Efficient Test Methodologies for High-Speed Serial Links

Lecture Notes in Electrical Engineering

Volume 51

For further volumes
http:/www.springer.com/series/7818

Dongwoo Hong • Kwang-Ting Cheng

Efficient Test Methodologies for High-Speed Serial Links

Springer

Dr. Dongwoo Hong
Broadcom Corporation
5300 California Ave.
Irvine CA 92617
USA
dwhong@broadcom.com

Prof. Kwang-Ting Cheng
University of California
Santa Barbara College of Engineering
Dept. Electrical & Computer Engineering
Santa Barbara CA 93106-9560
USA
timcheng@ece.ucsb.edu

ISSN 1876-1100 e-ISSN 1876-1119
ISBN 978-90-481-3442-7 e-ISBN 978-90-481-3443-4
DOI 10.1007/978-90-481-3443-4
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009937220

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to Joomi, my wife.

Preface

With the increasing demand for higher data bandwidth, communication systems' data rates have reached the multi-gigahertz range and even beyond. Advances in semiconductor technologies have accelerated the adoption of high-speed serial interfaces, such as PCI-Express, Serial-ATA, and XAUI, in order to mitigate the high pin-count and the data-channel skewing problems. However, with the increasing number of I/O pins and greater data rates, significant challenges arise for testing high-speed interfaces in terms of test cost and quality, especially in high volume manufacturing (HVM) environments.

This dissertation proposes novel testing techniques for high-speed interfaces, which can significantly reduce testing costs while maintaining high test coverage. The primary focus is on efficient testing methods for jitter and bit-error-rate (BER), which are widely used to represent the quality of a communication system. The reader is assumed to have a basic understanding of high-speed I/O and its test methodologies.

There are eight chapters in this dissertation. Chapter 1 gives a general introduction to high-speed I/O systems and describes commonly used testing methods along with their limitations.

Chapter 2 presents an efficient jitter measurement technique using fast comparator sampling. The comparator under-sampling technique is discussed, followed by the efficient jitter calculation algorithm. To validate the accuracy of the technique, the experimental setup and results using a high-speed sampling prototype and various types of measurement instruments are presented.

The BER estimation technique for high-speed serial links that incorporate the linear clock and data recovery (CDR) circuit is described in Chapter 3. The jitter transfer characteristics of the linear CDR loop are analyzed based on the conventional phase-locked loop (PLL) theory. This chapter then describes how the input jitter and the CDR circuit's internal jitter affect the recovered clock jitter and the dependency of the BER on the characteristics of the CDR circuit. The BER estimation technique is extended to the serial links which incorporate the bang-bang (BB) CDR circuit. This method is explained in Chapter 4. Due to the highly non-linear characteristics of the BB CDR loop, the jitter transfer function of the loop strongly varies depending on the input jitter magnitude. Thus, the loop's dependency on the jitter magnitude is fully characterized for accurate BER estimation.

Chapter 5 first reviews the basic concepts of the timing margining test, which is a widely adopted design-for-test (DFT) technique. Then, this chapter describes four possible gaps that the timing margining test might have compared to conventional jitter testing. Chapter 6 deals with the total jitter (TJ) estimation technique for improving the quality of the timing margining test, which can accurately predict the TJ at a very low BER level using only the information from the higher BER region. The limitations of the existing TJ estimation method, which is based on the dual-Dirac model, is described, and then a high-order polynomial fitting technique is proposed to overcome the limitations.

A two-tone testing method for continuous-time adaptive equalizers is described in Chapter 7. This chapter starts with an introduction to the continuous-time adaptive equalizers. Then, the proposed two-tone testing method is described followed by the transistor-level implementation details of the technique.

Chapter 8 concludes the dissertation by summarizing the results of the research conducted and discussing possible future research areas.

This research could not be accomplished without guidance and support of professors, colleagues, family members, and friends. First of all, I would like to thank my advisor, Professor Tim Cheng, for his endless support and great advice during the challenging periods. I feel grateful and privileged to have worked with him. I would also like to thank a number of professors and professionals who reviewed the dissertation and gave me helpful feedback: Professor Forrest D. Brewer, Professor Stephen I. Long, Professor Patrick Yue, and Dr. Mike P. Li.

I would like to acknowledge Teradyne for providing test equipment and facilities for experiments. Especially, I would like to thank Cameron Dryden, Michael Panis, Jacob Scherb, and Wolfgang Maichen for their assistance and feedback. I would like to thank Anne Meixner, Benoit Provost, James Jaussi, and Bryan Casper of Intel. Their great technical expertise always inspired me and helped me to solve several challenging problems on a number of occasions.

I would like thank Linda Dailey Paulson for her attentive help in improving my writing skill throughout the writing process of this dissertation. I would like to express my sincere gratitude towards my parents for their support and encouragement during this project. Finally, I give my warmest thanks to my wife, Joomi Kim. Her love and continual encouragement have been a great source of strength that allowed me to complete this work.

Contents

Chapter 1
Introduction

1.1 Overview of High-Speed Serial Links

As devices become increasingly complex and faster, high-speed serial interfaces such as PCI-Express, SerialATA, XAUI and others are proliferating. Several trends in semiconductor technologies accelerate the adoption of serial interfaces, in order to mitigate the high pin-count and the data-channel skewing problems. In this section, we describe the basics of operation and test of high-speed serial links.

1.1.1 High-Speed Serial Link System

High-speed serial links are composed of a transmitter (TX) and a receiver (RX) communicating over a channel. Figure 1.1 shows the typical block diagram of a transceiver for high-speed serial links. Due to the limited number of I/O pins in a chip and density constraints on the number of wires between the chips, the links usually convert parallel data to serial one using a serializer before transmitting the data. In the receiver side, this serial data is reconverted to the original parallel data using a deserializer. A clock and data recovery (CDR) circuit in the receiver extracts the clock information from the data to synchronize the receiver with the incoming data because, in serial communication systems, the clock signal is embedded in the data. Thus, the CDR circuit plays a critical role in determining the quality of serial communication systems, including influencing metrics such as bit error rate (BER). As the data rates continue to increase and approach speed of multi-gigabits/second the signal is distorted by the bandwidth limitation of the channel. In order to compensate for channel loss, a pre-emphasis at the TX and an equalizer at the RX are implemented in the system. In addition, a simple pattern generator and an error detector are found in most transceiver designs for testing purposes.

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_1,

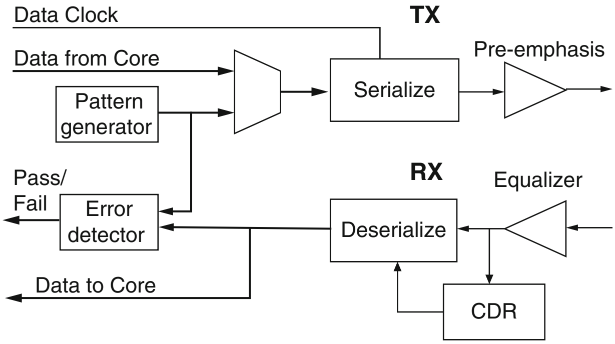


Fig. 1.1 Block diagram of high-speed transceiver

1.1.2 Testing High-Speed Serial Links

BER and jitter have been widely used as measurements to ensure the performance of high-speed communication systems. The BER represents the ratio of the number of bits received in error to the total number of bits transmitted. Jitter is defined as the deviation of a signal event from its ideal position in time. Total jitter (TJ) is typically divided into two categories – deterministic jitter (DJ) and random jitter (RJ). DJ is further divided into periodic jitter (PJ) (also known as Sinusoidal Jitter (SJ)), data-dependent jitter (DDJ), and bounded uncorrelated jitter (BUJ). The characteristics and sources of different kinds of jitter are summarized as follows [1–3].

Random jitter (RJ) – RJ comes from device noise sources, which include shot noise, flicker noise and thermal noise. It is commonly modeled by the Gaussian distribution function.

Periodic jitter (PJ) – PJ is typically caused by external deterministic noise sources coupling into a system, such as switching power supply noises or strong local RF carriers. It may also caused by an unstable CDR circuit.

Data-dependent jitter (DDJ) – DDJ is correlated with the bit sequence in a data stream. DDJ is mainly caused by the duty-cycle distortion (DCD) and inter-symbol interference (ISI). The slew rate differences between the rising and falling signal edges and the voltage offsets between the differential inputs generate the DCD, while the bandwidth limitation of the transmission channel causes the ISI.

Bounded uncorrelated jitter (BUJ) – BUJ is typically due to coupling from adjacent lines (i.e. crosstalk) or on-chip random logic switching. It is bounded, and its characteristics depend on the data pattern, coupling signal, and coupling mechanism.

