true art of simulation becomes one of knowing the strengths and weaknesses of the tools and which ones work the best for a given application. Additional references on these topics can be found in [1, 2].

G.1 Circuit Simulators

The primary simulation tool for electrical engineers has always been the circuit simulator, such as the well-known SPICE tool with built-in models for each of the elements in a circuit. Originally these tools focused on lumped element design forcing transmission line effects to be modeled with large quantities of lumped elements to accurately predict the phase and magnitude of a high-speed signal. Increasing the number of elements in a circuit simulator to handle all of the transmission line paths increases the simulation time and reduces the productivity of the tool. The significant increase in data rates for digital I/O on today's circuits has forced most of the circuit simulators to include more efficient transmission line models and some tools are even starting to provide integrated cosimulation with EM field solvers that work into the microwave frequencies. Examples of this type of circuit simulator are HSPICE from Synopsis, Hyperlynx from Mentor, and ADS from Agilent Technologies.

¹In collaboration with Heidi Barnes.

Figure G.1 shows an example using ADS where the objective is to compare the loss versus frequency profile for two transmission lines of different lengths. The simulation tool provides a variety of transmission line models based on the physical topology of the transmission line such as microstrip routed on an external PCB layer or stripline routed on an internal layer. Each model also requires the user to correctly set up a substrate definition to define the properties of the materials being used, the dimensions of the signal trace, and the distances to the ground planes. Figure G.2 compares the simulated insertion loss versus frequency for the two different lengths of stripline that are being modeled. Figure G.3 shows the same model being simulated but now the excitation is in the time domain using a pseudo-random bit sequence pattern source.

A simulation tool is just that, a simulated prediction of a circuit and not a real physical signal. This means that no simulation tool provides perfect results and there are always simplifications and model assumptions inherent to a given circuit simulation tool. The best way to understand the limitations of a given tool and find ways to verify and improve the accuracy is through the use of measurements. Utilizing measurement based modeling provides the ultimate in cross-checking of simulation results so that a model can be optimized for a given application and provides the necessary insights for solving circuit design problems.

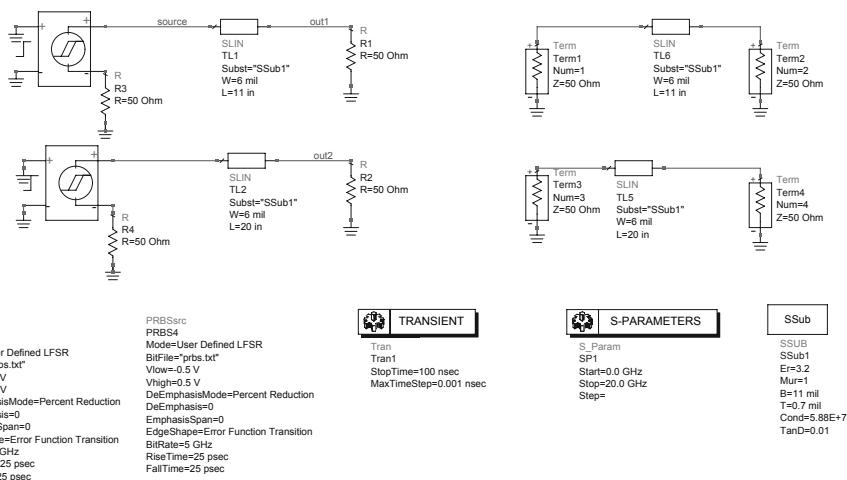


Figure G.1 Example of ADS time- and frequency-domain simulation setup.

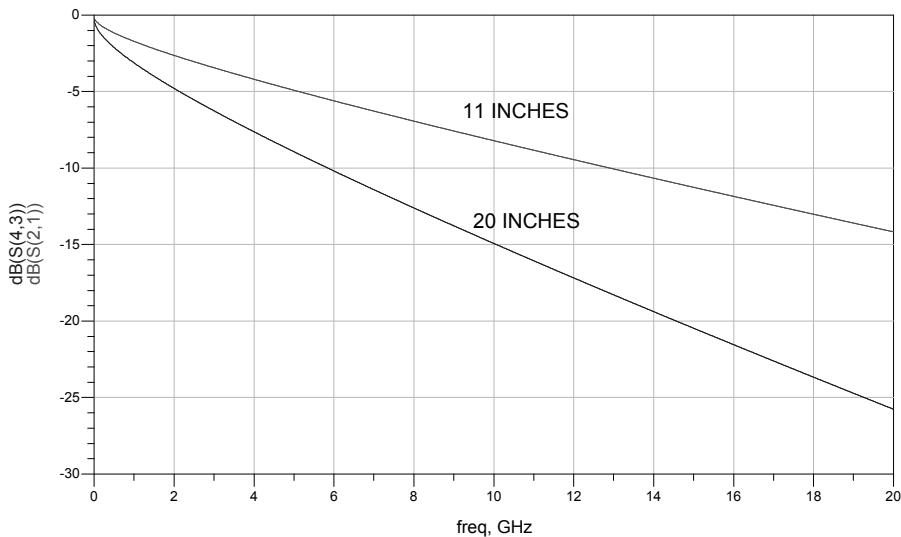


Figure G.2 Results from the ADS simulation in the frequency domain.

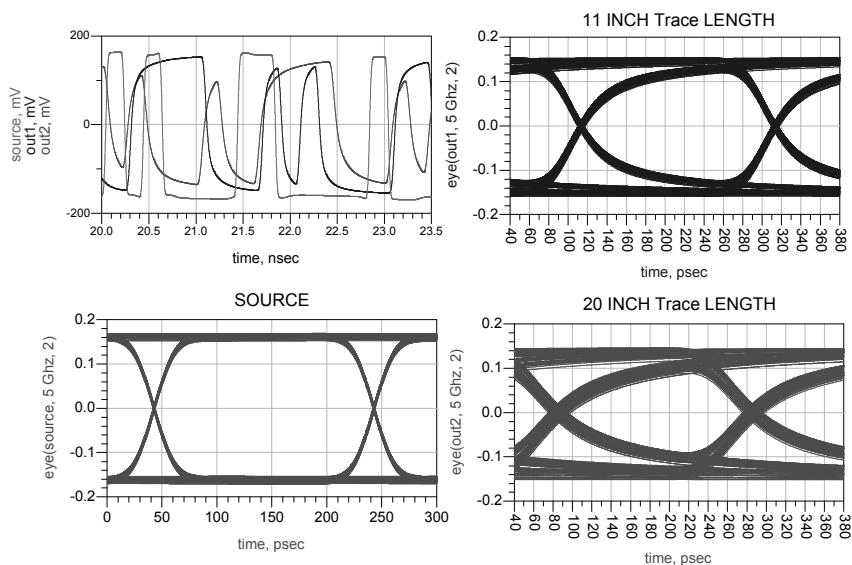


Figure G.3 Results from the ADS simulation in the time domain.

G.2 3D EM Field Solvers

3D EM field solvers provide a very powerful way to analyze a circuit when simplified models are unavailable or no longer accurate enough. In a 3D EM simulation a 3D model of the problem is created with the appropriate material properties defined and then Maxwell's equations for the fundamental properties of electricity and magnetism are solved through a finite element process that divides the 3D structure into small units [3]. Some of the major differences in the available 3D EM field solvers are in the methodology of how the small units are created and whether Maxwell's equations are solved in the time domain or in the frequency domain. References [2, 4] provide a good overview of 3D EM simulation methodologies and Figure G.4 shows an example of a 3D EM field solver with a pogo via design problem.

In the context of test fixture design, 3D EM simulation is critical for the analysis and optimization of key structures like vias, pogo vias, transitions to connectors, transitions to relays, and so on. These structures can vary significantly from those used on the final DUT application board and the 3D EM field solver is the best way to quantify the differences and optimize the impedances for high-speed ATE performance. The 3D EM field solver is based on the fundamental properties of electromagnetic fields; however, it is still a simulator with some model assumptions and user-entered material properties that can still benefit from measurement based modeling techniques to improve the accuracy [5].

G.3 2D Planar Field Solvers

Since the individual layers of a PCB are planar structures, it is possible to simulate them without resorting to a full 3D EM field solver. There is a special class of EM field solvers called 2D EM field solvers that are able to simulate planar structures in a faster way than a 3D EM field solver [2]. Figure G.5 shows one example of a commercial 2D EM field solver. Although 2D EM field solvers are faster than full 3D EM field solvers, they are not able to properly address problems that are truly 3D in nature like a connector, a via, or a DUT socket. However, via models are typically added so that they can be used for signals that route on more than one layer of a PCB. The faster simulation time makes it practical to simulate multiple signal traces on an ATE test fixture to look at signal performance and crosstalk issues.

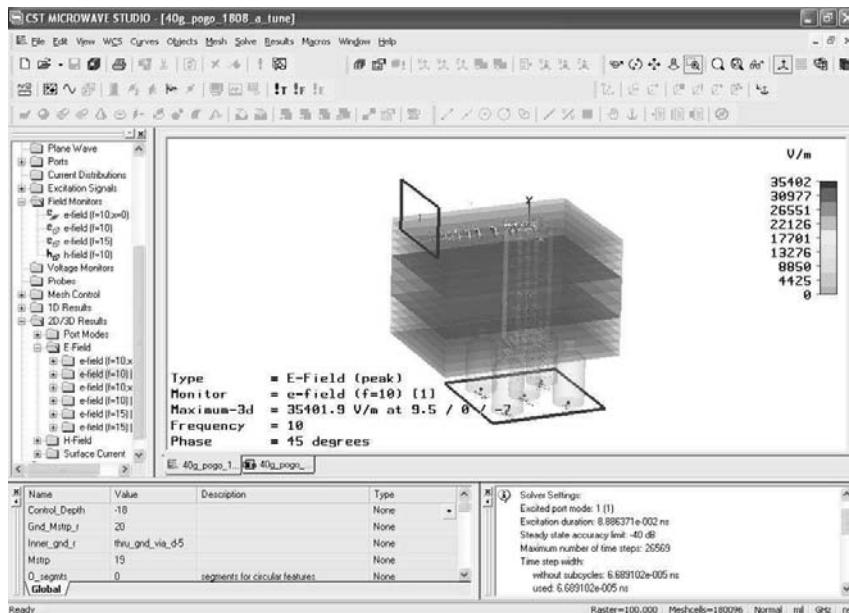


Figure G.4 Example of 3D EM simulation of a pogo via design using CST Microwave Studio.

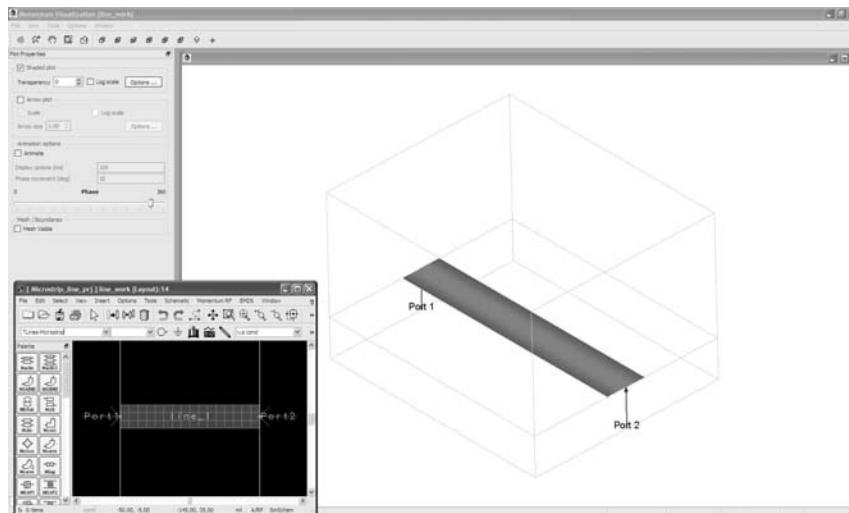


Figure G.5 Example of a microstrip line simulation using Agilent Technologies Momentum 2D EM field solver.

G.4 Power Integrity

Highly specialized tools can be found in the area of power integrity where field solvers and models are combined to specifically address the design of a PCB power distribution network (PDN) [6]. These tools simplify the user interface for modeling a PDN network, such as importing from a PCB layout tool, and entry and placement of decoupling capacitors.

The models and field solvers are also optimized for the large planar surfaces and hundreds of connecting ground vias so that simulation times remain reasonable. These tools allow the user to address some of the specific topics that are associated with a test fixture PDN design like optimum placement of decoupling capacitors, power plane resonances, unwanted coupling between PCB features, and so on. Figure G.6 shows an example of a power distribution network simulation software package.

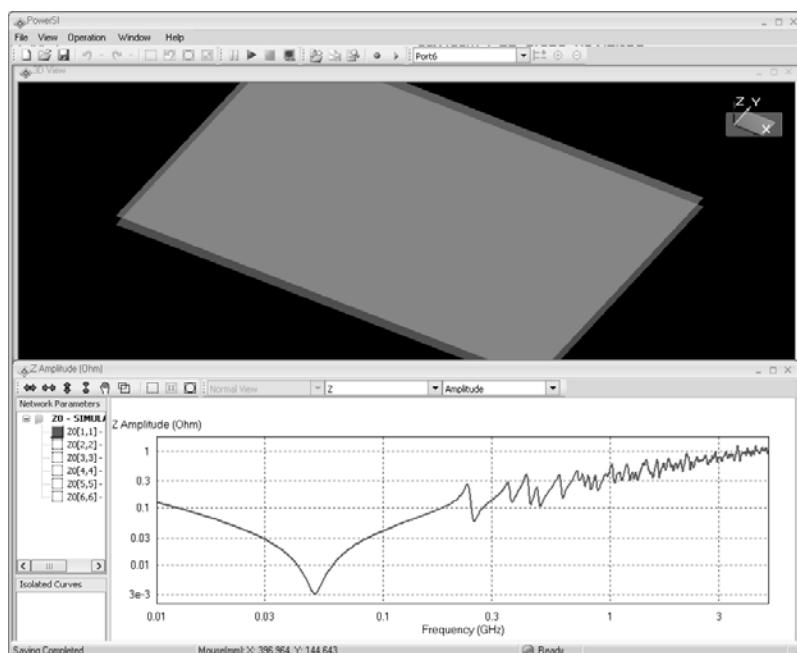


Figure G.6 Example of a PCB power distribution network analysis software modeling the impedance of a power plane (courtesy of Sigrity).

G.5 Model Generation

Ideally, it is possible to use the different simulation tools previously described together to address an engineering project; for example a 3D EM simulator might be used for the connector and via design, a 2D EM simulator for the PCB board layout, and a SPICE model for the I/O cell. This approach can be quite challenging since tool integration is not always optimal and one may need to translate or develop a model of the data to import the data from one tool to another. Model generation is also quite useful for transforming measured data into a format that is compatible with a simulator tool. In some cases this model creation can be automated like, for example, generating a SPICE model from measured S-parameters. A full featured simulation package typically includes this type of model generation and verification capabilities but stand-alone software packages are also available.

G.6 Other Tools

One very important CAD tool in the design of high-speed digital I/Os is the layout tool for simulating how the fabricated PCB layers will look. The word simulate is used here to emphasize that there are manufacturing tolerances like etching and drilling that need to be considered as data rates move into the multigigabit domain, and one may need to work closely with the fabricator to understand any modifications that are made to the data. Layout tools are rather prolific but few are specifically designed for high-speed digital layout and simple things like backdrilling to remove signal via stubs, trace necking into a BGA, and rounded trace bends may not be completely automated and therefore require a significant amount of manual checking. Figure G.7 shows an example of these types of tools.