Jitter or BER can be measured using a variety of methods, either using external equipment or on-chip circuitry. The external instruments include oscilloscope, spectrum analyzer, time interval analyzer (TIA), and bit-error-rate tester (BERT) [1,4]. A TIA measures many single-shot edge-to-edge time intervals on a non-continuous and random basis. The statistics of these measurements can be used to perform the total jitter calculation. BERT is composed of a pattern generator and an error detector. In order to obtain the amount of eye closure as a function of BER, the BERT can vary the clock edge placement with respect to the data edge for BER measurement, which is called the BERT scan technique. Various on-chip measurement techniques have been proposed [5–9]. For jitter measurement, a ring oscillator and a calibration circuit are used in [5]. This technique is capable of measuring jitter with a resolution as low as a single gate delay in the given process. In [6, 7], researchers have proposed novel ways of measuring jitter with a resolution below a single gate delay either by applying an undersampling technique or a Vernier delay line. In [8, 9], the BER or timing margin is measured by simply providing a loopback path from the transmitter to its own receiver. A transceiver includes a pattern generator and an error-detector for BER measurement, and has features to control the clock phase in order to adjust the point in the eye from which data samples are taken for margining test.

1.2 Challenges in Testing High-Speed Serial Links

With the increasing number of I/O pins and greater data rates, testing high-speed interfaces has posed significant challenges in terms of test cost and quality. In this section, we describe a few main challenges in testing these high-speed communication links.

Currently available jitter measurement techniques either require expensive measurement instruments or else they do not guarantee sufficient test quality. Although jitter measurements using external instruments can be performed in the lab for characterization, it is not appropriate for high volume production due to its cost and scalability limitations. Some instruments, such as the oscilloscope and spectrum analyzer, do not lend themselves to fast parallel testing of devices with a large number of high-speed interfaces due to their hardware complexity. In addition, current on-chip jitter measurement circuits do not have sufficient resolution [5, 7], or cannot separate various jitter components [6].

BER measurement down to the 10^{-12} level, which is required to ensure system reliability for most multi-gigabit communication standards, is test time prohibitive. In order to guarantee 10^{-12} BER with 95% confidence level, at least 3×10^{12} bits need to be captured without a single error. Even at 3 Gb/s data rate, it takes more than 15 min to capture that many bits. The jitter tolerance test is even more troublesome than that because it has to measure the BER by sweeping periodic jitter (PJ) in frequency and amplitude. Although the testing time can be reduced via extrapolation from higher BERs using the BERT scan method, the results may not be accurate if the extrapolation is from inappropriate BER levels.

Loopback-based margining test may not detect some failures in devices. With the challenges of keeping test costs down and testing large numbers of serial I/Os in a single chip, the timing margining test along with loopback configuration has been widely adopted by companies [8, 9]. However, a simple loopback test that determines maximum and minimum limits on phase offset (i.e. timing margin) does not provide adequate test coverage for analog performance variations [10]. Thus, it might not fully replace either conventional jitter or BER testing.

While equalizer design has been studied for a long time and a number of novel architectures have recently been developed to mitigate channel bandwidth limitation, high-quality and cost-effective production test methods for these equalizers are not yet well developed. The pre-emphasis block in the TX can be directly observed from output pins; however, since equalizer outputs are typically embedded in the RX, they may not be directly measured using external equipment unless additional pins are added. In addition, measuring the eye-diagram on-chip to guarantee the operation of equalizers in the multi-gigahertz range requires a significant amount of internal circuitry.

1.3 Contributions of the Dissertation

In this dissertation, we propose novel testing techniques that address the previously mentioned challenges in high-speed interface testing. The contributions of the research can be categorized into four groups:

1. **An efficient jitter measurement technique using fast comparator sampling**
   We have proposed an efficient jitter measurement technique using fast comparator sampling. This technique uses a simple sampler circuit and an efficient analysis algorithm for jitter measurement. Due to its simplicity, small size, and high bandwidth, it is applicable for either on-chip jitter measurement or measurement using Automatic Test Equipment (ATE). The approach combines partial measurements based on individual data edge regions, in contrast to more common approaches that first accumulate data from multiple edge regions. This enables the technique to effectively separate RJ from DDJ and low-frequency PJ.
2. **BER estimation technique**
   We have proposed the BER estimation technique for high-speed serial links, which utilizes the jitter spectral information extracted from the transmitted data and some key characteristics of the CDR circuit in the receiver. In addition to giving insight into both the behavior of the CDR loop and the contribution of the jitter to the BER, the estimation technique can be used to accelerate the jitter tolerance test by eliminating the conventional BER measurement process. We propose two different versions: one for use with linear CDR circuits and the other for non-linear CDR circuits. Experimental results comparing the estimated BER and the measured BER demonstrate the high accuracy of the proposed technique.
3. **Gap analysis in timing margining test and solution for reducing the gap**
   Timing margining test used in conjunction with loopback configuration has become a popular design-for-test (DFT)-based test method for high speed interfaces. The timing margining test can evaluate most of the transceiver's functionality without relying on very expensive high-speed and high-pin-count ATE. However, does it adequately cover every type of jitter specification? To

answer this question, we first explore possible sources that might generate some gaps between the timing margining test and the conventional jitter test. In order to reduce the gap between two methods, we also propose an accurate total jitter estimation technique. In this technique, instead of relying only on the timing margin, a few BER measurements are taken at high BER levels (i.e. higher than 10^{-6} levels). Then, the timing margin at 10^{-12} level is estimated via extrapolation using different fitting algorithms. We demonstrate its value for a more efficient and accurate total jitter estimation at a very low BER level.

4. **Two-tone test method for continuous-time adaptive equalizer**
   A novel test method for continuous-time adaptive equalizers is proposed. This technique applies a simple two-sinusoidal-tone signal as a stimulus and includes a root-mean-square (rms) value detector for testing. The advantages of the proposed technique are as follows: (1) The test stimuli can be easily generated and applied. (2) The test output is a DC signal and thus can be easily measured. (3) The extra on-chip circuitry needed for supporting the technique requires only a very small area overhead and does not result in any performance degradation. To validate the technique, we used a recently published adaptive equalizer as our test case and conducted both behavioral and transistor-level simulations. Simulation results demonstrate that the technique is effective in detecting defects in both the equalization filter and the adaptation loop, either of which might not be easily detected using the conventional eye-diagram method.

Chapter 2
An Efficient Jitter Measurement Technique

Most high-speed communications standards specify the jitter performance to ensure the quality of the communication links. As jitter is the major contributor to BER, communications link performance can be verified relatively quickly by measuring jitter.

Jitter can be measured with a variety of methods [1, 4–7, 15, 16]. Some instruments and methods are more accurate if the jitter is Gaussian, which is often not the case (e.g., if PJ is present), and each testing method has its own pros and cons as discussed in Chapter 1.

In this chapter, we propose a rapid RJ measurement technique that builds on the Probability Digitizer technique [11, 12] and unpublished algorithms developed by Dr. Leonard Monk and Francine Hallé, and incorporated in Teradyne's Serial Port Qualifier (SPQ) and Source Synchronous Pin Electronics (SSPE) instruments. It is a variant of Time-Domain Data Acquisition [4], with two important modifications. First, it uses a fast sampling frequency that can even exceed the data rate to minimize test time and reject low-frequency PJ. Second, the fact that only individual, as opposed to cumulative, edge measurements are made permits the use of short pattern sub-sequences and filters out DDJ and low-frequency PJ. When measuring RJ, we do not want PJ to affect the result, especially PJ of such a low frequency that it will be ignored by receiving devices.

In Section 2.1, we briefly describe the comparator undersampling technique. Section 2.2 details how our techniques measure RJ and their accuracy. Section 2.3 presents simulation and measurement results to validate our technique. Section 2.4 summarizes our conclusions and suggestions for further work.

2.1 Comparator Undersampling Technique

As device data rates increase, comparator undersampling has advantages both for ATE [13] and for on-chip testing [6] due to its simplicity, small size, and high bandwidth.

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_2,

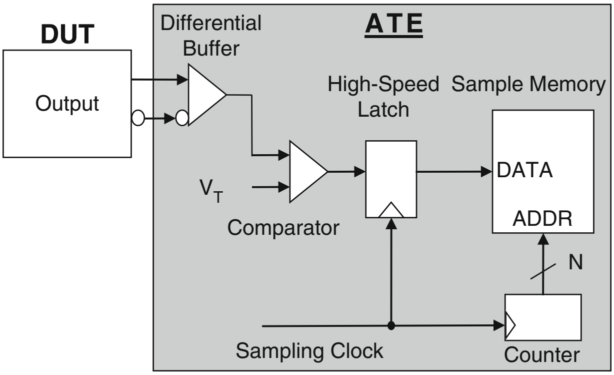


Fig. 2.1 Block diagram of comparator sampler

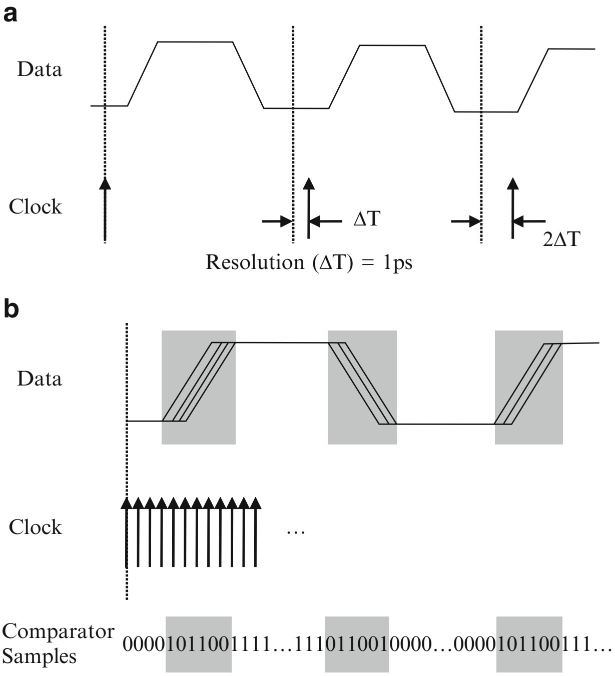


Fig. 2.2 Comparator data acquisition: (**a**) real-time data sampling, (**b**) successive data samples

Figure 2.1 is a block diagram of our comparator sampler. Differential data from a device transmitter is buffered and compared against a threshold, which represents the edge transition voltage.

Figure 2.2 illustrates the comparator data acquisition. In Fig. 2.2a, the clock period is set at 1ps greater than the period of a repeating pattern, so each clock samples the pattern 1ps later than the previous sample. The clock acts as a strobe walking across the pattern with 1ps resolution.

Figure 2.2b illustrates the reconstructed data that appear to have been sampled very densely with each sample separated by 1ps. If the data contains jitter, the result in memory is a generally non-monotonic transition region between all 0s and all 1s (gray region). We define the transition region for a zero-to-one transition as the first-one-to-last-zero, and vice versa.

A "1" ("0") sample indicates that the buffered data transitioned high (low) at a time before the specific clock edge that took the sample. Since jittered edge positions are stochastic, the likelihood of an edge occurring in a particular position can be described using statistical functions. If samples were taken at the same position multiple times, the density of ones would represent the probability that a rising edge occurred prior to that effective position.

The distribution of increasing probabilities could be described as a cumulative distribution function (CDF) of the jitter. The derivative of the CDF yields the probability distribution function (PDF), which in this case describes the probability that an edge occurred in a specific region. Since random jitter can be defined as the probability of an edge occurring over a range of positions, we can analyze the PDF to determine the edge's jitter.

It should be noted that in our method, we only sample each effective position once. This results in a CDF that is only a crude estimation of the edge's actual CDF. In fact, it appears little like a CDF, which would be monotonic. Despite this, if the sample resolution is small enough and the latch bandwidth high enough, this estimate is good enough to produce accurate measurements of RJ, as will be shown, although non-Gaussian jitter sources complicate the matter [14, 15].

The mean and standard deviation are subsequently calculated from each transition region's PDF. The mean edge positions can be used to estimate DDJ. Rms jitter is calculated by an aggregate estimate of each region's standard deviation. The details of how each region's mean and standard deviation is calculated and the validity of this technique are described in the following sections.

2.2 Random Jitter Measurement

Looking at an individual edge region, i, RJ for that edge equals its standard deviation:

$$\mathrm{RJ}_{\mathrm{Edge\ (i)}} = \sigma_{\mathrm{Edge\ (i)}}$$

However, most practical measurements will deduce RJ from multiple edge measurements. The RJ component must be measured accurately to avoid overstating TJ, which is often used to differentiate good devices from bad. Given a system with only deterministic and Gaussian random jitter, TJ is estimated by [1]:

$$\mathrm{TJ} = \mathrm{DJ} + (14.069^{*}\mathrm{RJ})$$

for a 10^{-12} BER. Since RJ is multiplied by 14, inaccurate RJ measurements have a large effect on the TJ estimate. Note that inferring TJ from RJ requires knowledge of the nature of the RJ distribution.

2.2.1 Proposed RJ Measurement Technique

Using the binary data for each edge transition region, our approach is to calculate the standard deviation of each individual region instead of first accumulating data for multiple regions. We then calculate an aggregate estimate. This approach is insensitive to timing disturbances occurring between edges, since the measurement does not accumulate timing errors from one edge to the next, unlike cumulative-edge techniques, such as a histogram measurement performed by a traditional scope. Also, calculating the standard deviation does not require edge distributions to be Gaussian or symmetric.

Our RJ measurement for N edge transition regions is computed according to:

$$\mathrm{RJ} = \sqrt{\frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \ldots + \sigma_\mathrm{N}^2}{\mathrm{N}}}$$

where σ_j is the standard deviation of the jth pattern edge. The optimal way to combine the measurement values depends on the nature of the data itself. In practice, the approach is selected primarily for computational efficiency and robustness to outliers, given an expected distribution.

When a sigma is calculated using many sample points per transition region, the root-mean-square average works well for combining the individual sigmas. However, it is not unusual to have fewer than a dozen points per region, causing the variances to follow a chi-squared distribution with a pronounced skew, rather than one that is nearly Gaussian. The longer tail on the high side of the distribution, along with the root-mean-square average (which more heavily weights large values) overestimates the true value of sigma. Our proposed method should work well given 20 or more samples per transition region. Other methods of combining the individual sigmas may work better in practice, given fewer points.

Using information sampled at a fixed sample resolution Δt, the mean and standard deviation for each edge transition are calculated as follows:

1. Compute discrete derivatives (Δ[i]) between successive samples as a discrete approximation of the PDF.
2. Compute the first and second statistical moments (M1 and M2) of the difference values for each edge region, given by:

$$\mathrm{M1} = \sum_{\mathrm{Edge\_region}} (\Delta[\mathrm{i}]^{*}(\mathrm{i}+0.5)) = \sum_{\mathrm{Edge\_region}} (\mathrm{M1_i})$$

Table 2.1 Moment calculations for a single edge region

		Δ [i] = A[i + 1]		
Sample #(i)	Data A[i]	−A[i]	$M1_i$	$M2_i$
	0			
103	0	0	0	0
104	0	0	0	0
105	0	1	105.5	$(105.5)^2 + (1/12)$
106	1	−1	−106.5	$-(106.5)^2 + (1/12)$
107	0	0	0	0
108	0	1	108.5	$(108.5)^2 + (1/12)$
109	1	−1	−109.5	$-(109.5)^2 + (1/12)$
110	0	1	110.5	$(110.5)^2 + (1/12)$
111	1	0	0	0
112	1	−1	−112.5	$-(112.5)^2 + (1/12)$
113	0	1	113.5	$(113.5)^2 + (1/12)$
114	1	−1	−114.5	$-(114.5)^2 + (1/12)$
115	0	1	115.5	$(115.5)^2 + (1/12)$
116	1	0	0	0
117	1	0	0	0
	1			
Sum			110.5	12237
Mean			110.5	
Variance (σ^2)			$12237 - (110.5)^2 = 26.75$	
Standard deviation (σ)			5.17	

$$M2 = \sum_{\text{Edge\_region}} \left(\Delta[i]^*(i+0.5)^2 + |\Delta[i]|/12\right) = \sum_{\text{Edge\_region}} (M2_i)$$

3. Calculate the mean and standard deviation:

$$\text{Mean}(\mu) = M1, \text{and}$$
$$\text{Standard deviation}(\sigma) = \sqrt{M2 - M1^2}$$

Table 2.1 gives an example of the steps necessary to calculate the mean and standard deviation for a single edge transition. In calculating M1, 0.5 is added to each sample number on the assumption that all information within the histogram bin is located in the center of the bin. This reduces the square sum by 1/12, compared to a uniform distribution in each bin, which must be compensated in computing the second moment. In every jitter region, the same steps are applied to derive the mean and standard deviation individually.

This technique generally yields different results than one superimposing data from multiple edges, because it filters out variations in the mean positions of each edge. This effect is illustrated in Fig. 2.3. This also greatly reduces low-frequency PJ and DDJ components in the data from appearing in the RJ measurement, since the same roll-off affects all the above components similarly. If a device's RJ has a substantial low-frequency component of concern, then the proposed approach would be disadvantageous. However, since Clock Data Recovery (CDR) circuits

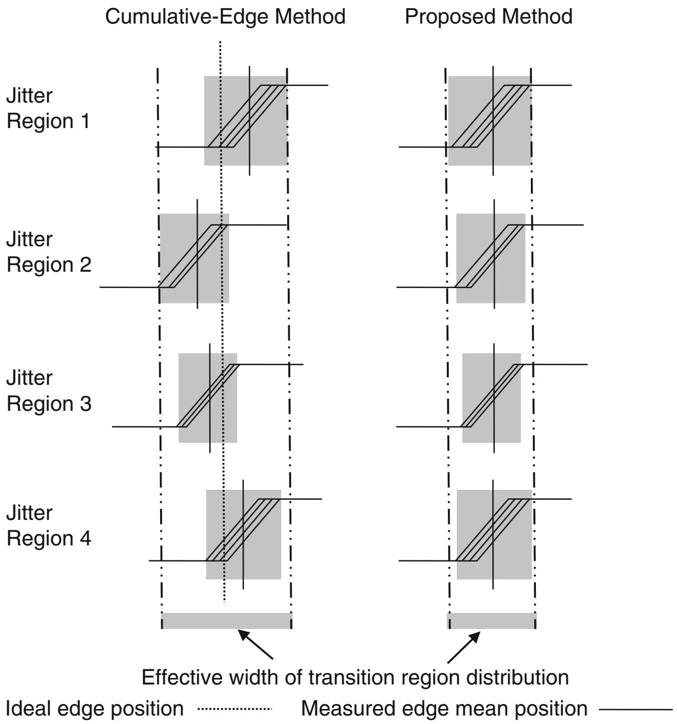


Fig. 2.3 Effects of cumulative-vs individual-edge processing on transition region

track large-amplitude, low-frequency jitter within their bandwidth, the benefit of this technique is that it can filter out all the RJ and PJ below a reasonable cutoff frequency (e.g., 1 MHz or so).