Typically when one only needs to visualize or check a PCB layout there exists a free layout viewer provided by the PCB layout software company. The layout viewer should be used with some caution since it is not always clear how the different positive and negative layers may be merged together to form a single PCB layer. To verify that the layout is being interpreted correctly it is always important on high-speed designs to look at the final CAD layers that are generated at the end of layout during the Gerber generation process. This process converts the variety of layout layers to a single image layer for each PCB layer and saves it in a Gerber format that is compatible with the tools and equipment at the PCB fabricator.

The Gerber layers are far easier to overlay and turn on and off for visual checking of ground and power plane antipad features that are critical in achieving multigigabit performance. In some special cases where the

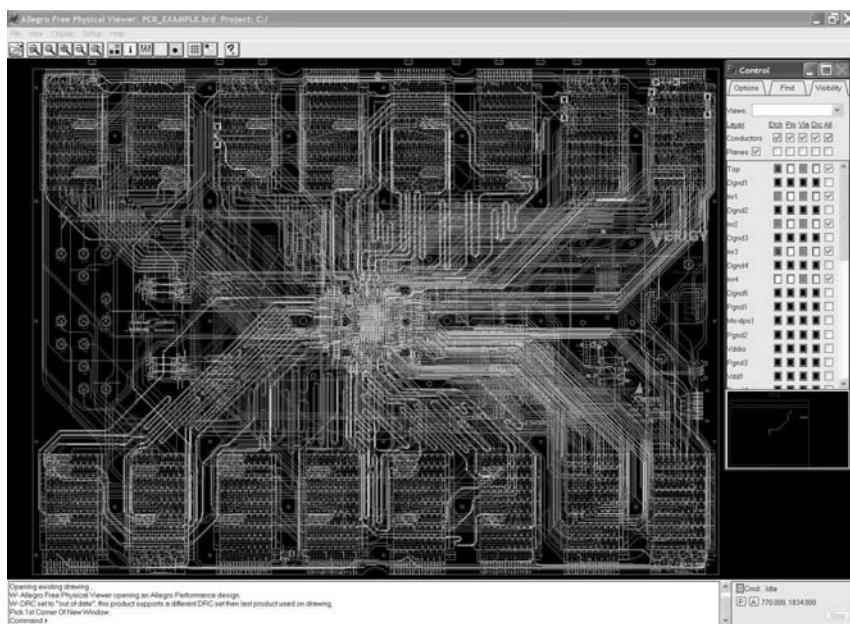


Figure G.7 Visualizing a test fixture layout using Cadence Allegro free physical viewer.

fabricator requests to add additional metal in the empty spaces of a signal layer to improve etching uniformity, one should also request the modified Gerbers from the PCB fabricator. Figure G.8 shows one example of a Gerber visualization tool.

Finally, another important tool when planning or designing a test fixture PCB is an impedance calculator that allows the design of the correct stack-up and trace geometry for a given target impedance and dielectric material. On the planning phase of a PCB test fixture this type of tool is very helpful in analyzing possible trade-offs and limitations on the stack-up for complex multilayer PCB test fixtures. Figure G.9 shows an example of this type of tool. Once the stack-up is determined it is important to request that the PCB fabricator provides the exact trace widths for a desired impedance since this could vary by 25 or 50 μm (1 or 2 mil) depending on the fabrication process.

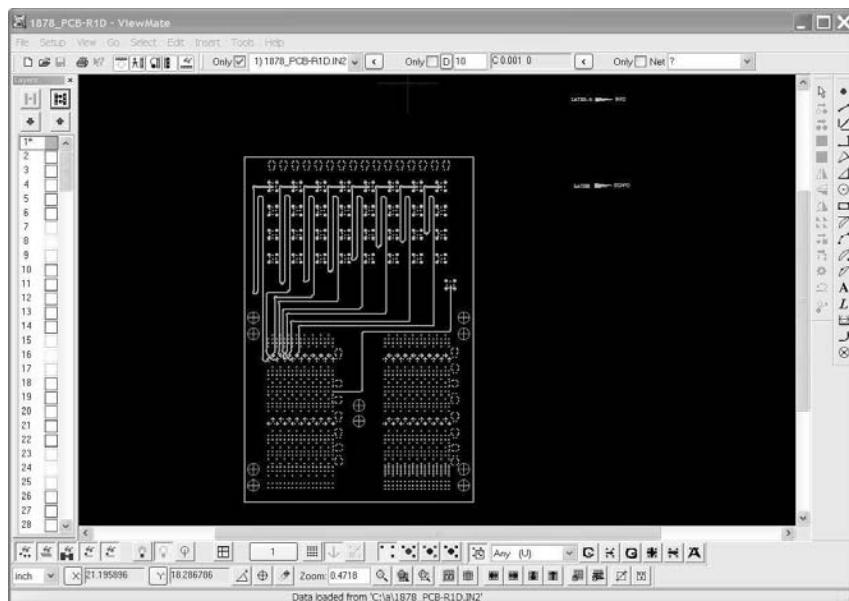


Figure G.8 Visualizing the Gerber files of a test fixture using Pentalogix ViewMate.

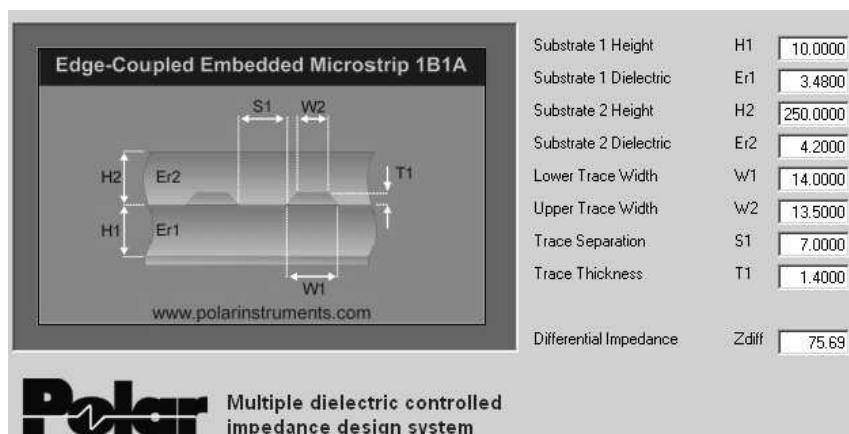


Figure G.9 Example of impedance computation for a microstrip using Polar Instruments impedance design system.

References

- [1] G. Edlund, *Timing Analysis and Simulation for Signal Integrity Engineers*. Prentice-Hall, 2008.
- [2] D. G. Swanson and W. J. R. Hoefer, *Microwave Circuit Modeling Using Electromagnetic Field Simulation*. Artech House, 2003.
- [3] K. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," *IEEE Transactions on Antennas and Propagation*, 1966.
- [4] T. Weiland, M. Timm, and I. Munteanu, "A Practical Guide for 3-D Simulation," *IEEE Microwave Magazine*, 2008.
- [5] Y. Shlepnev, A. Neves, T. Dagostino, and S. McMorrow, "Practical Identification of Dispersive Dielectric Models with Generalized Modal S-Parameters for Analysis of Interconnects in 6-100 Gb/s Applications," *IEC DesignCon*, 2010.
- [6] O. P. Mandhana and J. Zhao, "Comparative Study of Effectiveness of On-Chip, On-Package and PCD Decoupling for Core Noise Reduction by Using Broadband Power Delivery Network Models," *Electronic Components and Technology Conference*, 2005.

H

Test Fixture Evaluation and Characterization¹

H.1 Measuring the Test Fixture Performance

Measuring the performance of a test fixture is an important and yet difficult task for high-speed digital applications. The ability to accurately measure the signal degradation at these higher data rates is necessary in order to fully characterize the test fixture and identify how it will impact the characterization and testing of a DUT. The measured performance of the test fixture also provides the ability to create full path simulations with models of the ATE pin electronics and DUT to further optimize future test fixture designs and evaluate methods of de-embedding the test fixture effects. To understand how critical it is to measure the performance of an ATE test fixture one can ask the following questions:

- Is this the first time that the PCB design house and/or fabricator has built this test fixture?
- Is a previous high-speed design being leveraged for use at even higher data rates?
- Were there signal performance problems with the prior test fixture design/fabrication?
- Have any of the PCB materials or components in the high-speed paths changed?

If the answer is yes to any of these questions, then measuring the test fixture performance is critical before starting implementation of the

¹In collaboration with Heidi Barnes.

application on the ATE system. Following through with this measurement step is not simple and is often skipped for one of the following reasons:

- **Lack of the needed equipment:** High-speed measurements with gigahertz bandwidths need specialized oscilloscopes, TDR/TDT modules, and vector network analyzers that are not always available in a standard ATE lab.
- **Lack of measurement expertise:** High-speed measurements can easily be degraded by the probes or techniques being used. Without the experience to understand when a measurement is being done poorly, it is very easy to end up with erroneous data. A lack of confidence in the measurement methods and equipment can then make it difficult to invest further in improving or fixing the measurement accuracy.
- **Lack of time:** This is the typical reason for skipping the measurement of a test fixture. Test engineers prefer not to delay the schedule for test fixture measurements and go straight to running the application on the ATE system in the hope that if everything works then no measurements are required. On high-speed designs this can often lead to longer delays with extremely long application debug times as one tries to sort between a test problem, a DUT silicon problem, and a test fixture problem.

The inability to accurately measure the test fixture performance makes it difficult to understand how the test fixture is influencing the DUT measurement and reduces the ability to correlate ATE measurements with bench instrumentation measurements of the DUT. One possible alternative is to subcontract the measurement of your test fixture. Figure H.1 shows an example setup of a probing station coupled with high-end bench instrumentation and micro-coaxial probes used to measure the performance of your test fixture PCB board. Of course, this task will entail a financial cost. One disadvantage is that the subcontractor evaluating your test fixture might not know your application in detail, so close cooperation may be required.

Since the accuracy of the test fixture measurements can depend on the techniques and probes being used, it is important to start planning for how the measurements will be made at the same time as the test fixture design. The addition of test coupons, interface boards for probing at the DUT socket, and calibration structures on a test fixture panel can have significant benefits in improving the ability to accurately characterize an ATE test fixture. The following sections go into additional detail on these methods for improving test fixture characterization.

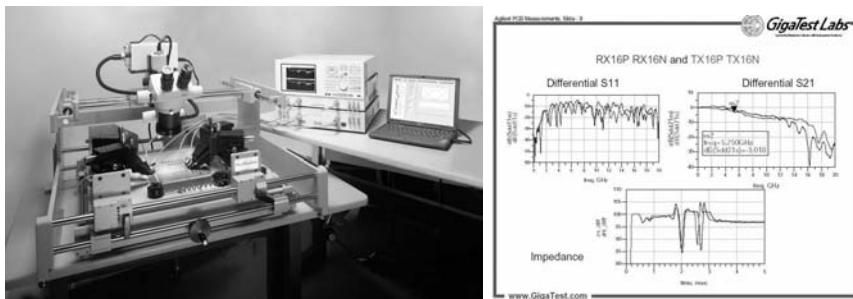


Figure H.1 GigaTest Labs probing station with measurement instrumentation and a report example (courtesy of GigaTest Labs).

H.1.1 Test Coupons

Test coupons are an excellent method for obtaining a significant amount of information on a test fixture. A test coupon takes advantage of the empty space left on a test fixture PCB panel to build a separate board with test patterns. The test patterns replicate some of the signal paths from the main board to check losses and impedance matching on the different routing layers, via transitions, back drilling performance, relays, and so on. To simplify the measurement with bench instrumentation, the test coupon can make use of coaxial connectors to connect to the PCB. For example, instead of the DUT PCB footprint and socket one could use a coaxial connector where the DUT socket pad would be.

An example of a test fixture and its associated test coupon is shown in Figure H.2. This example has several traces on the test coupon that correspond to the shortest and longest length traces on the test fixture with a connection on one end to a coaxial edge connector and at the other end to a pogo pin PCB via array. Trace bends and via transitions were also included for obtaining information on how an individual element of a design can affect the signal integrity.

This test coupon highlights the need to include the transition from the ATE pogo assembly to the test fixture via array since the overall performance can be significantly influenced by the PCB via topology. This can be achieved by using a modified pogo pin assembly with short cables and SMA connectors together with a mechanical bracket to perform the test coupon measurement as shown in Figure H.3 [1].

With this measurement setup it is possible to measure the performance of the different traces on the test coupon including in this case the effect of the pogo via and pogo assembly as shown in Figure H.4. The measurements were done by connecting the ports on a VNA to the pogo assembly SMA connector

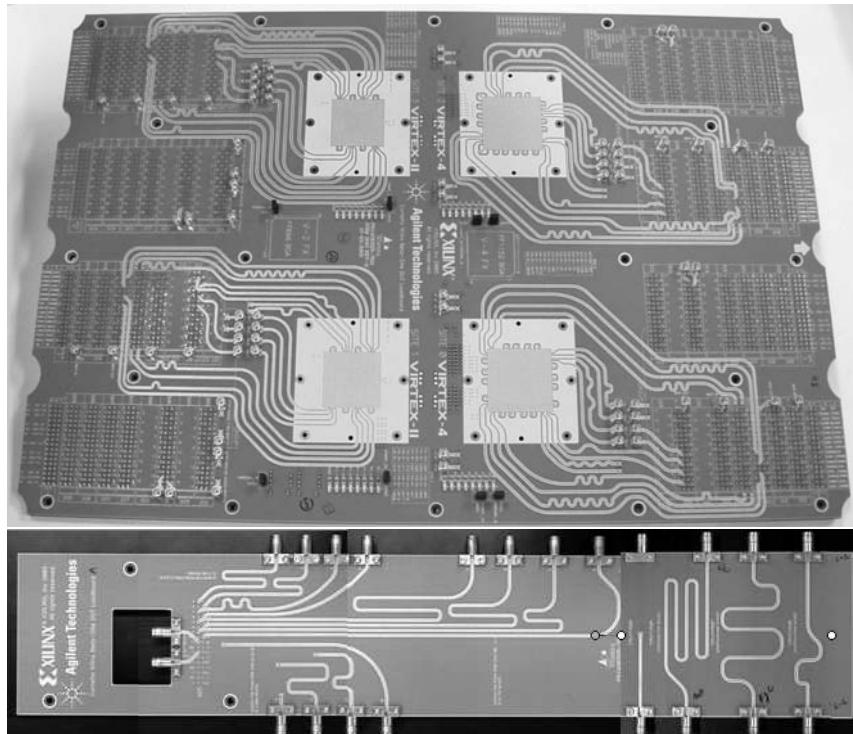


Figure H.2 Example of a test fixture (top) and its corresponding test coupon (bottom) (courtesy of Verigy).



Figure H.3 An ATE bench pogo pin assembly (left) docked to the test coupon in a lab environment (right) (courtesy of Verigy).

and to the edge-mounted connector at the end of the trace and measuring the insertion loss. Other types of measurements using different instruments are also possible with this approach.

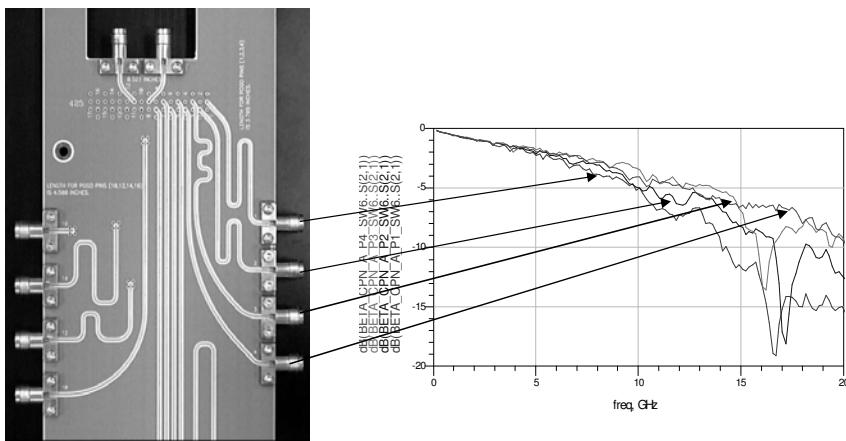


Figure H.4 Measurement results for some structures on the test coupon in Figure H.3.