2.2.2 Limitations of the Technique

To make accurate measurements, it is important that the sample resolution is a fraction of the standard deviation of the jitter. On the other hand, if the sample resolution is too small, the acquisition time per apparent edge will be long and any high-frequency PJ will influence the calculated RJ result. In practice, a dozen to several dozen samples in each transition region should be acquired, and if PJ is fast enough so that one or more cycles are captured per transition, it can corrupt the distributions of individual region. These requirements must be balanced in actual use. All of the time-domain sampling approaches surveyed suffer from this limitation, but higher sample rates raise the cutoff frequency. We suggest further work to quantify the effects of high-frequency PJ.

Note that as with all sampling techniques, this technique is subject to aliasing effects. An example of how aliasing can affect jitter measurement is as follows:

Sampling frequency = 10 MHz
PJ = 100 MHz

Every time a strobe occurs, it will hit the same phase of PJ; thus, the PJ will not be observed. The data rate is irrelevant for the above example. However, if one considers the following example, where

Data rate = 1 Gb/s (nominal, ignoring sample resolution)
Bits/pattern = 30
Pattern rate $\approx$ 30 MHz (ignoring sample resolution)

In this case, the PJ would affect the DUT signal, even though the measurement would not capture it, due to the relative synchronicity of the PJ and the sampling frequency.

2.3 Experimental Results

To validate our technique, both simulations and experiments were performed. A MATLAB simulation is first performed, and the results compared to those of the high-speed sampling prototype measurement. Three different combinations of jitter are injected with data:

Case 1: Only RJ is injected
Case 2: RJ + PJ is injected
Case 3: RJ + DDJ is injected

All three cases were simulated; only the first two cases were experimentally verified. Common conditions were as follows:

Data rate (F_{DATA}) : 6.4 Gb/s
Pattern
– Cases 1, 2: 1010 pattern
– Case 3: 20-bit pattern, 00001100110010101111
Number of samples: 32,000
Sampling resolution (Res): 0.5 ps
Number of Bits skipped (N_{SKIP}) between adjacent samples:
– Cases 1, 2: 6
– Case 3: 20
Frequency of adjacent samples:

$$\{F_{ADJ\_SAMPLE} = 1/((N_{SKIP}{}^{*}T_{DATA}) + Res)\}$$

– Cases 1, 2: 1.0661 GHz
– Case 3: 319.9488 MHz

2.3.1 Simulation Results

2.3.1.1 Simulation: Case 1

First, data with only injected RJ was analyzed. Calculations were performed using the proposed method and a conventional histogram method that does not adjust mean edge positions, for comparison purposes. Figure 2.4 illustrates the simulation results. In each case, 20 iterations are simulated and averaged to reduce measurement variability.

In the absence of injected PJ or DDJ, both methods measure the pure RJ accurately, as expected. If we inject PJ as shown in the next section, the differences increase.

2.3.1.2 Simulation: Case 2

A single PJ tone was injected along with RJ to observe measurement leakage. We injected a 2 ps rms value of RJ with different frequencies of PJ with a peak-to-peak amplitude of 12 ps. PJ frequencies were chosen based on a bandwidth roll-off characteristic frequency (F_c, in Hertz), defined as:

$$F_c = (F_{ADJ\_SAMPLE}{}^* \text{ Res})/A_{JA}$$

where A_{JA} is the jitter amplitude in picoseconds. The characteristic frequency F_c is the frequency at which a single jitter period is captured in one edge transition region. F_c gives some indication of periodic jitter frequencies that may leak into the RJ measurement. F_c is a function of the sampling frequency, sample resolution, and the jitter being measured. Note that A_{JA} is calculated as

$$A_{JA} = K^* RJ_{rms} + 2\sqrt{2}^* PJ_{rms},$$

where RJ_{rms} is the rms value of RJ, and PJ_{rms} is the rms value of PJ.

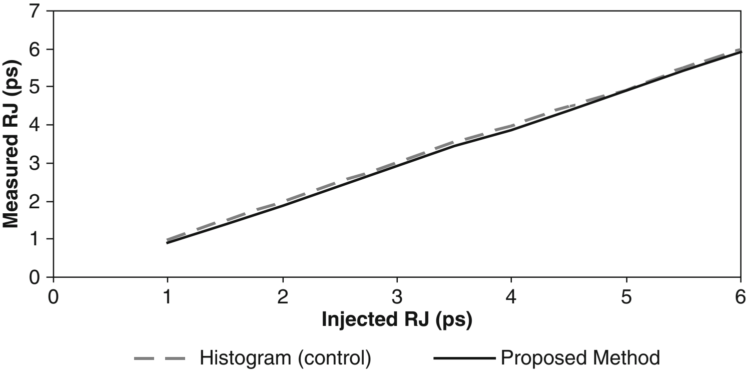


Fig. 2.4 RJ simulation results with only RJ injected

K is a scalar value relating the random jitter rms value to peak-to-peak jitter amplitude. K is a weak function of both the sample resolution granularity relative to the sigma of the distribution and the number of edges captured during the measurement. Our simulations suggest that for 100–1,000 edges captured and 5–50 sample steps per sigma, $2.5 < K < 3.5$. For $K = 2.8, A_{JA} = 17.66$ps. For Case 2, the characteristic frequency is:

$$F_c = (1.0661\,\text{GHz}^{*}0.5\,\text{ps})/(17.66\,\text{ps}) = 30\,\text{MHz}$$

Figure 2.5 shows simulated RJ measurement error as a function of PJ frequency for both methods, with the error representing the difference between the injected RJ and the measured RJ.

RJ measured using the cumulative histogram is nearly constant and overstated for the entire frequency range, and the reported value is similar to the overall rms value of both RJ and PJ, combined:

$$\text{RMS}_{\text{RJ+PJ}} = \sqrt{\text{RJ}^2_{\text{rms}} + \text{PJ}^2_{\text{rms}}} = 4.69\,\text{ps}$$

(Note: Since the errors between the injected and the measured RJ are represented in Fig. 2.5, the values are close to $4.69 - 2 = 2.69$ ps). The proposed technique is accurate to within 0.5 ps range up to $F_c/3$ (10 MHz). Above this frequency, the PJ component appears in the RJ measurement.

2.3.1.3 Simulation: Case 3

A 20-bit pattern was used for this simulation. Figure 2.6 shows the data pattern and how much DDJ is injected on each edge transition. The DDJ applied to each edge was chosen arbitrarily, not based on a specific channel model. As before, a 2 ps rms value of RJ is injected.

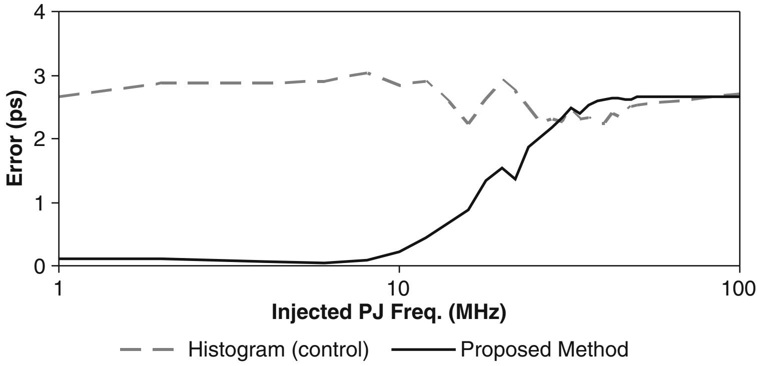


Fig. 2.5 Simulated RJ measurement error as function of PJ frequency

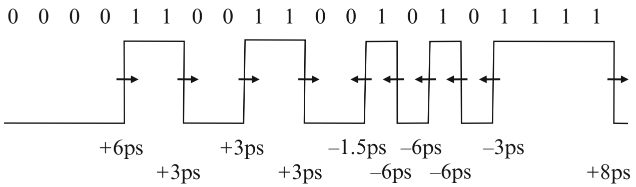


Fig. 2.6 Data with DDJ

Table 2.2 RJ simulation results with DDJ + RJ injection

	Measured RJ (histogram)		Measured RJ (proposed method)	
RJ (RMS)	Rms (ps)	Error (ps)	Rms (ps)	Error (ps)
2 ps	5.54	3.54	1.87	0.13

Table 2.2 shows simulated RJ measurement results for both methods. As can be seen, measured RJ from the proposed method is not affected by DDJ – it returns the same value as Case 1 (i.e., when only RJ is injected). On the other hand, the cumulative histogram overstates the measurement substantially (to be fair, a traditional scope could be set up to make this measurement correctly).

2.3.2 *Measurement Results*

Two measurement results, one from an undersampling oscilloscope and another from a high-speed sampling prototype, are compared in this section. The oscilloscope performs a measurement similar to the cumulative histogram used in the simulation.

Figure 2.7 shows the experimental setup. An Anritsu MP1763B Pulse Pattern Generator (PPG) generates a 6.4 Gb/s differential data pattern. An Anritsu 3692A signal generator generates a single-ended clock for the fast sampler; the scope measurements are instead triggered by the PPG clock. A Micronetics 5624 noise generator generates injected RJ, and an HP 8648C signal generator generates PJ. The RJ, PJ, or both waveform voltages are summed with the data pattern using a pair of power splitters. The timing jitter they introduce is due to the voltage noise summed during the rise/fall of the data edges. Jitter is measured by the Tektronix TDS8200 oscilloscope or by the fast sampling prototype. Test conditions are the same as for the simulations.

2.3.2.1 Experiment: Case 1

First, we injected RJ using the same conditions as for the simulation and measured it using both the oscilloscope and fast sampling prototype. Figure 2.8 illustrates the results.

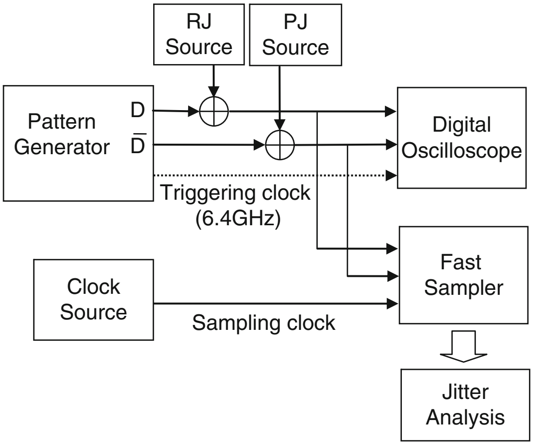


Fig. 2.7 Experimental setup

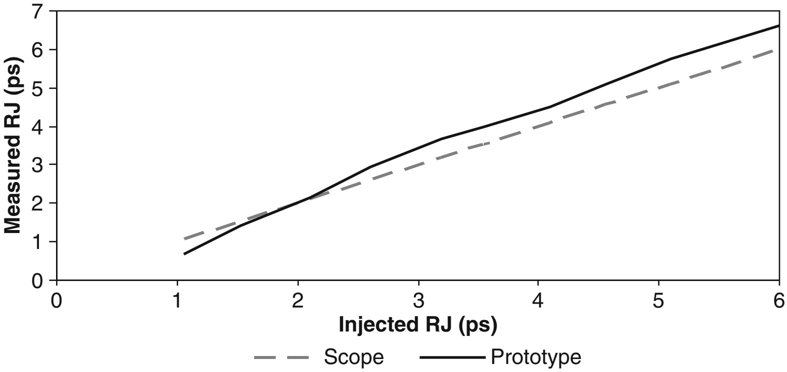


Fig. 2.8 RJ measurement results with only RJ injected

In most cases, the error between the scope and the prototype measurements are less than 0.5 ps, and the prototype generally overstates the result. The scope result is plotted as the ideal, although it is difficult to know precisely how much jitter is delivered to the experimental setup. We suspect that the measurement differences are due to the way the aggregate estimate is computed, but that close correlation between the techniques could be achieved with additional calibration.

2.3.2.2 Experiment: Case 2

Next, we injected PJ + RJ. The rms value of RJ is fixed at 2 ps, and we injected different PJ frequencies, one at a time, with the same peak-to-peak amplitude as the

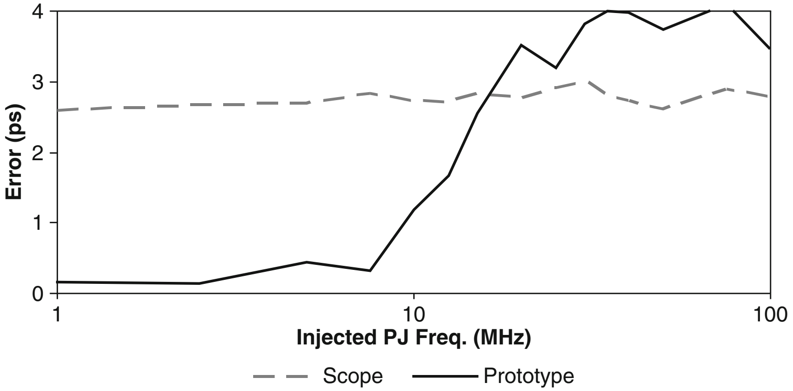


Fig. 2.9 Experimental RJ measurement error as PJ frequency

simulation (i.e. 12 ps). Then, RJ was measured using the fast sampler and the scope. Figure 2.9 illustrates the experimental results. The error represents the difference between the injected RJ and the measured RJ.

As expected, RJ measured by the scope overstates the result due to PJ, regardless of frequency. However, the proposed method measures RJ much more accurately, within 0.5 ps error, up to $F_c/4$. The proposed method overstates the result above about $F_C/2$, as PJ elongates the individual edge transition regions.

2.4 Summary

A random jitter measurement technique using a simple sampler circuit and efficient analysis algorithm has been demonstrated. The technique has advantages for ATE applications testing multiple high-speed serial interfaces. Simulation and measurement results are compared between the proposed technique and the cumulative histogram based technique, and the proposed technique demonstrably measure RJ more accurately in the presence of DDJ and low-frequency PJ, up to 25% of a deduced characteristic frequency.

High-frequency PJ is shown to contaminate the RJ measurement results. Further work should be to develop a method of separating PJ from the measured RJ, and to further quantify the effect of sample resolution and total number of samples on measurement accuracy.

Chapter 3
BER Estimation for Linear Clock and Data Recovery Circuit

BER is a widely used figure of merit to indicate the quality of a communication system. A BER at 10^{-12} or even lower is often required for most communication systems. To measure such a low BER (at the level of 10^{-12}) with reasonable accuracy, it would be necessary to capture more than 10^{12} samples. As a result, the test time would be excessively long even if the system runs in the gigahertz range. Thus, the conventional testing method, which compares each captured bit for error detection, requires a long test time and is therefore not cost-effective for high-volume production testing [17].

One of the major contributors to the BER is jitter. Jitter is specified in terms of the RJ and the DJ in most high-speed communication standards [4, 18, 19]. These values are based on the time-domain characteristics (e.g. the peak-to-peak or the rms values), and the spectral contents of the jitter (e.g. the amplitude and the frequency of the PJ) cannot be extracted from them. The CDR circuit, which recovers the clock signal from the data, can tolerate a certain frequency range of jitter. Thus, both the jitter spectral information and the characteristics of the CDR circuit have to be considered for measuring the overall system performance (i.e. BER), if PJ is present.

Several recently proposed techniques have studied jitter in the spectrum domain [20,21]. The CDR circuit's low frequency jitter tracking capability is also mentioned in some recent publications [20, 22, 23]. However, most of the literature limits the discussion to the magnitude response of the CDR circuit's jitter transfer function. Although a couple of methods take into account the phase transfer characteristic for jitter analysis [24, 25], none of them thoroughly analyze its effects on BER performance.

In this chapter, we propose a method for estimating the BER using the following two sets of parameters: (1) the jitter spectral information, extracted from the signal at the input of the receiver. The information includes the rms value of the RJ and the DJ characteristics, such as frequencies and amplitudes of the PJ components. (2) The CDR circuit's jitter transfer characteristics including both the magnitude and the phase responses. We assumed that the jitter spectral information, which can be measured using the techniques proposed in [20, 21], is available. After illustrating the basic analysis based on these parameters, we further extend the analysis to incorporate the intrinsic noise from the CDR circuit itself.

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_3,

In principle, if the frequency of input jitter is relatively low, the CDR circuit can track the jitter perfectly, and thus no bit errors will occur. However, if the input jitter varies rapidly, the CDR circuit may not track the jitter, and some errors will occur. On the other hand, the CDR circuit has an opposite reaction to the CDR circuit's internal noise. That is, the high frequency component of the internal noise is transparent (and thus will be directly added) to the recovered clock instead of being filtered out by the CDR circuit. In addition, the phase response of the CDR circuit, which determines the timing response in clock recovery, has a very strong correlation to the BER. If the jitter frequency falls into the range in which the phase delay is non-zero, the CDR circuit introduces some timing delay to the recovered clock which will, in turn, contribute to the BER. Within a specific frequency range, this timing delay can cause a significant increase in the BER. Detailed descriptions of how the magnitude and the phase responses of the CDR circuit affect the BER will be described in the following sections.

In the next section, we first summarize the relationship between the BER and the rms value of the RJ. Section 3.2 gives details of the BER variations, which depend on the jitter spectrum and a CDR circuit's jitter transfer characteristics. Section 3.3 extends the BER estimation by including the noise of the CDR circuit itself. Section 3.4 presents the experimental setup and results for the validation of the proposed technique. Section 3.5 summarizes the chapter and discusses future work for further improving the technique.