The test coupon does a good job of characterizing the basic design of the test fixture and identifying any major design flaws, but it is not a total guarantee of the test fixture performance. Variations in etching and registration across a panel will result in differences between the test coupon at the edge of the panel and signal paths routed in the center. These small variations are typically ignored at data rates below 1 Gbps, but as data rates increase so does the sensitivity to these PCB tolerances. Also, it is difficult to replicate the DUT footprint, power distribution network, and the routing density to accurately predict all of the crosstalk and signal losses at higher speeds on a test coupon.

H.1.2 Test Fixture Socket and Socket Via Field Probing

Directly measuring the test fixture high-speed signal paths is the ideal characterization method. To perform this measurement the connection to the ATE pogo via arrays on one end of the test fixture can be done using the same modified pogo assembly that was used on the test coupon with the short cables and SMA connectors. Connecting to the other end with the DUT socket is not as simple. The challenge at the DUT socket is to provide a matched impedance probe connection to the DUT socket that replicates the signal and ground topology that the DUT will see. Variations in BGA pitch and ballout topology

for signal and ground connections along with single-ended and differential connections increase the complexity of defining a method for accurate and repeatable measurements.

An understanding of the probe performance for a specific application may not be as easy as just looking at the data sheet from the probe vendor. Physical differences between the probe topology and the test fixture can lead to additional impedance mismatches that can significantly degrade a multigigabit signal. Common oscilloscope type probes with a side ground lead like those discussed in Section 7.11.5 suffer from the added inductance or higher impedance as this ground wire moves further from the signal path.

Commercially available impedance matched probes with extremely small submillimeter size probe tips have long been used in the wafer probing industry for RF/microwave applications and have the ability to provide the best possible high-speed connection to the ATE test fixture. Leveraging this probing technology does require a special interface or “interposer” PCB [2] to optimize the transition from the probe to the test fixture both mechanically and electrically. The interposer provides the mechanical strength to compress the DUT socket pogo pins and an optimized electrical transition from the ground signal spacing of the probe to that of the test fixture.

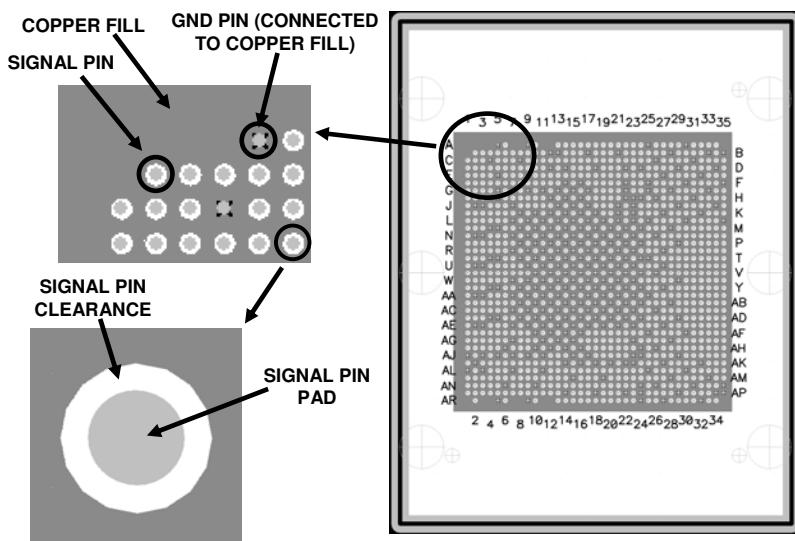


Figure H.5 Description of the layout details for one example of a DUT socket interposer.

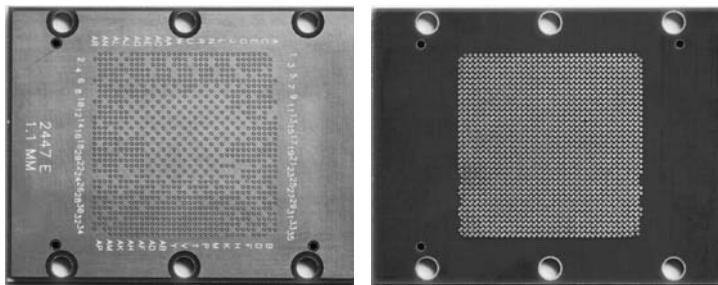
The PCB interposer design is simply an array of vias that line up with the exact DUT BGA footprint on the bottom side and are filled and plated

over to provide a flat surface for the socket pogo pin connection. On the top side a ground fill is added that connects all of the ground vias together and surrounds each signal and power via so that the via pad to ground fill spacing matches the ground to signal spacing of the micro-coaxial probe. Figure H.5 shows the details of the top ground fill for a probe interposer board. Figure H.6 shows how the interposer is attached to the test fixture socket with the mounting holes and alignment pins. The figure also shows the micropositioner used to correctly position the probe with the help of a microscope. Advanced interposer designs utilize 3D EM simulations to add internal ground layers and additional interstitial ground vias to reduce crosstalk and improve impedance matching between the test fixture and the micro-coaxial probes as shown in Figure H.7.

Note that to use an interposer with a socket it is necessary to mechanically modify the socket to obtain a flat surface where the interposer can attach as shown in Figure H.8.

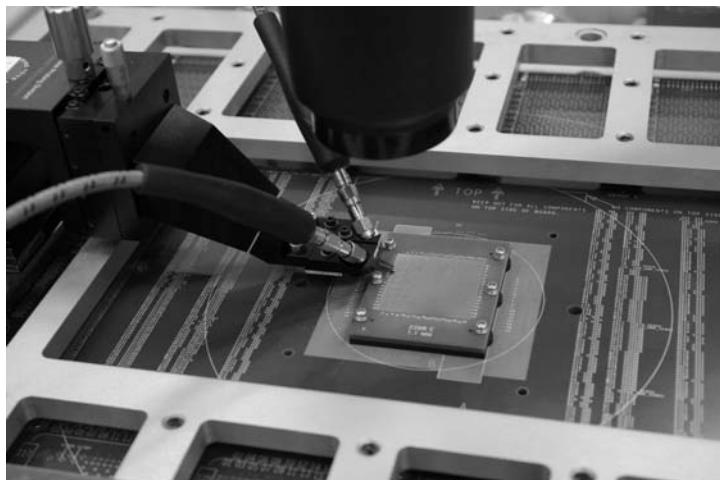
The PCB probe interposer also enables the measurement of the test fixture without the pogo pin style socket. The footprint on the bottom of the interposer matches directly with that of the test fixture PCB. With the addition of a thin vertical interconnect material compressed between the PCB probe interposer and the test fixture DUT PCB footprint a good electrical connection can be made. Figure H.9 shows an example of a thin <0.5 mm vertically conductive elastomer material that uses columns of silver plated nickel balls with high current carrying capacity to make multiple connections per signal path. The ability to measure the test fixture with and without the ATE socket provides additional information on the impact this transition has on the overall signal integrity of the signal path.

The selection of which micro-coaxial probes to use is less dependent on the manufacturer and more a function of selecting the best configuration and pitch for the ground and signal connections [2]. Micro-coaxial probes with two side ground connections (ground-signal-ground or GSG topology), provide a low loss connection that is more tolerant of the variability in ground via locations on the interposer. A tighter spacing between the GSG leads will reduce crosstalk with adjacent probes and reduce the size of the transition discontinuity to a micro-coaxial probe. Figure H.10 shows two examples of micro-coaxial probes in use with the PCB probe interposer to measure an ATE test fixture. Figure H.11 shows the complete view of an ATE test fixture with pogo assembly connections and micro-coaxial probe connections to a bench VNA for complete frequency characterization of a signal path. It is also important to remember that when performing any measurement to characterize a test fixture that the calibration and de-embedding of the measurement setup is critical [3, 4].

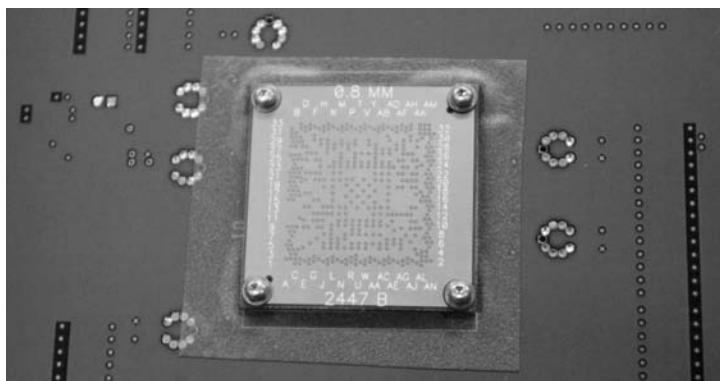


(a) Interposer (top).

(b) Interposer (bottom).



(c) Using a probe together with an interposer to measure the test fixture performance including the DUT socket.



(d) Interposer connected directly to the test fixture without the socket using a Paricon material sheet.

Figure H.6 (a-d) Example of an interposer in test fixture characterization (courtesy of Verigy).

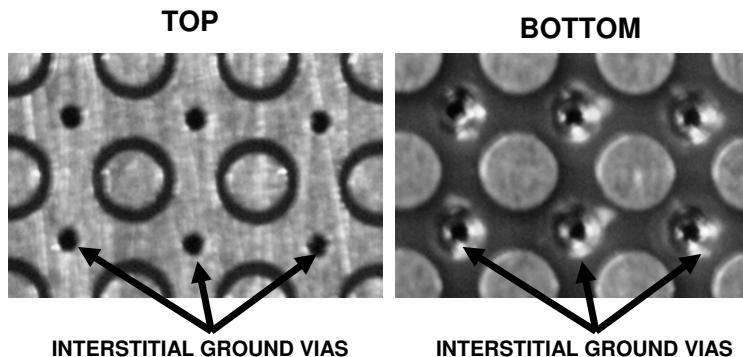


Figure H.7 Example of an interposer design with interstitial ground vias surrounding the signal vias used for probing.

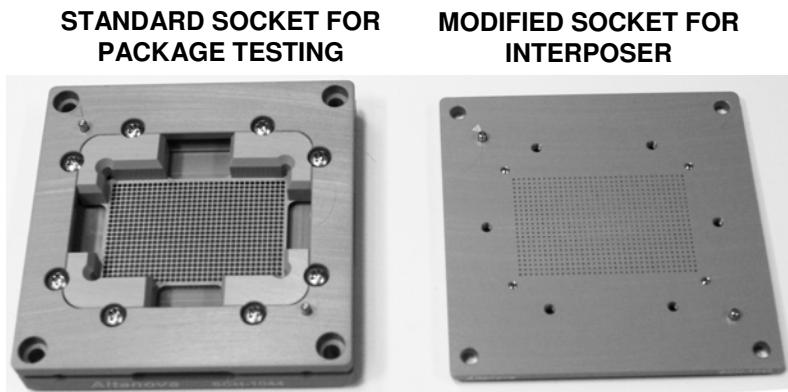


Figure H.8 Example of a modified socket for use with an interposer (courtesy of Verigy).

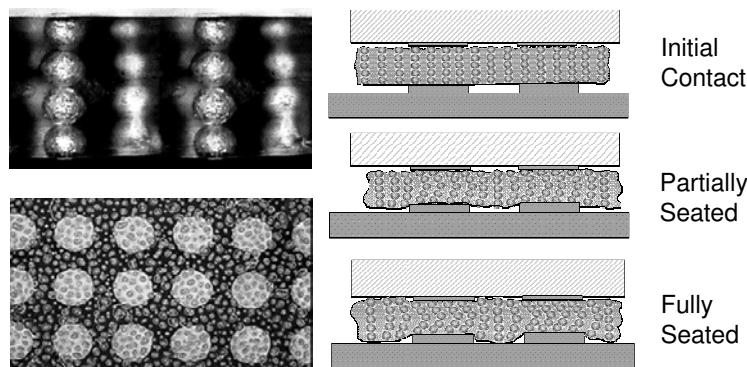


Figure H.9 Picture showing the Paricon material and how it works (courtesy of Paricon Technologies).

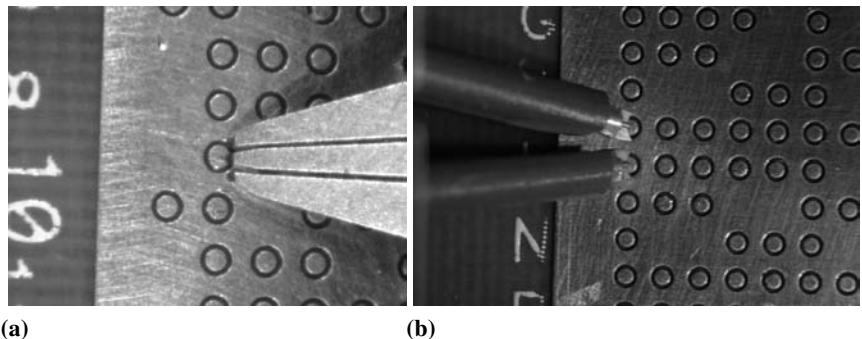


Figure H.10 Probing the signal via at the interposer (a) using a SUSS single-ended (GSG) coaxial probe, and (b) using a GGB picoprobe dual probe (GSGGSG).

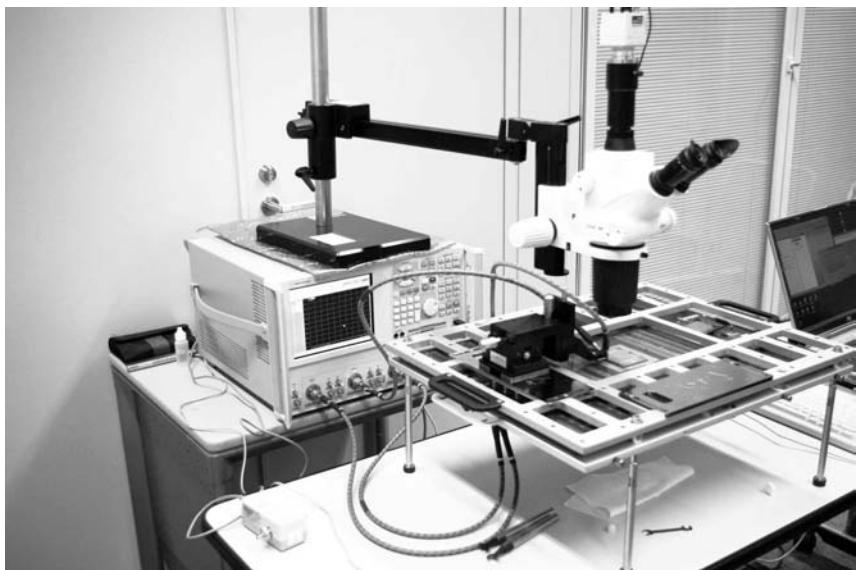


Figure H.11 Example of a bench characterization setup for ATE test fixtures (courtesy of Verigly).

The ability to probe the DUT ATE test fixture at the DUT location can also be used for in situ measurements with the test fixture docked on the tester. The micro-coaxial probe makes it possible to send a signal from a calibrated source into the test fixture for measurement by the ATE system or to utilize calibrated bench instrumentation to measure the performance of the ATE transmitted signal that reaches the DUT socket interface. This type of in situ measurement is discussed in Section 9.3 and [5] captures the full path performance of the test fixture and the ATE system which can then be used for focus calibrating an ATE system for improving its accuracy.