3.1 BER Analysis with Random Jitter

3.1.1 Error Occurrences

In a serial communication system, the clock is embedded in the transmitted data. At the receiver side, the clock is recovered through the CDR circuit. The recovered clock is then used for latching the data. Thus, the alignment between the data and the recovered clock is crucial for latching the correct data. If there is no timing jitter in the transmitted data, the sampling point will be in the middle of the data eye. However, in the presence of jitter, the sampling point will not be right in the middle of the data eye. If a large amount of jitter causes a misalignment between the data and the recovered clock exceeding a certain level, a bit error occurs. Figure 3.1 illustrates error occurrences when jitter is present in the transmitted data. This illustration assumes that the jitter has only a random component, which is assumed to have a Gaussian distribution.

3.1.2 BER Estimation with Random Jitter

If only RJ is present in the transmitted data, the BER can be easily estimated [1]. Since the CDR circuit can track only the low frequency components of the jitter, the

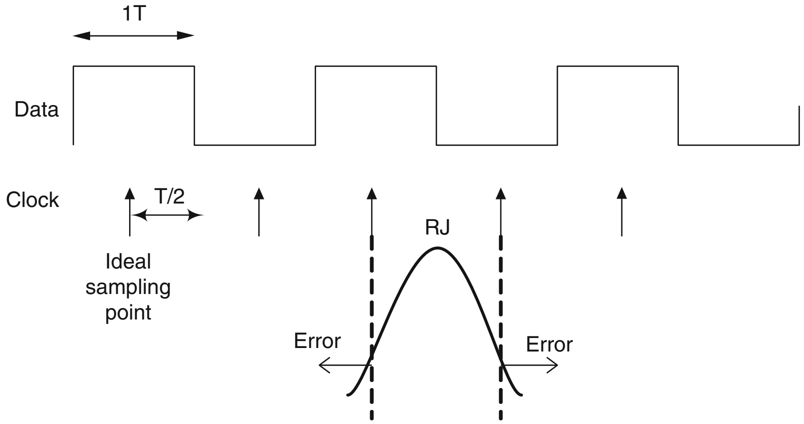


Fig. 3.1 Error occurrence

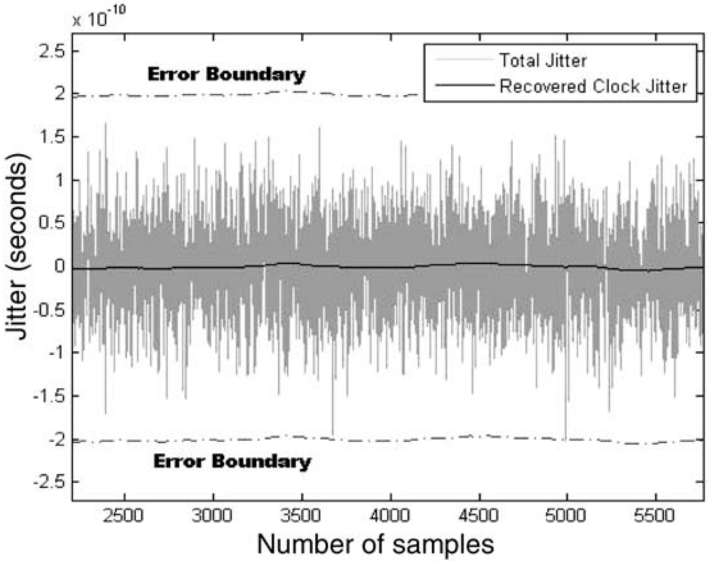


Fig. 3.2 The input jitter and the recovered clock jitter

rapidly varying RJ cannot be tracked. Figure 3.2 shows the jitter in the transmitted data (2.5 Gbps) with only the RJ component and the recovered clock jitter produced by the CDR circuit (from simulation). As observed, the recovered clock does not track input jitter at all (i.e. the jitter of the recovered clock is close to zero whereas RJ in the transmitted data is significant). Thus, errors occur when the input jitter exceeds plus or minus half of a Unit Interval T/2 (where T is denoted as one Unit Interval (UI)).

RJ is commonly characterized by a zero-mean Gaussian distribution function. Therefore, the probability that the RJ exceeds a certain threshold can be calculated using the Q-function, which is defined as:

$$Q(x) = \left[\frac{1}{(1-a)x + a\sqrt{x^2+b}}\right] \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{3.1}$$

where $a = 1/\pi$ and $b = 2\pi$ [26]. This Q-function is used to calculate the probability that the random component, which has zero mean and unity standard deviation, is larger than any given value x. For the case of our interest, errors occur when the magnitude of the RJ is larger than T/2 and the rms value of RJ (σ_{RJ}) is not unity. Therefore, the threshold value x would be

$$x = \frac{T/2}{\sigma_{RJ}}$$

Then, the BER can be estimated as:

$$BER = 2 \cdot \rho_T \cdot Q\left(\frac{T/2}{\sigma_{RJ}}\right). \tag{3.2}$$

The Q-function is multiplied by 2 because the error occurs on both sides of the data transition (i.e. when jitter is greater than the threshold or less than $(-1)^*$ threshold). In addition, the BER is linearly proportional to the transition density (ρ_T) of the data [27]. For a clock-like pattern (i.e. a periodic "1010" stream), the transition density is equal to 1.

3.2 BER Analysis with Random Jitter and Periodic Jitter

The estimation becomes much more complex when both PJ and RJ components are present. Table 3.1 shows the simulation results of the BER with different combinations of RJ and PJ (the details of the simulation setup will be explained in Section 3.4). In this experiment, we injected the PJ into the transmitted data at four different frequencies −50 KHz, 1 MHz, 10 MHz, and 100 MHz – with the same amplitude (0.1 T). In addition, we injected RJ with an rms value of T/7.8. We then

Table 3.1 BER simulation results

PJ		RJ	BER		
Amp	Frequency	Rms	PJ only	RJ only	PJ + RJ
0.1T	50 KHz	T/7.8	0	9.6e−5	9.6e −5
	1 MHz		0		1.15e − 4
	10 MHz		0		1.82e − 4
	100 MHz		0		1.69e − 4

compared the resulting BER for the following three cases: (1) only PJ is injected; (2) only RJ is injected; and (3) both PJ and RJ are injected. As shown in Table 3.1, if the frequency of the injected PJ is low (say, at 50 KHz), the BER for Case (3) (i.e. containing both PJ and RJ) is the same as the BER for Case (2) (i.e. containing only RJ). However, when the frequency of the PJ increases, the BER of these two cases becomes different although the presence of the PJ alone (i.e. Case (1)) does not cause any error. In the following sections, we will give a thorough analysis to explain how the frequency of the PJ affects the BER.

3.2.1 Jitter Transfer Characteristics of a CDR Circuit

As discussed earlier, the CDR circuit can tolerate low frequency components of the input jitter. This characteristic results in an increase of BER when the frequency of the PJ increases. However, this alone cannot completely explain the phenomena observed in Table 3.1. To understand these effects more precisely, we have to take into account the phase response of the CDR circuit in addition to its magnitude response.

A CDR circuit is commonly implemented using the architecture of a Phase-Locked Loop (PLL). The basic block diagram of the CDR circuit under such architecture is shown in Fig. 3.3.

The open- and closed-loop transfer functions of the CDR circuit can be expressed by Eqs. 3.3 and 3.4, respectively [28],

$$\mathrm{H_O(s)} = \frac{2\xi\omega_n + {\omega_n}^2}{m\frac{2\xi}{\omega_n}s^3 + (m+1)s^2} \tag{3.3}$$

$$\mathrm{H_C(s)} = \frac{\theta_o(s)}{\theta_i(s)} = \frac{2\xi\omega_n + {\omega_n}^2}{m\frac{2\xi}{\omega_n}s^3 + (m+1)s^2 + 2\xi\omega_n s + {\omega_n}^2} \tag{3.4}$$

where m is the capacitance ratio, ξ is the damping ratio, and ω_n is the natural frequency. These parameters can be expressed by the following equations:

$$m = \frac{C_2}{C_1}$$

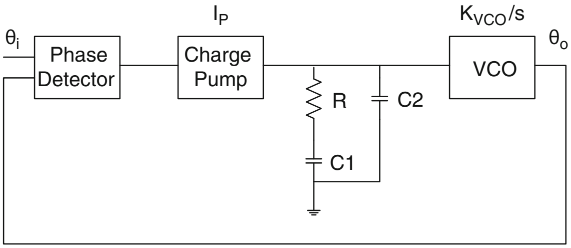


Fig. 3.3 Block diagram of a CDR circuit

$$\omega_n{}^2 = \frac{I_P K_{VCO}}{2\pi C_1}$$

$$2\xi\omega_n = \frac{I_P R K_{VCO}}{2\pi}$$

Note that Eqs. 3.3 and 3.4 are valid only if the data follow a clock-like pattern. If random data are applied, the transfer function will be different because the phase detector (PD) updates the phase error only when a transition occurs in the data. Thus, if the data have a lower transition density, the PD gain will be reduced proportionally. Since the transition density of a pseudo-random bit stream (PRBS) pattern, commonly used as a general purpose test pattern, is half of a clock-like pattern's transition density [29], the open-loop and closed-loop transfer functions for a PRBS pattern could be expressed by Eqs. 3.5 and 3.6.

$$H_{O\_PRBS}(s) = \frac{1}{2} \cdot H_O(s) \tag{3.5}$$

$$H_{C\_PRBS}(s) = \frac{2\xi\omega_n + \omega_n{}^2}{m\frac{4\xi}{\omega_n}s^3 + 2(m+1)s^2 + 2\xi\omega_n s + \omega_n{}^2} \tag{3.6}$$

The solid lines in Fig. 3.4 show the magnitude and the phase responses of the CDR circuit derived from Eqs. 3.4 and 3.6. The parameters are set to $m = 0.005$, $\omega_n = 2\pi \cdot 2 \cdot 10^5$ rad/s, and $\xi = 5$, which result in a 2 MHz loop bandwidth and less than 0.1 dB jitter peaking for the clock-like pattern. Note that jitter peaking could cause degradation to the BER, if its value is large. For our analysis, the jitter peaking was chosen to be at a level low enough so that it will not cause any non-trivial BER degradation. To validate these transfer functions and to observe their dependency on the transition density, a behavioral model of the CDR circuit was designed for MATLAB simulation. The differences of the magnitude and the phase between the injected PJ and the recovered clock jitter were computed for both clock-like and PRBS patterns. The circles and the stars on the lines in Fig. 3.4 represent the simulation results for the clock-like and the PRBS patterns, respectively. As shown, the simulation results matched the responses derived from the equations very well for both patterns.

To analyze the BER, which depends on the jitter frequency, we divide the frequency into four regions based on the magnitude and phase responses as shown in Fig. 3.4:

1. Region 1 (clock-like: 0–70 KHz, PRBS: 0–50 KHz) where the magnitude gain is 1, and the phase curve is flat (i.e. 0°).
2. Region 2 (clock-like: 70 KHz–2 MHz, PRBS: 50–700 KHz) where the magnitude gain is 1, and the phase curve has a non-zero slope.
3. Region 3 (clock-like: 2–40 MHz, PRBS: 700 KHz–30 MHz) where the magnitude gain is less than 1, and the phase curve has a non-zero slope.
4. Region 4 (clock-like: 40 MHz and greater, PRBS: 30 MHz and greater) where the magnitude gain is negligible.

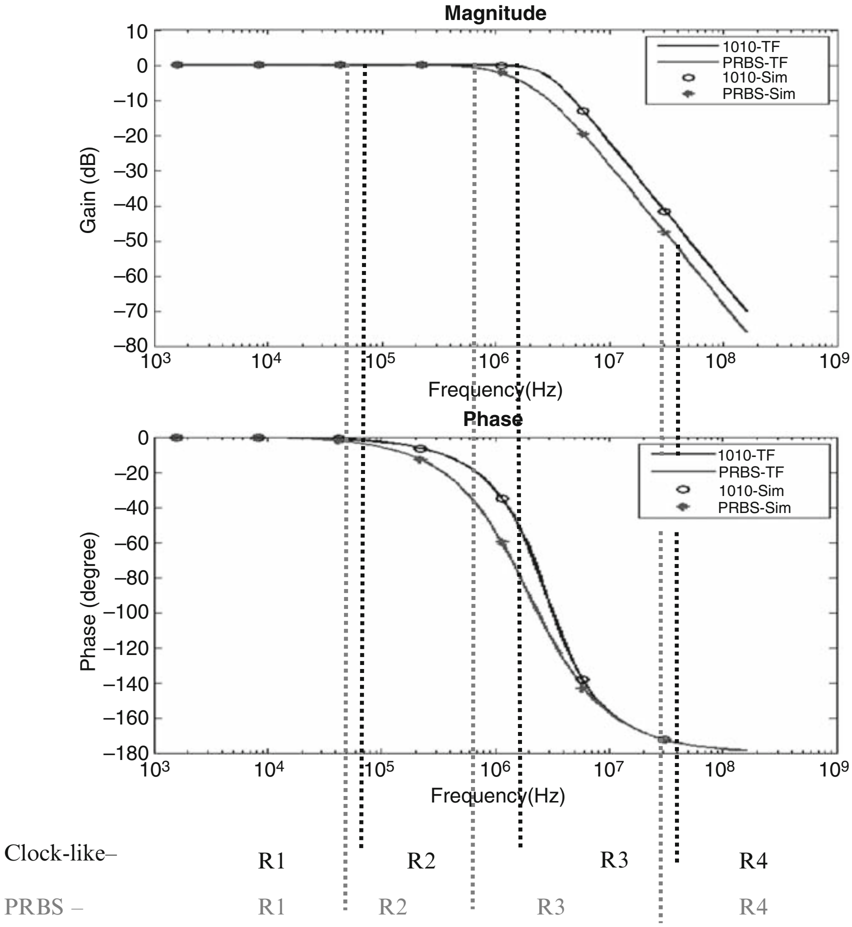


Fig. 3.4 The jitter transfer characteristics of a CDR circuit

3.2.2 BER Estimation with RJ and PJ

In the following section, we first describe the modification to the conventional dual-Dirac model for the BER estimation. Then, we describe the method for estimating the BER in each region.

3.2.2.1 Dual-Dirac Model and Its Modification

One popular approach to the BER estimation is the use of the dual-Dirac model [14]. To calculate the total jitter distribution, it approximates the DJ as two Dirac delta functions and convolves them with the RJ. Figure 3.5 represents this convolution of the DJ and the RJ.

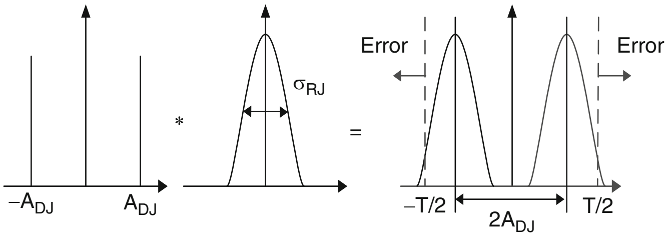


Fig. 3.5 Example of convolution and error occurrence

As discussed in Section 3.1, an error will occur when the total jitter exceeds plus or minus T/2 without considering the clock recovery function (as shown in Fig. 3.5). Thus, in this case, the threshold value x for the Q-function would be

$$\mathrm{x} = \frac{\mathrm{T}/2 - \mathrm{A}_{\mathrm{DJ}}}{\sigma_{\mathrm{RJ}}}.$$

Note that, compared to Eq. 3.2, the mean position of the Gaussian distribution is shifted by the amount of A_{DJ}. Therefore, the BER can be estimated as:

$$\mathrm{BER} = 2 \cdot \rho_{\mathrm{T}} \cdot \mathrm{Q}\left(\frac{\mathrm{T}/2 - \mathrm{A}_{\mathrm{DJ}}}{\sigma_{\mathrm{RJ}}}\right). \tag{3.7}$$

In our case, both PJ and RJ are considered for the BER estimation. However, because the PJ distribution cannot be properly modeled as two Dirac delta functions, we make some modifications to Eq. 3.7. We use two new variables – the effective mean shifting ($\mathrm{A}_{\mathrm{eff}}$) and the effective rms value (σ_{eff}) – to replace A_{DJ} and σ_{RJ}. The details of the derivation of these two new variables, which are based on the ratio of the amount of RJ to the amount of PJ, are given in Appendix A. With the variable replacement, the BER becomes:

$$\mathrm{BER} = 2 \cdot \rho_{\mathrm{T}} \cdot \mathrm{Q}\left(\frac{\mathrm{T}/2 - \mathrm{A}_{\mathrm{eff}}}{\sigma_{\mathrm{eff}}}\right). \tag{3.8}$$

3.2.2.2 BER Estimation Taking into Account Clock Recovery Function

Now we take into account the receiver's clock recovery capability for BER estimation. Each region in the jitter transfer function of the CDR circuit, specified in Section 3.2.1, has distinct magnitude and phase responses to the input jitter. Therefore, we derive different equations for computing $\mathrm{A}_{\mathrm{eff}}$ and σ_{eff} for different regions and use the resulting values to estimate the BER. We first consider the clock-like pattern, and then extend the analysis for the PRBS pattern.

For Clock-Like (1010) Pattern

When the input data is a clock-like pattern, the transition density is one. Thus, the BER will be:

$$\mathrm{BER_{CLK}} = 2 \cdot \mathrm{Q}\left(\frac{\mathrm{T}/2 - \mathrm{A_{eff}}}{\sigma_{\mathrm{eff}}}\right). \tag{3.9}$$

1. **Region 1**

   If the RJ and the PJ in Region 1 are present in the data, the PJ will be perfectly tracked whereas the rapidly varying RJ cannot be tracked by the CDR circuit. Therefore, only the RJ contributes to the BER.