H.2 Measuring the Test Fixture Power Distribution Network

The power distribution network (PDN) of a test fixture is a critical part of a high-speed digital application, especially when high-current applications like microprocessors are involved. It is important to note that some test procedures like a scan test can have significantly different power requirements when compared to the nominal operation of the DUT.

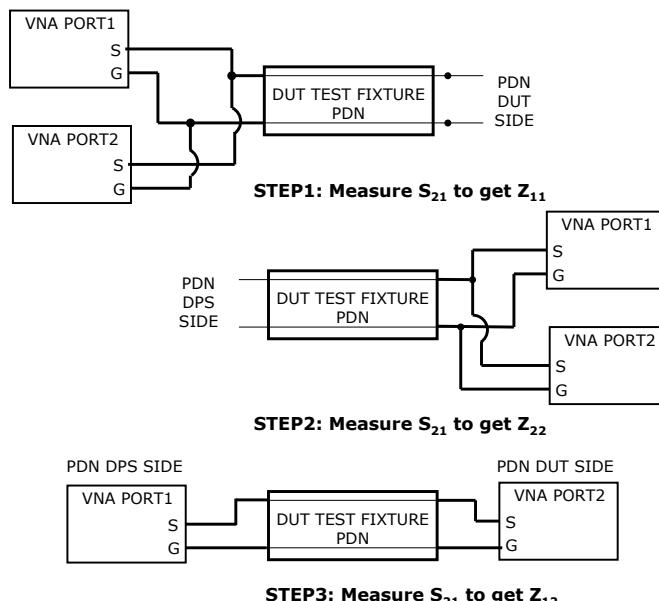


Figure H.12 Setup for a PDN measurement in the frequency domain with a VNA.

Given the importance of the test fixture PDN, it becomes necessary to measure its performance on the final manufactured test fixture with all

the decoupling capacitors already assembled. There are several techniques to measure the performance of a power distribution network. One typical approach is characterization on the frequency domain by using a VNA. Figure H.12 shows a diagram of a setup for measuring the performance of a test fixture PDN with a VNA based on the techniques presented in [6–8]. The figure shows that to obtain a complete impedance measurement it is necessary to perform three measurement steps. The final values of the impedance matrix Z can then be computed using (H.1) [7]:

$$\begin{aligned} Z_{11} &= \frac{Z_{\text{VNA}}}{2} \frac{S_{21}^{(1)}}{1 - S_{21}^{(1)}} \\ Z_{22} &= \frac{Z_{\text{VNA}}}{2} \frac{S_{21}^{(2)}}{1 - S_{21}^{(2)}} \\ Z_{12} = Z_{21} &= \frac{Z_{\text{VNA}}}{2} S_{21}^{(3)} \frac{1 + \frac{Z_{11}}{Z_{\text{VNA}}} + \frac{Z_{22}}{Z_{\text{VNA}}} + \frac{Z_{11}}{Z_{\text{VNA}}} \frac{Z_{22}}{Z_{\text{VNA}}}}{1 + S_{21}^{(3)} \frac{Z_{11}}{2Z_{\text{VNA}}}} \end{aligned} \quad (\text{H.1})$$

where $S_{21}^{(1)}$ is the S_{21} value measured in step 1 of Figure H.12, $S_{21}^{(2)}$ is the S_{21} value measured in step 2, and $S_{21}^{(3)}$ is the S_{21} value measured in step 3. Z_{VNA} is the VNA impedance value which is typically 50Ω . For low impedance values (which is the case of a properly designed PDN), (H.1) can be further simplified into (H.2).

$$\begin{aligned} Z_{11} &= 25 S_{21}^{(1)} \\ Z_{22} &= 25 S_{21}^{(2)} \\ Z_{12} = Z_{21} &= 25 S_{21}^{(3)} \end{aligned} \quad (\text{H.2})$$

In most cases, the test engineer is only interested in the performance of the power planes and decoupling capacitors in the immediate vicinity of the DUT. In this case it is enough to perform step 2 measurement of Figure H.12 to obtain the PDN self-impedance at the DUT. Another critical point when measuring a test fixture PDN is how the probes are connected to the test fixture. Note that since one usually needs to use a measurement point connected to a via to probe the PDN, the inductance of the via will dominate the measured results unless a proper probing configuration is used. This is shown in Figure H.13 where the optimum configuration requires that the PDN is probed simultaneously from both the top and the bottom of the PCB via [7].

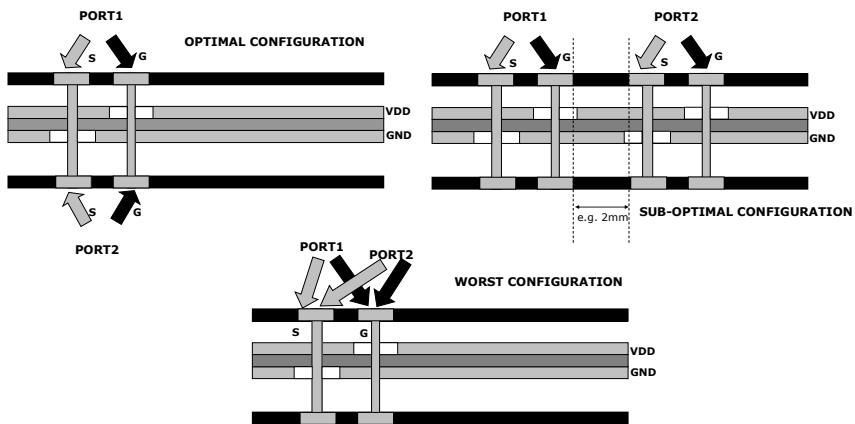


Figure H.13 Possible approaches to probe the test fixture PDN showing how to avoid that the via inductance masks the PDN measurement.

After computing the Z matrix, it is important to check if the matrix is passive.² Passivity of a Z impedance matrix can be verified by computing the eigenvalues of the $Z + Z'$ matrix at each frequency point where Z' is the conjugate transpose matrix. For the Z matrix to be passive all the eigenvalues must be positive or zero [9].

Figure H.14 shows a picture of two measurement setup examples used to measure the power distribution network impedance of an ATE test fixture.



Figure H.14 Pictures of power distribution network measurement setups (left: measuring Z_{22} ; right: measuring Z_{12}).

²A system is passive if it is unable to generate energy. Clearly this is a requirement for a PDN since the PDN itself cannot generate energy, only the power supply. Reference [9] provides a precise mathematical definition.

Figures H.15 and H.16 present the results of measuring the PDN of a real DUT ATE test fixture. Figure H.15 shows the measured insertion loss (S21) in each step of the measurement procedure shown in Figure H.12, and Figure H.16 shows the computed impedance values using (H.1).

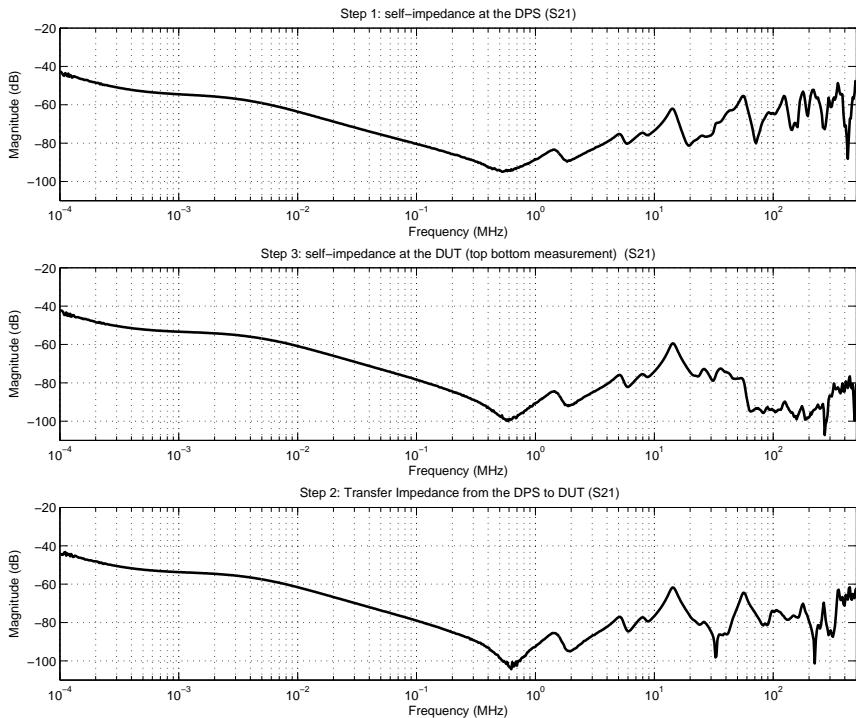


Figure H.15 Measured insertion loss (S21) for step 1, step 2, and step 3 of Figure H.12.

These results can then be used for verifying the performance of the test fixture PDN by comparing with the expected results (e.g., from simulation) or used as the basis to develop a simulation model of the PDN to be used together with models of the ATE DUT power supply and the DUT power requirements.

As a final note, Figure H.17 shows the measured results for the optimal and suboptimal case PDN measurement methods as described in Figure H.13. Note that when using the worst case configuration the via inductance will dominate the measured data after 200 kHz and in this way mask the real impedance of the test fixture PDN. In this example the test fixture height was about 5.8 mm (200 mil) with the power planes situated in the middle of the stack-up. This shows the importance of optimizing the probing approach for PDN measurements [7].

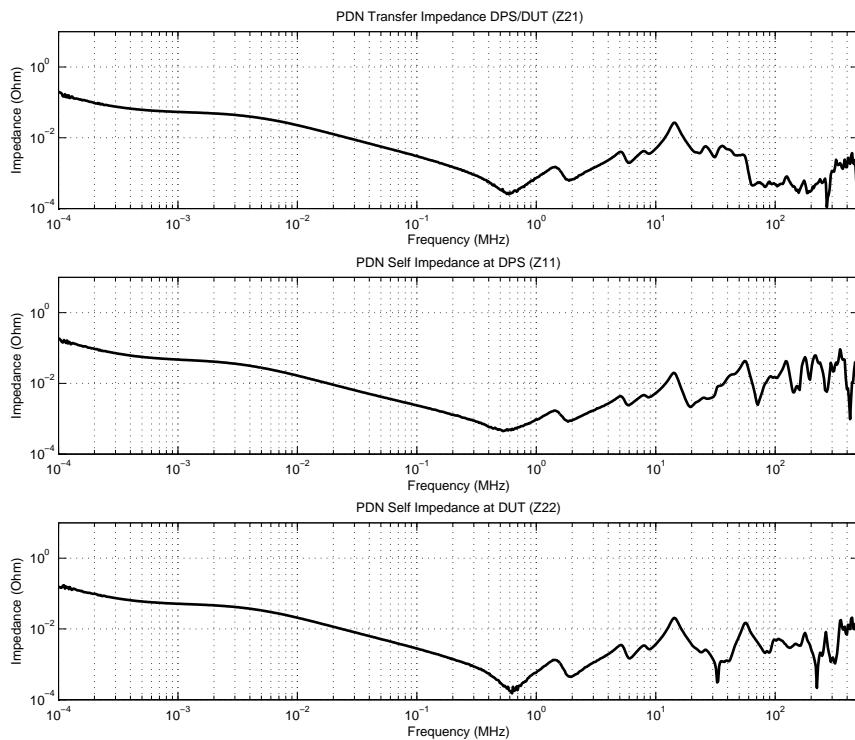


Figure H.16 Computed impedance values for the power distribution network of a test fixture: Z21 transfer impedance between the DPS and the DUT (top); Z11 self-impedance at the DPS (center); and Z22 self-impedance at the DUT (bottom).

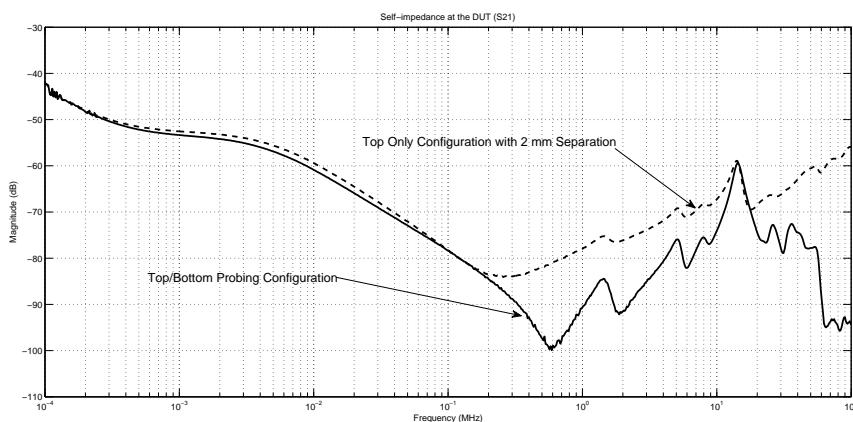


Figure H.17 Comparison of the measured S21 for a top/bottom probing configuration with a top only configuration with the two probes 2 mm apart.

Another measurement of interest when analyzing a test fixture PDN is the power spectrum of the voltage at the DUT power supply terminal. Since this measurement requires that the DUT is being stimulated with a certain pattern, it is necessary to add a monitor mechanism during the test fixture design. Figure H.18 shows one possible approach where a $50\ \Omega$ transmission line is connected to one DUT power supply via and then routed to a probing point on the top of the test fixture. The $50\ \Omega$ functions as a high impedance probe in comparison with the low impedance of the test fixture PDN.

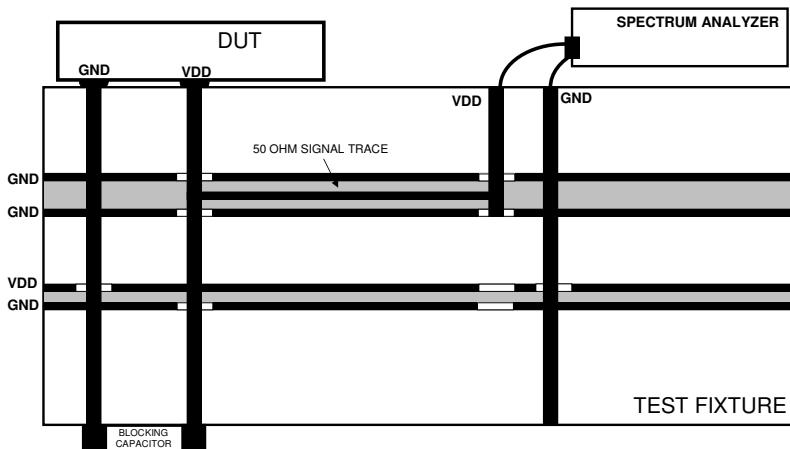


Figure H.18 Measuring the power spectrum of the voltage at the DUT power supply terminal using a $50\ \Omega$ signal trace connected to the DUT power supply via.

References

- [1] J. Moreira, H. Barnes, C. McCowan, and R. Winters, "A Time Domain Reflectometry Kit for ATE Test Fixtures," *Verigy Users Conference*, 2008.
- [2] H. Barnes, J. Moreira, M. Comai, A. Islas, F. Tamayo-Broes, M. Resso, A. Ciccomancini, O. Bell, and M. Tsai, "Performance at the DUT: Techniques for Evaluating the Performance of an ATE System at the Device Under Test Socket," *IEC DesignCon*, 2008.
- [3] M. Resso and E. Bogatin, *Signal Integrity Characterization Techniques*. IEC, 2009.
- [4] H. Barnes, A. Ciccomancini, M. Resso, and M. Tsai, "Differential PCB Structures using Measured TRL Calibration and Simulated Structure De-Embedding," *IEC DesignCon*, 2007.
- [5] J. Moreira and B. Roth, "Characterization and Focus Calibration of ATE Systems for High-Speed Digital Applications," *IEC DesignCon*, 2009.

- [6] I. Novak, "Frequency-Domain Power-Distribution Measurements – An Overview," *IEC DesignCon East*, 2003.
- [7] I. Novak and J. R. Miller, *Frequency-Domain Characterization of Power Distribution Networks*. Artech House, 2007.
- [8] I. Novak, *Power Distribution Network Design Methodologies*. IEC, 2008.
- [9] P. Trivero, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, "Stability, Causality and Passivity in Electric Interconnect Models," *IEEE Transactions on Advanced Packaging*, vol. 30, Nov. 2007.

Jitter Injection Calibration

One of the most complicated challenges for a test engineer is to understand the error associated with a certain measurement setup and how to correlate the setup with other measurement instruments. For measurements like the frequency of a clock signal, it is possible to obtain an accurate estimation of the measurement uncertainty, since the measurement setup calibration uncertainty can be traced to an international recognized standard (e.g., a cesium clock).

As already discussed in Section 5.5.7, correlation of jitter measurements between different instruments and algorithms can be very challenging. One possible solution to this challenge is to analyze the jitter measurement of a given measurement setup or algorithm by using a source with a calibrated amount of jitter.

This appendix presents some techniques to address the challenge of calibrating a stimulus source (e.g., ATE driver) to deliver a specific amount of random and deterministic jitter [1]. The objective is to use calibration techniques that do not depend on a specific instrument (both hardware and proprietary algorithms), but use as much as possible standard measurement instrumentation that itself can be calibrated to known standards.

I.1 Sinusoidal Jitter Injection Calibration

Sinusoidal jitter injection is of key importance for I/O cell characterization, especially in tests like receiver jitter tolerance and transfer. Two methods for sinusoidal jitter injection calibration are presented in this section. The first method, called the J_1/J_0 Bessel Approach, is based on an amplitude ratio measurement between the main carrier of the clock signal with sinusoidal jitter injected and its first phase modulation spectral line. This measurement is

done with a spectrum analyzer and is an established procedure for sinusoidal jitter injection calibration [2, 3].

The second method, called the RJ Subtraction Approach, is based on the assumption that only random jitter is present on the driver when no sinusoidal jitter is injected. This method uses an oscilloscope for performing the calibration measurement in the time domain by subtracting the random jitter value from the measured total jitter value.

I.1.1 The J_1/J_0 Bessel Approach

In this approach we assume that the DUT signal is a clock pattern, and if we ignore the harmonics due to the usual shape of a quasi-square wave bit clock (i.e., assuming that the bit clock is a pure sinusoidal signal), then injecting a sinusoidal jitter signal can be thought of as a simple phase modulating signal. The jittered (i.e., phase modulated) signal would then be described by:

$$V(t) = \sin(2\pi F_{BC}t - \pi J_{AMPL} \cos(2\pi f_J t)) \quad (\text{I.1})$$

where F_{BC} is the frequency of the bit clock for a given data rate (e.g., 3.2 GHz for a 6.4-Gbps data rate). The sinusoidal jitter modulating signal has the expression $J_{AMPL} \cos(2\pi f_J t)$ with f_J the value of the injected sinusoidal jitter frequency and J_{AMPL} the value of the injected sinusoidal jitter amplitude (the peak-to-peak jitter value will be twice the sinusoidal jitter amplitude, i.e., $J_{PK-PK} = 2J_{AMPL}$). One can now define a modulation index M_f that is related to the injected jitter amplitude by the following expression:

$$M_f = \pi J_{AMPL} \quad (\text{I.2})$$

$$J_{AMPL} = \frac{M_f}{\pi} \quad (\text{I.3})$$

The spectrum of the phase modulated clock will consist of a carrier and a theoretically infinite series of sideband pairs. The value of each sideband pair is computed by the Bessel function value for the modulation index (M_f) being used as shown in Figure I.1 for the carrier amplitude J_0 and the first pair of sidebands J_1 .

One important point is that typically one does not have a sinusoidal clock when using a clock pattern but a bit clock (i.e., theoretically the waveform should be a square clock). Figure I.2 shows a comparison of the spectrum of a phase modulated sinusoidal clock with a bit clock. Notice that although the spectrum is different, around the carrier frequency it is easy to distinguish the sidebands due to the phase modulation. Although the presented approach

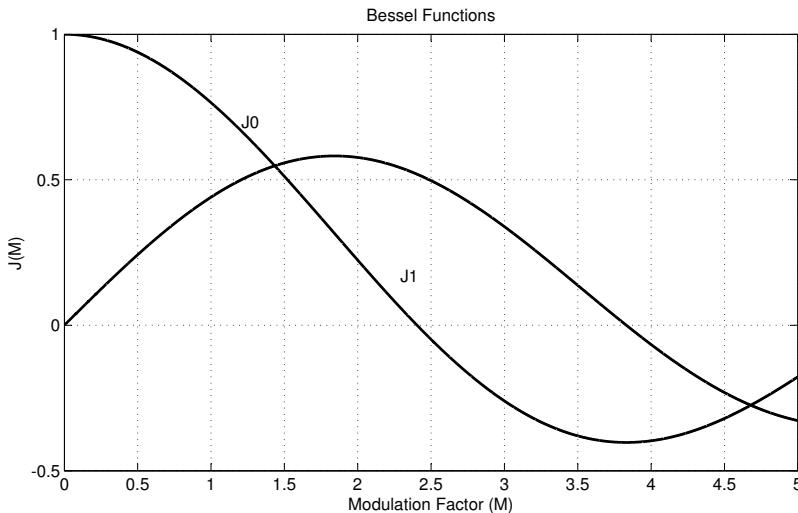


Figure I.1 J_0 and J_1 Bessel function plot.

assumes a sinusoidal clock, for most situations we can apply the method directly to a bit clock signal.

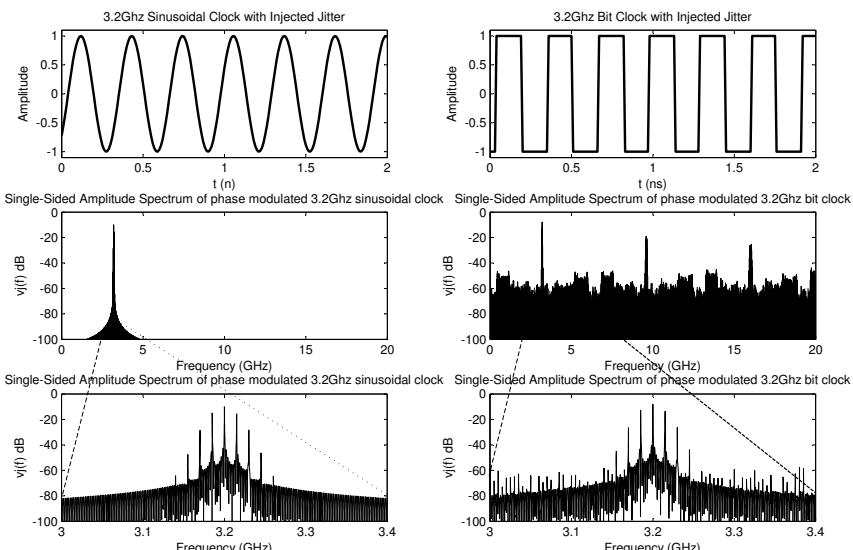


Figure I.2 Comparison of the spectrum of a phase modulated 3.2-GHz sinusoidal clock and a bit clock (15-MHz sinusoidal jitter frequency with 0.25-UI amplitude).

The methodology for calibrating the sinusoidal jitter injection amplitude is then to first compute the needed modulation index from (I.3) and then compute the values of the J_0 and J_1 Bessel functions for that modulation index. With these values we can compute the ratio J_1/J_0 . When calibrating a stimulus source for a given injected sinusoidal jitter amplitude, one needs to change the driver parameters (e.g., the sinusoidal jitter source amplitude) until the desired J_1/J_0 ratio is obtained in the bit clock measured spectrum [2]. Using (I.3) now and a Bessel function table or a mathematical software package (e.g., MATLAB), it is possible to generate a table that provides a J_1/J_0 value for each value of sinusoidal jitter injection amplitude as shown in Table I.1.

Let us now take a real example. Let us suppose we would like to inject 0.25 UI sinusoidal jitter amplitude at 15 MHz on a 6.4-Gbps signal (0.5 UI peak-to-peak sinusoidal jitter). The bit clock frequency for this signal is 3.2 GHz and Table I.1 shows that for 0.25 UI of injected sinusoidal jitter amplitude, the J_1/J_0 ration needs to be 0.426. Figure I.3 shows the jittered bit clock signal measured with an equivalent-time oscilloscope and the signal spectrum measured with a spectrum analyzer where one can easily see the main carrier harmonics from the fact that we are measuring a quasi-square wave bit clock and not a sinusoidal clock.

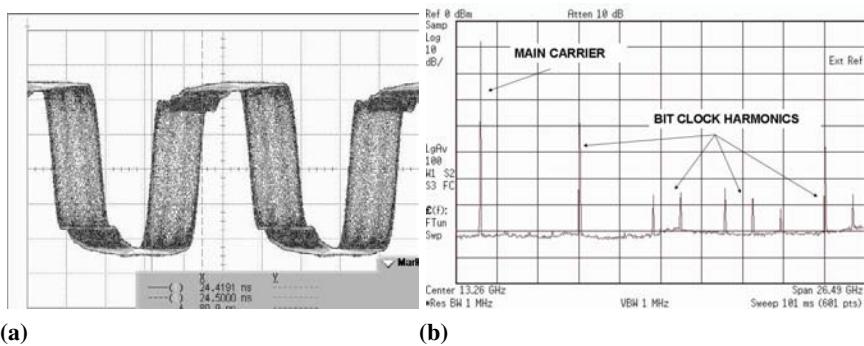


Figure I.3 (a) Waveform of a jittered 6.4-Gbps clock pattern with 15-MHz sinusoidal jitter injection with 0.25-UI amplitude and (b) its spectrum.

Figure I.4 shows the main carrier and the first sidebands in a linear scale so that a measurement of the ratio J_1/J_0 can be made using the spectrum analyzer markers and the marker delta function. The measured value was 42.94% (0.4294) which is very close to the desired value (a ratio value of 42.94 corresponds to a jitter injection amplitude value of 0.251 UI by consulting Table I.1, which would be an approximately 156 fs (0.001 UI)

Table I.1Table for Calibrating the Injected Sinusoidal Jitter Using the J_1/J_0 Method

INJECTED JITTER (UI)	MOD FACTOR	J_0	J_1	J_1/J_0	INJECTED JITTER (UI)	MOD FACTOR	J_0	J_1	J_1/J_0
0.01	0.0314	0.9998	0.0157	0.0157	0.51	1.6022	0.4541	0.5701	1.2554
0.02	0.0628	0.999	0.0314	0.0314	0.52	1.6336	0.4362	0.573	1.3137
0.03	0.0942	0.9978	0.0471	0.0472	0.53	1.665	0.4181	0.5755	1.3763
0.04	0.1257	0.9961	0.0627	0.063	0.54	1.6965	0.4	0.5776	1.4438
0.05	0.1571	0.9938	0.0783	0.0788	0.55	1.7279	0.3819	0.5792	1.5169
0.06	0.1885	0.9911	0.0938	0.0947	0.56	1.7593	0.3636	0.5805	1.5963
0.07	0.2199	0.9879	0.1093	0.1106	0.57	1.7907	0.3454	0.5813	1.6832
0.08	0.2513	0.9843	0.1247	0.1267	0.58	1.8221	0.3271	0.5818	1.7785
0.09	0.2827	0.9801	0.14	0.1428	0.59	1.8535	0.3088	0.5818	1.8839
0.1	0.3142	0.9755	0.1551	0.159	0.6	1.885	0.2906	0.5815	2.0012
0.11	0.3456	0.9704	0.1702	0.1754	0.61	1.9164	0.2723	0.5807	2.1325
0.12	0.377	0.9648	0.1852	0.1919	0.62	1.9478	0.2541	0.5795	2.2809
0.13	0.4084	0.9587	0.2	0.2086	0.63	1.9792	0.2359	0.578	2.4501
0.14	0.4398	0.9522	0.2146	0.2254	0.64	2.0106	0.2178	0.576	2.6451
0.15	0.4712	0.9452	0.2291	0.2424	0.65	2.042	0.1997	0.5737	2.8725
0.16	0.5027	0.9378	0.2435	0.2596	0.66	2.0735	0.1817	0.5709	3.1416
0.17	0.5341	0.93	0.2576	0.277	0.67	2.1049	0.1638	0.5678	3.4654
0.18	0.5655	0.9216	0.2716	0.2947	0.68	2.1363	0.1461	0.5643	3.8632
0.19	0.5969	0.9129	0.2854	0.3126	0.69	2.1677	0.1284	0.5604	4.3645
0.2	0.6283	0.9037	0.2989	0.3308	0.7	2.1991	0.1109	0.5561	5.0164
0.21	0.6597	0.8941	0.3122	0.3492	0.71	2.2305	0.0935	0.5514	5.9005
0.22	0.6912	0.8841	0.3253	0.368	0.72	2.2619	0.0762	0.5464	7.17
0.23	0.7226	0.8737	0.3382	0.3871	0.73	2.2934	0.0591	0.5411	9.1507
0.24	0.754	0.8628	0.3508	0.4066	0.74	2.3248	0.0422	0.5353	12.6798
0.25	0.7854	0.8516	0.3632	0.4265	0.75	2.3562	0.0255	0.5292	20.7583
0.26	0.8168	0.84	0.3753	0.4467	0.76	2.3876	0.009	0.5228	58.2918
0.27	0.8482	0.8281	0.3871	0.4675	0.77	2.419	-0.0074	0.516	-70.207
0.28	0.8796	0.8157	0.3986	0.4887	0.78	2.4504	-0.0235	0.5089	-21.7019
0.29	0.9111	0.803	0.4099	0.5104	0.79	2.4819	-0.0393	0.5015	-12.7533
0.3	0.9425	0.79	0.4208	0.5327	0.8	2.5133	-0.055	0.4938	-8.9844
0.31	0.9739	0.7766	0.4315	0.5556	0.81	2.5447	-0.0703	0.4857	-6.9048
0.32	1.0053	0.7629	0.4418	0.5791	0.82	2.5761	-0.0855	0.4774	-5.5848
0.33	1.0367	0.7488	0.4518	0.6033	0.83	2.6075	-0.1003	0.4687	-4.6714
0.34	1.0681	0.7345	0.4614	0.6283	0.84	2.6389	-0.1149	0.4598	-4.0007
0.35	1.0996	0.7198	0.4708	0.654	0.85	2.6704	-0.1292	0.4506	-3.4866
0.36	1.131	0.7049	0.4798	0.6806	0.86	2.7018	-0.1432	0.4411	-3.0794
0.37	1.1624	0.6897	0.4884	0.7082	0.87	2.7332	-0.1569	0.4313	-2.7483
0.38	1.1938	0.6742	0.4967	0.7367	0.88	2.7646	-0.1703	0.4213	-2.4734
0.39	1.2252	0.6585	0.5046	0.7663	0.89	2.796	-0.1834	0.411	-2.2411
0.4	1.2566	0.6425	0.5122	0.7972	0.9	2.8274	-0.1962	0.4005	-2.042
0.41	1.2881	0.6263	0.5194	0.8293	0.91	2.8588	-0.2086	0.3898	-1.869
0.42	1.3195	0.6099	0.5262	0.8628	0.92	2.8903	-0.2206	0.3789	-1.7171
0.43	1.3509	0.5932	0.5326	0.8978	0.93	2.9217	-0.2324	0.3677	-1.5824
0.44	1.3823	0.5764	0.5387	0.9346	0.94	2.9531	-0.2437	0.3564	-1.462
0.45	1.4137	0.5594	0.5444	0.9731	0.95	2.9845	-0.2548	0.3448	-1.3535
0.46	1.4451	0.5422	0.5497	1.0137	0.96	3.0159	-0.2654	0.3331	-1.255
0.47	1.4765	0.5249	0.5545	1.0565	0.97	3.0473	-0.2757	0.3212	-1.1651
0.48	1.508	0.5074	0.559	1.1018	0.98	3.0788	-0.2856	0.3092	-1.0825
0.49	1.5394	0.4898	0.5631	1.1498	0.99	3.1102	-0.2951	0.297	-1.0063
0.5	1.5708	0.472	0.5668	1.2009	1	3.1416	-0.3042	0.2846	-0.9355

amplitude error compared to the desired 0.25 UI at 6.4 Gbps). Also note that we have not discussed the accuracy of the spectrum measurement since the spectrum analyzer noise floor is expected to be below the needed measurement accuracy for a jitter injection calibration application.

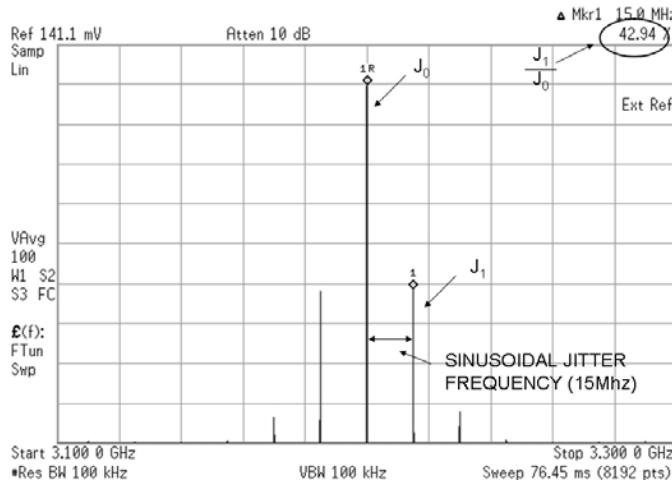


Figure I.4 Spectrum of a jittered 6.4-Gbps clock pattern with 16-MHz sinusoidal jitter injection with 0.25-UI amplitude (0.5 UI peak-to-peak) zoomed around the 3.2-GHz carrier frequency.

The method presented in this subsection is appropriate for low amplitudes of sinusoidal jitter injection (e.g., 1 to 5 UI). For higher amplitudes (e.g., 50 UI) similar methodologies are available, like the null Bessel method, which are more appropriate [2, 3].

I.1.2 The RJ Subtraction Approach

In this approach the main idea is to assume that all the jitter present in the stimulus driver is random for a bit clock pattern. In this case any injected sinusoidal jitter will add on top of the random jitter and since they should be independent processes it is possible to add their variances [4].

Sinusoidal jitter with a peak-to-peak jitter value J_{PK-PK} is described by the following probability distribution [5]:

$$p_{\text{sinusoidal}}(x) = \begin{cases} \frac{1}{\pi \sqrt{\left(\frac{J_{PK-PK}}{2}\right)^2 - x^2}} & |x| < \frac{J_{PK-PK}}{2} \\ 0 & \text{Otherwise} \end{cases} \quad (\text{I.4})$$

Note that the probability distribution is independent of the frequency of the sinusoidal jitter. The variance of the distribution is computed by the following equation:

$$\sigma_{SINUSOIDAL}^2 = \frac{\left(\frac{J_{P_K-P_K}}{2}\right)^2}{2} \quad (I.5)$$

Figure I.5 presents a graph of the probability distribution for a signal with a sinusoidal jitter peak-to-peak value of 4 ps. Note that the sinusoidal jitter amplitude value will be half of the peak-to-peak value for sinusoidal jitter.

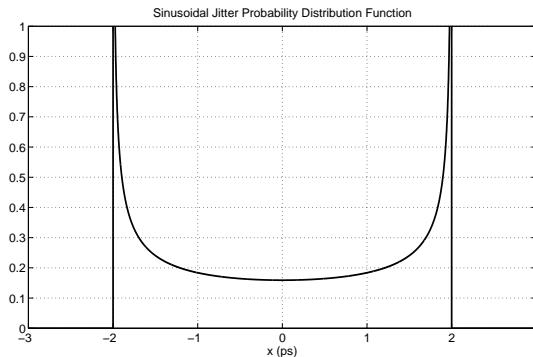


Figure I.5 Sinusoidal jitter probability distribution function ($J_{P_K-P_K} = 4$ ps).

Assuming that there is only random jitter and the injected sinusoidal jitter on the data signal from the stimulus driver, since one typically can consider the random jitter process independent (not correlated) to the sinusoidal jitter injection process, the total variance can then be computed by the square sum of the random and sinusoidal jitter variances.

$$\sigma_{TOTAL}^2 = \sigma_{SINUSOIDAL}^2 + \sigma_{RANDOM}^2 \quad (I.6)$$

Substituting (I.5) into (I.6) and solving it for the sinusoidal peak-to-peak jitter value, we obtain:

$$J_{P_K-P_K} = 2\sqrt{2}\sqrt{(\sigma_{TOTAL}^2 - \sigma_{RANDOM}^2)} \quad (I.7)$$

This means that to compute the value of the injected peak-to-peak sinusoidal jitter amplitude it is necessary to measure the variance of the random jitter and the variance of the total jitter. Figure I.6 shows an example on how (I.7) can be used for sinusoidal jitter injection calibration with a time domain measurement instrument (in this case an equivalent-time

oscilloscope). Since the methodology assumes that when no sinusoidal jitter is injected only random jitter exists on driver output, one needs to use a bit clock pattern to prevent any jitter due to ISI. The driver can have other jitter forms like DCD (duty-cycle distortion) or other forms of periodic jitter. Since test and measurement equipment drivers are typically well designed it is acceptable to assume that the other types of deterministic jitter are very small, although one should always verify this assumption. Figure I.6 (left) shows the measured variance using the histogram function of the oscilloscope when no sinusoidal jitter is injected, and on the right the measured variance when the sinusoidal jitter is injected. In this example 78.125 ps of sinusoidal peak-to-peak jitter at 6.4 Gbps (0.5 UI) was injected at 15 MHz. The injected jitter was calibrated with the J_1/J_0 Bessel approach of Section I.1.1.

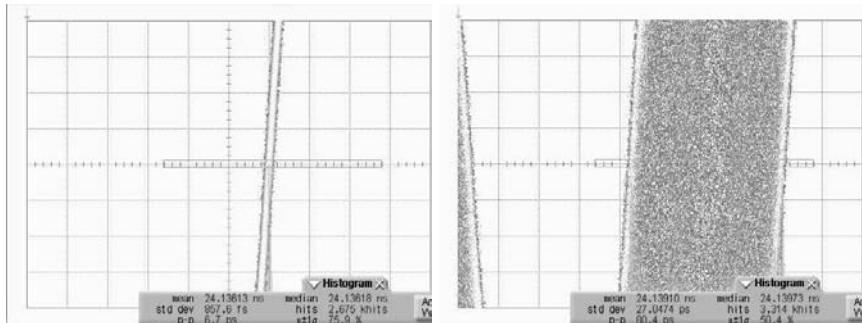


Figure I.6 Measured histogram of the driver output without jitter injection (left) and with sinusoidal jitter injected (right).

From the measurements and using (I.7) the peak-to-peak amplitude of the sinusoidal jitter is:

$$J_{PK-PK} = 2\sqrt{2}\sqrt{(27.05)^2 - (0.86)^2} = 76.47 \text{ ps} = 0.4894 \text{ UI} \quad (\text{I.8})$$

This means that in this specific example, this methodology showed an error of approximately 1.6 ps (78.125 ps - 76.47 ps) for the injected peak-to-peak jitter amplitude compared with the J_1/J_0 Bessel methodology of Section I.1.1.

On the measurements shown in Figure I.6, the measured peak-to-peak total jitter value was 80.4 ps which corresponds to an error of 2.27 ps if the measured peak-to-peak total jitter value was used as the peak-to-peak value of the sinusoidal jitter injected. Note that the error on the total jitter measurement peak-to-peak value is low (2.27 ps). This is because in the presented case, the random jitter of the driver was very low, which means that the measured total

peak-to-peak jitter is very close to the injected sinusoidal peak-to-peak jitter value. Figure I.7 shows another example where the driver has significantly more random jitter and in this case the measured peak-to-peak total jitter value is 105.6 ps (we calibrated a 0.5-UI sinusoidal peak-to-peak jitter injection using the J_1/J_0 Bessel method) which means a 27.5 ps error, while if one uses (I.7) the value is $2\sqrt{2}\sqrt{27.75^2 - 6.3^2} = 76.4$ ps. An error of 1.7 ps is computed from the measurement.

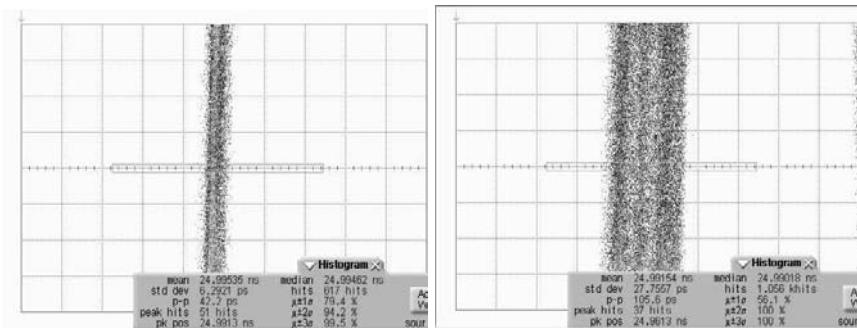


Figure I.7 Measured histogram of the driver output without jitter injection (left) and with sinusoidal jitter injected (right) for a driver with a significant amount of random jitter.

I.2 Random Jitter Injection Calibration

Random jitter injection calibration typically requires the calibration of the standard deviation σ (also known as RMS jitter) of a noise source that injects random jitter into a data signal. It is important to note that in some high-speed standards like PCI Express [6], the standard deviation for random jitter injection is specified for specific frequency bands as shown in Figure I.8.

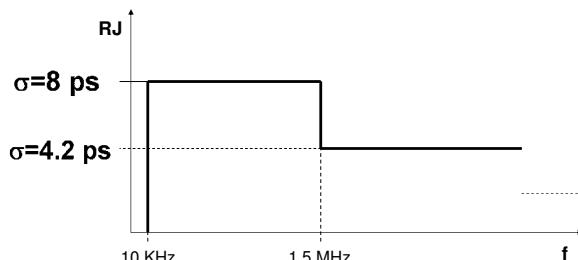


Figure I.8 PCI Express random jitter tolerance requirements [6].

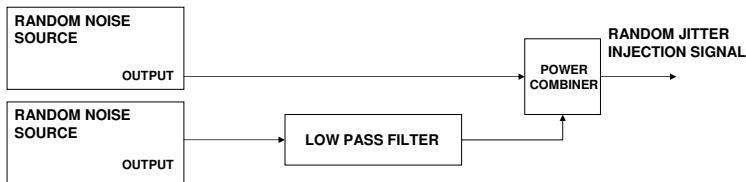


Figure I.9 Creating the random jitter injection profile of Figure I.8 using two random noise sources, a low pass filter, and a power combiner.

This implies the usage of multiple random noise sources with appropriate band limiting filters as shown in Figure I.9. It is important to note that limiting the bandwidth of a Gaussian noise source can have consequences on its properties [7]. Figure I.10 shows the frequency profile of an example implementation of the methodology described in Figure I.9.

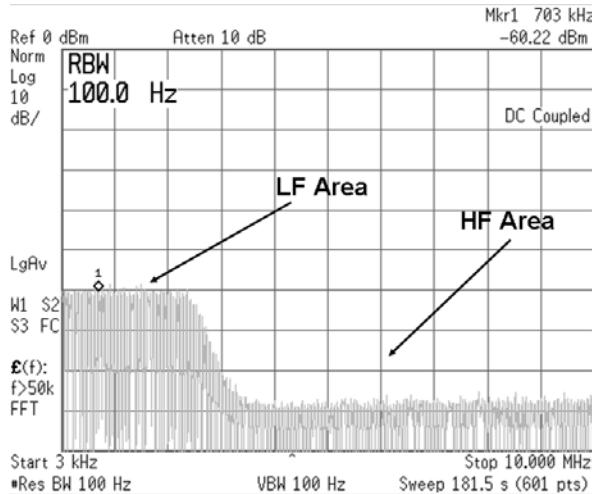


Figure I.10 Example of a measured frequency profile using the random jitter injection methodology of Figure I.9.

Section 7.6 discusses noise sources in more detail. In this section we will assume that one needs to calibrate a random jitter source that has a predefined noise bandwidth to a given standard deviation value (random jitter RMS value). One option is to measure the standard deviation of the data waveform with the random jitter injected using a time domain measurement instrument like an oscilloscope as shown in Figure I.11. In this example a clock pattern was used at the data rate of 6.4 Gbps. The usage of a clock pattern is important since with this approach we can ensure that no data-dependent jitter is present in the signal. The measured standard deviation (RMS jitter) is approximately

5.3 ps. Note that in this example the histogram shows some asymmetries due to the fact that the delay line used for the jitter injection is not completely linear.

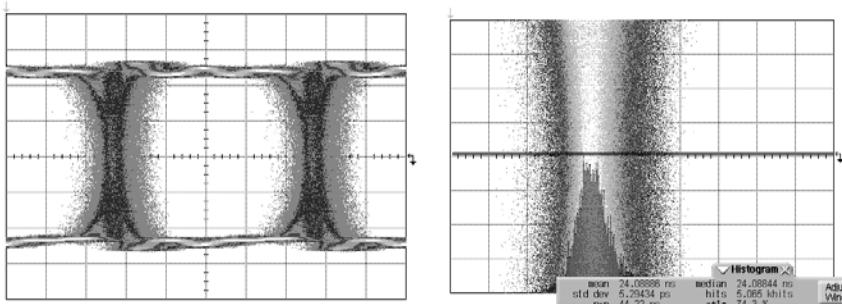


Figure I.11 Measuring the standard deviation (RMS value) of a bit clock signal with random jitter injected.

With the approach described above, the user would tune the amplitude of the noise source until the measured standard deviation corresponds to the desired value of random jitter. The problem with this approach is that a distribution with a given standard deviation does not mean that it corresponds to a Gaussian distribution with that standard deviation value. This is the case when the driver delay line linearity is not able to translate a Gaussian noise distribution perfectly into Gaussian random jitter in the signal. Since noise sources and stimulus sources can vary significantly in terms of quality this is important (see Section 7.6). Another option would be to perform a fit of the histogram data to a Gaussian distribution to infer the standard deviation value. Figure I.12 shows an example of this procedure using the histogram data measured in Figure I.11. The standard deviation of the fitted Gaussian is now approximately 4 ps (compared to 5.3 ps of simply computing the standard deviation of the histogram). Note that verifying that a random distribution is truly Gaussian or which methodology to use for fitting a Gaussian curve to a measured distribution is not trivial [8].

One point that was not discussed is the intrinsic random jitter of the measurement instrument used for measuring the jitter histogram. For an accurate calibration of the injected random jitter in a stimulus driver, this intrinsic jitter should be calibrated out. This is easy if one assumes that the intrinsic jitter present on the measurement instrument is also random with a Gaussian distribution (a reasonable assumption for well-designed measurement instrumentation). In this case the intrinsic jitter from the measurement instrument can be removed from the measured random jitter standard deviation by the following equation:

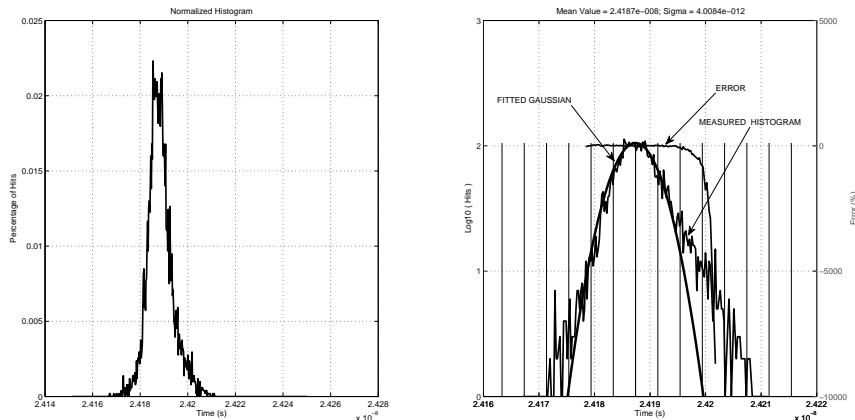


Figure I.12 Fitting a histogram to a Gaussian distribution. Note that a logarithmic scale is used on the right figure (based on the algorithm presented in [9]).

$$\sigma_{RJ} = \sqrt{\sigma_{MEASURED}^2 - \sigma_{INTRINSIC}^2} \quad (I.9)$$

This still leaves the question of how one should measure the intrinsic jitter of a measurement instrument. One option is to use the measurement setup described in Figure I.13. In this setup a very low phase noise RF source is provided to both the trigger input and the measurement input of the instrument. In this case any jitter measured by the instrument is due to the jitter of the RF source which should be very low compared to the expected instrument jitter, and the instrument intrinsic jitter. Figure I.14 presents an example of this approach using the Agilent Technologies 86100C equivalent-time oscilloscope.

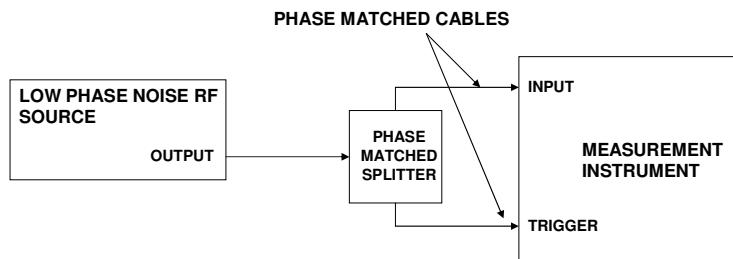


Figure I.13 Measuring the intrinsic jitter of a time-domain measurement instrument (e.g., equivalent-time oscilloscope).

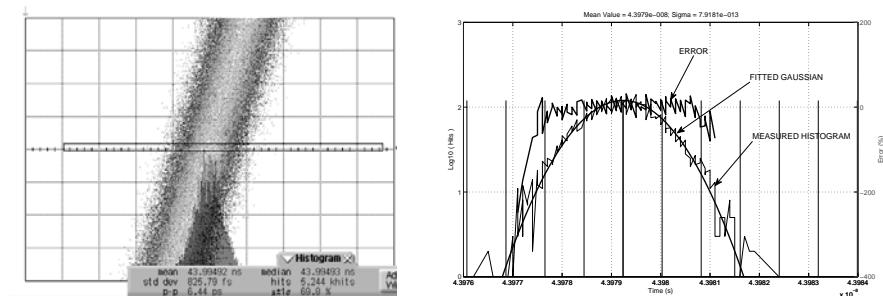


Figure I.14 Measuring the intrinsic jitter of an equivalent-time oscilloscope using a RF source.

I.3 ISI Jitter Injection Calibration

Calibrating data dependent jitter injection in the form of ISI presents significant challenges. ISI can be defined via a filter specification, a lossy PCB signal trace (e.g., 50.8 cm (20 in), 127 μ m (5 mil) trace on FR4), or sometimes by a simple peak-to-peak jitter value.

In standards like PCI Express [6], DDJ injection is defined by a PCB signal trace with a certain geometry on an FR4 type dielectric material. This type of definition creates significant difficulties since it does not provide an exact description of the amount of DDJ that is injected. This can be observed in the example shown in Figure I.15 where the PCB board includes a 127 μ m (5 mil) wide, 50.8 cm (20 in) long microstrip and stripline trace. The measured insertion loss shows a difference between both traces with the microstrip trace showing the worst performance. Although the performance of the microstrip should be better since half of its dielectric is expected to be air, in reality the microstrip plating and solder mask can significantly increase the microstrip loss as shown in Section 8.3.1.

If we now synthesize the data eyes for a 2.5-Gbps PRBS $2^7 - 1$ data pattern for each signal trace using the measured S-parameter data, we obtain the data eyes shown in Figure I.16. The data eyes show that there is a significant difference on the measured injected jitter. Note that on the simulation no random jitter or any other forms of deterministic jitter are included, only ISI jitter from the signal trace loss. This means that although following the guideline of a particular standard, the way the PCB is implemented can have a significant impact on the obtained ISI value. Although this example concentrates on the differences between a microstrip or stripline implementation, similar results could be obtained for differences between the specific FR4 material used. Such issues become more significant when moving to higher data rates.

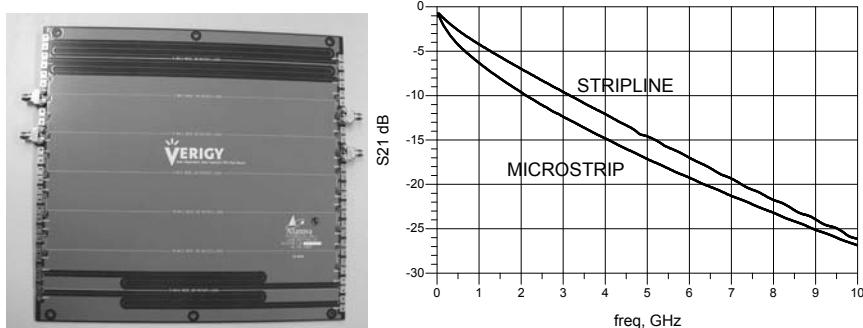


Figure I.15 Comparison of the insertion loss (S_{21}) of a $127\ \mu\text{m}$ (5 mil) wide, 50.8 cm (20 in) long trace implemented as microstrip and stripline on a FR4 dielectric.

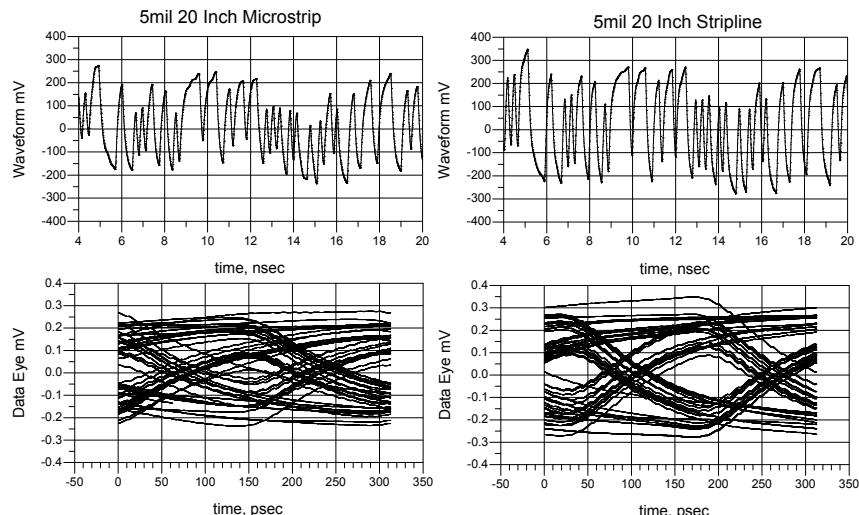


Figure I.16 Comparison of the synthesized data eye at 6.4 Gbps for a $127\ \mu\text{m}$ (5 mil) wide, 50.8 cm (20 in) long microstrip in FR4 (left) and a stripline (right).

This means that for ISI jitter injection, the appropriate way to define and calibrate the amount of jitter injected is to define the response of the lossy signal channel (either a filter or a PCB trace) through the full S-parameter response of the channel (or the time domain step response). From that data it is then possible to compute the corresponding peak-to-peak jitter value for the ISI jitter through simulation.

It is very important to remember that although the injected ISI jitter is defined by a lossy channel, its peak-to-peak value will depend on the spectral properties of the used pattern that are determined, for example, by the PRBS sequence type as shown in Figure I.17.

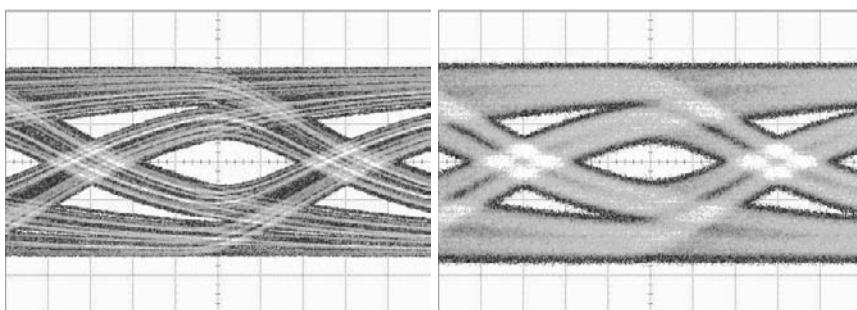


Figure I.17 Influence of the used pattern on the amount of injected DDJ (127 μm (5 mil) wide, 50.8 cm (20 in) long trace on an FR4 dielectric), left: a PRBS $2^7 - 1$ pattern; right: a PRBS $2^{31} - 1$ pattern.

In the case the DDJ is defined as a peak-to-peak value, one approach to generate a PCB signal trace definition is to use a simulation tool to optimize the PCB trace geometry to obtain the desired peak-to-peak DDJ value for a given data pattern. When following this approach it is critical that the simulation models for the PCB trace loss have been verified and optimized with test boards.

References

- [1] J. Stimple and R. Stephens, "Precision Jitter Transmitter," *IEC DesignCon*, 2005.
- [2] Agilent Technologies, "Understanding Jitter and Wander Measurements and Standards," *Application Note 5988-6254EN*, 2003.
- [3] D. Derickson and M. Mueller, *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization*. Prentice-Hall, 2007.
- [4] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.

- [5] W. Maichen, *Digital Timing Measurements*. Springer, 2006.
- [6] PCI-SIG, *PCI Express Base Specification Revision 2.1*, Mar. 2009.
- [7] R. Stephens and M. Mueller, “Analysis of Random Noise and the Effect of Band-Limited Noise on Stressed-Eye Receiver Tolerance Tests,” *IEC DesignCon*, 2009.
- [8] R. Stephens and R. Muro, “Characterization of Gaussian Noise Sources,” *IEC DesignCon*, 2008.
- [9] Tektronix, “Controlled Jitter Generation for Jitter Tolerance and Jitter Transfer Testing,” *Application Note*, 2005.

About the Authors

José Moreira has worked for the last 10 years in the ATE industry where he has held several positions at Agilent Technologies and Verigy. He has supported multiple customers and developed customized hardware and software solutions for ATE characterization and production testing of high-speed digital interfaces. He has also submitted several patents and published multiple papers in the areas of ATE high-speed digital testing, jitter testing, test fixture design, and focus calibration. Mr. Moreira received a master of science degree in electrical and computer engineering from the Instituto Superior Técnico of the Technical University of Lisbon and he is currently a senior engineer at Verigy R&D in Boeblingen, Germany. He is a senior member of the IEEE.

Hubert Werkmann is currently employed as principal consultant at Verigy Germany GmbH, where he works on ATE test solution development for high-speed I/O interfaces and high-speed memory devices in combination with the deployment of these test solutions to leading semiconductor manufacturers. He has 20 years of experience in the semiconductor design and test industry with various positions at IMS Chips, Agilent Technologies, and Verigy. Dr. Werkmann has multiple published papers in the areas of scan-based design for testability and ATE high-speed testing of computation and memory devices. Dr. Werkmann received a Diploma degree in computer science from the University Stuttgart, Germany, and holds a Ph.D. in engineering from the University Stuttgart.

Index

- 8B/10B coding, 478
- Accelerated graphics port (AGP), 56
- Accessories, 298
- Accuracy, 391
- Active probe, 305
- Address bus inversion (ABI), 108, 112
- Algorithm-based pattern, 240
- Algorithmic pattern generation (APG), 137, 171
- Aliquot, 154
- Amplitude noise, 34, 208
- Analog power supplies, 160
- Antialiasing filter, 151
- Application programming interface (API), 154
- Arbitrary function generator, 277
- Arbitrary waveform generator, 277
- Arming
 - latency, 190
- At-cycle source synchronous interfaces, 40
- Attenuators, 299
- Automated test equipment (ATE), 2, 133
 - calibration, 396
 - testhead, 133
- Balanced unbalanced transformer, 307
- Balun, 307
- Bandwidth, 18
- Bench instrumentation, 269
- Bessel functions, 544
- BGA ballout, 342
- Bidirectional, 144
- Bidirectional interface, 42, 354, 409
- BIP-N, 485
- Bit alignment, 165
- Bit clock, 544
- Bit error rate (BER), 21
 - bathub curve, 185
 - contour plot, 210
 - function, 453
- Bit error rate tester (BERT), 274
- Bit period, 11
- Bit uncertainty, 163
- Bits per second, 55
- Blocking capacitors, 299
- Built-in self-test (BIST), 249
- Bus inversion, 108
- Bypass capacitors, 375
- Capacitor
 - arrays, 380
 - ceramic, 380
 - controlled ESR, 380
 - low inductance chip, 380
 - tantalum, 380
- Central limit theorem, 449
- Channel assemblies, 134
- Channel multiplexing, 249, 262, 399
- Checksum, 485
- Clamshell mode, 111
- Clock data recovery (CDR), 42, 43, 46, 139, 414
- Coaxial cables, 287
- Coaxial connectors, 293

- Coaxial terminations, 299
- Comma character, 481
- Comma symbol, 481
- Common clock interfaces, 39
- Complementary error function, 458
- Contact
 - cantilever spring, 348
 - crown, 348
 - elastomer, 348
 - fuzz button, 348
 - spring pin, 348
 - tweezer, 348
- Continuous jitter test pattern (CJPAT), 472
- Continuous random test pattern (CRPAT), 472
- Continuous time linear equalizer (CTLE), 50
- Correlation, 5
- Cost of test (COT), 237
- CRC-8-ATM, 487
- Crest factor, 279
- Crosstalk, 213
- Cumulative density function (CDF), 451
- Cyclic redundancy check (CRC), 108, 486
- Data bus inversion (DBI), 108, 112
- Data eye
 - diagram, 11, 176, 273
 - height, 181
 - height calibration, 402
 - mask, 183
 - width, 180
- Data eye profile, 406
 - correlation, 408
- DC blocks, 299
- DCA, 270
- DDJ Injection, 438
- De-embedding, 423
- De-emphasis, 50, 191
- Decision feedback equalization (DFE), 430
- Deconvolution, 423
- Decoupling capacitors, 375
- Delay line, 302
- Delay locked loop (DLL), 43, 46
- Design for test (DfT), 2, 4
- Device interface board (DIB), 5, 311
- Device power supplies (DPS), 160, 364
- Dielectric loss, 320
- Dielectric material, 322
- Differential signaling, 12
- Digital power supplies, 160
- Digitizer, 150
- Dirac delta function, 462
- Disparity, 476
- Double data clocking (DDC), 246
- Double data rate (DDR), 107
- Double-edge response (DER), 408
- Driver
 - sharing, 263
 - skew, 188
- Dual Dirac jitter model, 199, 461
- Dual transmission line, 138, 354
- Dual transmission line (DTL), 410
- DUT loadboard, 5
- EDC hold pattern, 113
- Edge placement accuracy (EPA), 392
- Edge termination, 375
- Electrical idle, 258
- Electrostatic discharge, 284
- Embedded clock interfaces, 42
- Equalization, 140, 430
- Equivalent series
 - inductance, 376
 - resistance, 376
- Equivalent time sampling oscilloscope, 270
- Erf, 458
- Erfc, 458
- Error density, 167
- Error detection, 484
- Error function, 458
- Errors
 - random, 233
 - systematic, 233
- ESD, 284
- ESL, 376
- ESR, 376
- Extreme data rate DRAM, 93
- Fall time measurement, 175, 244
- Far-end crosstalk (FEXT), 499
- Far-end loopback, 241
- Fast data eye mask, 242
- Feed-forward equalizer (FFE), 50
- Flexible-pattern, 240
- FlexPhase, 94
- Fly-by, 138, 354, 410
- Fly-by termination, 102

- Focus calibration, 401
Forwarded clock interfaces, 41
Frequency offset, 223
Frequency synchronization, 285
Front side bus (FSB), 74
Gaussian probability distribution, 449
GDDR SRAM, 107
Gerber generation, 521
Golden device, 238
GPU, 107
Ground plane, 370
Hand-made coaxial probe, 305
Hifix, 314
High-definition multimedia interface (HDMI), 128
Histogram, 25, 192
Hold time, 211
HyperTransport, 73
IDD, 160
IDQ, 160
Impedance
 common mode, 230
 differential, 230
 measurement, 493
 mismatches, 316
 single-ended, 230
 test, 227
Impedance test, 227
Insertion gain, 503
Insertion loss, 498, 503
Instrument synchronization, 284
Interleaved parity, 485
Internal loopback mode (ILM), 85, 91
Intrinsic random jitter, 553
Inverse complementary error function, 459
ISI jitter injection calibration, 555
 J_1/J_0 Bessel method, 544
Jitter, 23
 absolute, 24
 bounded uncorrelated (BUJ), 34
 data dependent (DDJ), 29, 143
 deterministic, 461
 deterministic (DJ), 29
 duty-cycle distortion (DCD), 29
 F/2, 29
 generation, 414
 histogram, 25, 192
 injection calibration, 404, 543
 intersymbol interference (ISI), 32
 periodic (PJ), 29
 random, 461
 random (RJ), 27
 spectrum, 36
 time interval error (TIE), 24
 tolerance, 214
 total, 187
 transfer, 221
 wander, 38
K28.5 character, 472
K28.5 pattern, 472
Known good die (KGD), 356
Linear feedback shift register (LFSR), 170, 469, 482
Loadboard
 active, 261
Loop delay, 251
Loopback
 active, 257
 near-end, 252
 near-end analog, 252
 near-end digital, 252
 near-end internal, 252
 parametric, 257
 passive straight, 253
 passive stressed, 256
 wire, 253
Loopback testing, 249
 at-speed, 4
Loss tangent, 320
Match-loop, 168
Mean value, 451
Measurement probes, 302
Measuring crosstalk, 498
Measuring data dependent jitter, 204
Microstrip, 330
Mil, 318
Mode register (MR), 115
Multisite efficiency, 262
Multisite testing, 262
Near-end crosstalk (NEXT), 499
Netlist, 312
Noise generators, 279

- Nonfunctional pads, 335
- Nonreturn to zero (NRZ), 9
- Normal probability distribution, 449
- Null bessel method, 548
- Nyquist
 - sampling limit, 150
- Octal data rate (ODR), 94
- On-die termination (ODT), 103, 111, 117
- Out-of-band signal, 11
- Overall time accuracy (OTA), 392
- Overequalization, 438
- Parity bit, 485
- Passive probes, 304
- Pattern
 - alignment, 168
- Pattern generator
 - algorithm, 240
 - flexible, 240
- Pattern transition density, 49
- Pattern uncertainty, 164
- Pattern vector, 171
- PCB layout viewer, 521
- PCI express
 - compliance pattern, 472
 - LTSSM, 62
- Peak-to-peak jitter, 194
- Performance verification (PV), 396
- Period jitter, 24
- Phase locked loop (PLL), 43
- Phase modulation, 544
- Phase noise, 193
- Pick-off T, 301
- Pin electronics cards, 133
- PMU, 227
- POD15, 122, 441
- Pogo pin assembly, 136
 - bench, 527
- Pogo tower, 356
- Population standard deviation, 452
- Power combiner, 298
- Power distribution network
 - design, 384
 - measurement (PDN), 535
 - simulation, 384
- Power distribution network (PDN), 363
- Power distribution system (PDS), 363
- Power divider, 298
- Power plane, 370
- Power splitter, 298
- Power supplies, 160
- Pre-emphasis, 50, 140, 191
- Probability density function (PDF), 450
- Probe card, 356
- Probe positioner, 306
- Production testing, 237
- Pseudo open drain (POD), 116
- Pseudo-random bit sequence (PRBS), 469
- Pseudo-random word sequences (PRWS), 470
- Pulse amplitude modulation (PAM), 9
- Q-factor, 465
- Rambus signaling level (RSL), 102
- Rambus signaling level differential (DRSL), 102
- Random jitter injection calibration, 551
- Random jitter tolerance, 216
- Real-time oscilloscope, 269
- Receiver hold measurement, 211
- Receiver sensitivity, 212
 - crosstalk, 213
- Receiver setup measurement, 211
- Repeatability, 392
- Reproducibility, 392
- Reshuffling, 154
- Resolution, 391
- Retiming, 259
- Return loss, 231, 495, 503
 - Common mode, 232
 - differential, 232
- Return to zero (RZ), 9
- Rise time converter, 308
- Rise time measurement, 175, 244
- RJ subtraction approach, 548
- RJ/DJ separation, 465
- Root mean square (RMS)
 - jitter, 193
 - value, 451
- Round-trip delay, 494
- Runlength, 478
- Running disparity, 477
- S-parameters, 277, 503
- S11, 495, 498
- Sample standard deviation, 452
- Sampler, 150

- SATA, 127
SATA clock, 225
SBD, 145, 414
Scattering parameters, 277, 503
Scrambling, 481
Self-impedance, 368
Serial ATA, 127
Setup and hold time measurement, 243
Setup time, 211
Shannon's theorem, 150
Shmoo
 fast-shmoo, 173
 overlay plot, 177
 shmoo test, 172
 shmoo-plot, 173
 tracking-shmoo, 173
Sideband signal, 10
Signal generators, 280
Simulation
 2D EM field solver, 518
 3D EM field solver, 518
 circuit simulator, 515
 impedance calculator, 522
 model generation, 521
 power distribution network, 520
Simultaneous bidirectional, 144, 145, 414
Sinc() function, 17
Sinusoidal
 clock, 544
 clock sources, 280
 jitter injection, 543
 jitter injection calibration, 544
 jitter tolerance, 216
Skew calibration, 298, 402
Skin depth, 319
Socket, 346
 contact resistance, 347
 electrical performance, 347
 insertion loss, 349
 maximum current, 349
 pin inductance, 383
 pin-to-pin capacitance, 349
 self-inductance, 349
Source synchronous, 39
Spectrum analyzer, 276
Spread spectrum clocking (SSC), 61, 224
Stack-up, 359
Stacked capacitors, 382
Standard deviation, 193, 451
Stripline, 330
Surface roughness, 320
Synthesized data eye diagram, 498
T-matrix, 510
Termination
 parallel, 16
 resistors, 15
 series, 15
 transmission line, 14
Test coupon, 527
Test fixture, 5, 311
 active, 261
 performance, 525
Test vector, 171
Testing multilevel interfaces, 421
Time domain reflectometry (TDR), 491
Time domain transmission (TDT), 491, 497
Time interval analyzer, 140
Time interval analyzer (TIA), 189, 275
Time stamper, 140
Transfer-impedance, 368
Transfers per second, 55
Transition density, 454
Transition time, 308
 converter (TTC), 308
 measurement, 244
Transmitter
 skew, 188
Trimming, 396
Trombone delay line, 302
Undersampling, 154
Unidirectional interfaces, 42
Unit interval (UI), 11
Unit test period (UTP), 154
Universal serial bus (USB), 128
Variance, 451
Vector network analyzer (VNA), 277
Voltage controlled oscillator (VCO), 44
Wafer
 probe interface, 356
 probing, 356
Wander, 38
Wire loopback
 DC testing, 254
 scan testing, 254

XDR, 93	SC, 94
DRAM, 93, 94	subcolumn, 94
dynamic point-to-point (DPP), 97	XIO, 94
IO, 94	
memory controller, 94	XMC, 94

Recent Titles in the Artech House Microwave Library

Active Filters for Integrated-Circuit Applications, Fred H. Irons

Advanced Techniques in RF Power Amplifier Design, Steve C. Cripps

Automated Smith Chart, Version 4.0: Software and User's Manual,
Leonard M. Schwab

Behavioral Modeling of Nonlinear RF and Microwave Devices,
Thomas R. Turlington

Broadband Microwave Amplifiers, Bal S. Virdee, Avtar S. Virdee, and
Ben Y. Banyamin

Computer-Aided Analysis of Nonlinear Microwave Circuits,
Paulo J. C. Rodrigues

Designing Bipolar Transistor Radio Frequency Integrated Circuits,
Allen A. Sweet

Design of FET Frequency Multipliers and Harmonic Oscillators,
Edmar Camargo

Design of Linear RF Outphasing Power Amplifiers, Xuejun Zhang,
Lawrence E. Larson, and Peter M. Asbeck

Design Methodology for RF CMOS Phase Locked Loops,
Carlos Quemada, Guillermo Bistué, and Iñigo Adin

*Design of RF and Microwave Amplifiers and Oscillators, Second
Edition*, Pieter L. D. Abrie

Digital Filter Design Solutions, Jolyon M. De Freitas

*Discrete Oscillator Design Linear, Nonlinear, Transient, and Noise
Domains*, Randall W. Rhea

Distortion in RF Power Amplifiers, Joel Vuolevi and Timo Rahkonen

*EMPLAN: Electromagnetic Analysis of Printed Structures in Planarly
Layered Media, Software and User's Manual*, Noyan Kinayman
and M. I. Aksun

*An Engineer's Guide to Automated Testing of High-Speed
Interfaces*, José Moreira and Hubert Werkmann

Essentials of RF and Microwave Grounding, Eric Holzman

FAST: Fast Amplifier Synthesis Tool—Software and User's Guide,
Dale D. Henkes

Feedforward Linear Power Amplifiers, Nick Pothecary

Foundations of Oscillator Circuit Design, Guillermo Gonzalez

*Fundamentals of Nonlinear Behavioral Modeling for RF and
Microwave Design*, John Wood and David E. Root, editors

Generalized Filter Design by Computer Optimization, Djuradj
Budimir

High-Linearity RF Amplifier Design, Peter B. Kenington

High-Speed Circuit Board Signal Integrity, Stephen C. Thierauf

Intermodulation Distortion in Microwave and Wireless Circuits,
José Carlos Pedro and Nuno Borges Carvalho

Introduction to Modeling HBTs, Matthias Rudolph

Lumped Elements for RF and Microwave Circuits, Inder Bahl

Lumped Element Quadrature Hybrids, David Andrews

Microwave Circuit Modeling Using Electromagnetic Field Simulation,
Daniel G. Swanson, Jr. and Wolfgang J. R. Hoefer

Microwave Component Mechanics, Harri Eskelinen and
Pekka Eskelinen

*Microwave Differential Circuit Design Using Mixed-Mode
S-Parameters*, William R. Eisenstadt, Robert Stengel, and
Bruce M. Thompson

Microwave Engineers' Handbook, Two Volumes,
Theodore Saad, editor

*Microwave Filters, Impedance-Matching Networks, and Coupling
Structures*, George L. Matthaei, Leo Young, and E.M.T. Jones

Microwave Materials and Fabrication Techniques, Second Edition,
Thomas S. Laverghetta

Microwave Mixers, Second Edition, Stephen A. Maas

Microwave Radio Transmission Design Guide, Second Edition,
Trevor Manning

Microwaves and Wireless Simplified, Third Edition,
Thomas S. Laverghetta

Modern Microwave Circuits, Noyan Kinayman and M. I. Aksun

Modern Microwave Measurements and Techniques, Second Edition,
Thomas S. Laverghetta

Neural Networks for RF and Microwave Design, Q. J. Zhang and
K. C. Gupta

Noise in Linear and Nonlinear Circuits, Stephen A. Maas

Nonlinear Microwave and RF Circuits, Second Edition,
Stephen A. Maas

*QMATCH: Lumped-Element Impedance Matching, Software and
User's Guide*, Pieter L. D. Abrie

Practical Analog and Digital Filter Design, Les Thede

Practical Microstrip Design and Applications, Günter Kompa

*Practical RF Circuit Design for Modern Wireless Systems, Volume I:
Passive Circuits and Systems*, Les Besser and Rowan Gilmore

*Practical RF Circuit Design for Modern Wireless Systems, Volume II:
Active Circuits and Systems*, Rowan Gilmore and Les Besser

*Production Testing of RF and System-on-a-Chip Devices for Wireless
Communications*, Keith B. Schaub and Joe Kelly

Radio Frequency Integrated Circuit Design, Second Edition,
John W. M. Rogers and Calvin Plett

RF Bulk Acoustic Wave Filters for Communications,
Ken-ya Hashimoto

RF Design Guide: Systems, Circuits, and Equations, Peter Vizmuller

RF Measurements of Die and Packages, Scott A. Wartenberg

The RF and Microwave Circuit Design Handbook, Stephen A. Maas

RF and Microwave Coupled-Line Circuits, Rajesh Mongia, Inder Bahl,
and Prakash Bhartia

RF and Microwave Oscillator Design, Michal Odyniec, editor

RF Power Amplifiers for Wireless Communications, Second Edition,
Steve C. Cripps

RF Systems, Components, and Circuits Handbook, Ferril A. Losee

The Six-Port Technique with Microwave and Wireless Applications,
Fadhel M. Ghannouchi and Abbas Mohammadi

Solid-State Microwave High-Power Amplifiers, Franco Sechi and
Marina Bujatti

Stability Analysis of Nonlinear Microwave Circuits, Almudena Suárez
and Raymond Quéré

Substrate Noise Coupling in Analog/RF Circuits, Stephane Bronckers,
Geert Van der Plas, Gerd Vandersteen, and Yves Rolain

System-in-Package RF Design and Applications, Michael P. Gaynor

*TRAVIS 2.0: Transmission Line Visualization Software and User's
Guide, Version 2.0*, Robert G. Kaires and Barton T. Hickman

Understanding Microwave Heating Cavities, Tse V. Chow Ting Chan
and Howard C. Reader

For further information on these and other Artech House titles, including previously considered out-of-print books now available through our In-Print- Forever® (IPF®) program, contact:

Artech House Publishers
685 Canton Street
Norwood, MA 02062
Phone: 781-769-9750
Fax: 781-769-6334
e-mail: artech@artechhouse.com
artech-uk@artechhouse.com

Artech House Books
16 Sussex Street
London SW1V 4RW UK
Phone: +44 (0)20 7596 8750
Fax: +44 (0)20 7630 0166
e-mail:

Find us on the World Wide Web at: www.artechhouse.com