   Suppose that the total jitter is composed of a PJ component $\mathrm{s(t)} = \mathrm{a_1}\sin(\omega \mathrm{t})$ and an RJ component n(t) with an rms value of σ_{RJ}. Errors occur when the input jitter exceeds the error boundaries, which are the recovered clock jitter plus or minus T/2. Because the CDR circuit can completely track the PJ, the recovered clock jitter will be the same as s(t). That is, errors occur when

   $$\begin{aligned}&\mathrm{a_1}\sin(\omega\mathrm{t}) + \mathrm{n(t)} \geq \mathrm{a_1}\sin(\omega\mathrm{t}) + \mathrm{T}/2, \text{or}\\ &\mathrm{a_1}\sin(\omega\mathrm{t}) + \mathrm{n(t)} \leq \mathrm{a_1}\sin(\omega\mathrm{t}) - \mathrm{T}/2.\end{aligned}$$

   The left hand sides represent the total input jitter, and the right hand sides represent the error boundaries. After simplification, we arrive at the following inequality:

   $$|n(t)| \geq T/2.$$

   Because only the RJ affects the BER, the values of $\mathrm{A_{eff}}$ and σ_{eff} for Equation (3.9) will be:

   $$\begin{aligned}\mathrm{A_{eff}} &= 0\\ \sigma_{\mathrm{eff}} &= \sigma_{\mathrm{RJ}}.\end{aligned} \tag{3.10}$$

2. **Region 2**

   Since the phase response of the CDR circuit has a non-zero slope, the PJ in this region is tracked by the CDR circuit with certain delay introduced into the recovered clock. This time delay also shifts the error boundaries, thus increasing the BER. Figure 3.6 shows the input jitter, the delayed recovered clock jitter, and the boundaries at which the errors occur.

   The time delay introduced into the recovered clock jitter depends on the slope of the phase curve. According to [30], the time delay at each frequency ω equals the negative of the slope of the phase at ω. That is:

   $$\tau(\omega) = -\frac{\mathrm{d}}{\mathrm{d}\omega}\{\angle \mathrm{H_C}(\mathrm{j}\omega)\} \tag{3.11}$$

   where $\angle \mathrm{H_C}(\mathrm{j}\omega)$ is the phase response of the CDR circuit.

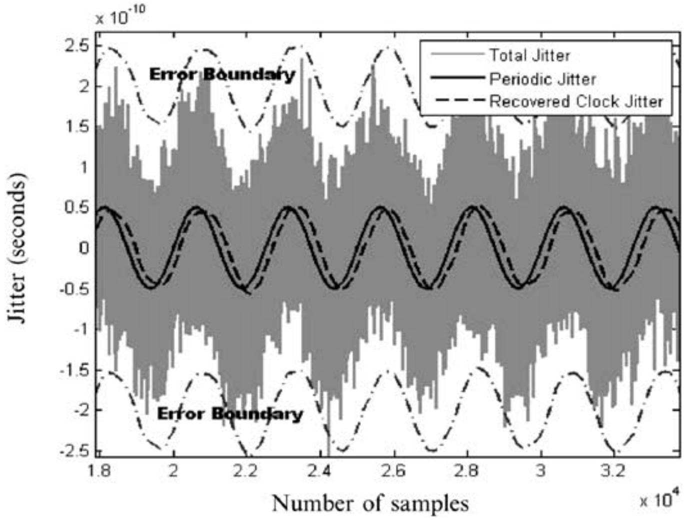


Fig. 3.6 The input jitter and the recovered clock jitter for Region 2

If the CDR circuit tracks the input PJ (i.e. $a_1 \sin(\omega t)$) with some delay t_0(t_0 can be obtained from Eq. (3.11) at a known frequency), the recovered clock jitter can be represented as $a_1 \sin(\omega(t - t_0))$. Therefore, errors occur when

$$\begin{aligned} a_1 \sin(\omega t) + n(t) &\geq a_1 \sin(\omega(t - t_0)) + T/2, \text{or} \\ a_1 \sin(\omega t) + n(t) &\leq a_1 \sin(\omega(t - t_0)) - T/2. \end{aligned} \tag{3.12}$$

Combining and simplifying these two inequalities yields the following inequality:

$$\left| a_1 \sqrt{2 - 2\cos(\omega t_0)} \sin(\omega t + \Theta_1) + n(t) \right| \geq T/2 \tag{3.13}$$

where $\tan \Theta_1 = \sin(\omega t_0)/(1 - \cos(\omega t_0))$.

We can treat the sinusoidal term in the left hand side of Inequality 3.13 as the effective PJ (PJ_{eff}). We can then derive A_{eff} and σ_{eff} by considering two different cases based on the analysis given in Appendix A.

i. When $\sigma_{RJ}/\sigma_{PJ_{eff}} > 1$

For this case, the RJ is more dominant and thus we can approximate PJ_{eff} as RJ (the rationale behind this is explained in Appendix A). Thus, the A_{eff} and σ_{eff} for the left hand side of Inequality 3.13 can be derived as:

$$\begin{aligned} A_{eff} &= 0 \\ \sigma_{eff} &= \sqrt{a_1{}^2(1 - \cos(\omega t_0)) + \sigma_{RJ}{}^2}. \end{aligned} \tag{3.14}$$

Note that σ_{eff} is based on Eq. A.1 in Appendix A, and the variance of a sinusoidal term with amplitude A is $A^2/2$. Therefore, the BER will be calculated simply by substituting Eq. 3.14 for Eq. 3.9.

ii. When $\sigma_{RJ}/\sigma_{PJ_{\text{eff}}} < 1$

A_{eff} and σ_{eff} for this case are:

$$A_{\text{eff}} = \left(1 - 0.9 \cdot \frac{\sigma_{RJ}}{a_1\sqrt{1-\cos(\omega t_0)}}\right) \cdot a_1\sqrt{2 - 2\cos(\omega t_0)}$$

$$\sigma_{\text{eff}} = \sqrt{(1 + 0.84^2) \cdot \sigma_{RJ}{}^2}. \tag{3.15}$$

They are derived based on Eqs. A.2 and A.3 in Appendix A. These values can then be applied to the BER estimation.

3. **Region 3**
In this region, the phase curve has a non-zero slope, and the magnitude gain is less than one. Therefore, the recovered clock jitter has a certain delay and smaller amplitude. In addition, since the time delay is significant (it could be as much as half of the period of the PJ when the phase response of this region is near 180°), the input PJ and the recovered clock jitter can be out of phase. That is, when the input PJ has a maximum (minimum) value, the recovered clock jitter can come close to a minimum (maximum) value. Since the error boundaries are determined by the recovered clock jitter, this out-of-phase phenomenon brings the error boundary curves closer to the input jitter, as indicated in Fig. 3.7. This can make the BER of Region 3 worse than those of other regions (including

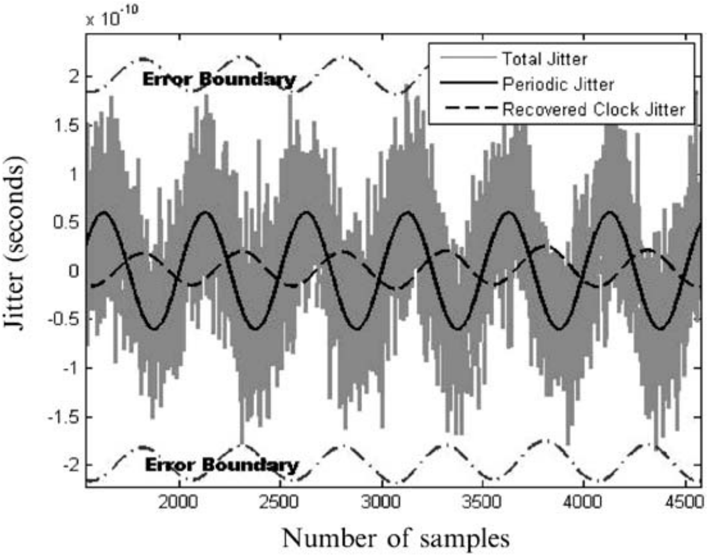


Fig. 3.7 The input jitter and the recovered clock jitter for Region 3

Region 4, which will be discussed later), when the same amount of PJ and RJ are present. This explains an interesting phenomenon: the CDR circuit's tracking ability may sometimes cause a worse BER than the case in which the CDR circuit cannot track the jitter at all (i.e. Region 3 has a higher BER than that of Region 4, as shown in Table 3.1).

If we denote the amplitude of the recovered clock jitter as $\mathrm{a_2}$, which is $\mathrm{a_2 = Gain \times a_1}$ (Gain is obtained from Eq. 3.4), then the error boundaries can be expressed as:

$$\begin{aligned} &\mathrm{a_1 \sin(\omega t) + n(t) \geq a_2 \sin(\omega(t - t_0)) + T/2, \text{ or}} \\ &\mathrm{a_1 \sin(\omega t) + n(t) \leq a_2 \sin(\omega(t - t_0)) - T/2.} \end{aligned}$$

Combining these two inequalities and further simplifying them result in the following inequality:

$$\left| \sqrt{\mathrm{a_1}^2 + \mathrm{a_2}^2 - 2\mathrm{a_1 a_2} \cos(\omega \mathrm{t_0})} \sin(\omega \mathrm{t} + \Theta_2) + \mathrm{n(t)} \right| \geq \mathrm{T}/2$$

where $\tan \Theta_2 = \mathrm{a_2} \sin(\omega \mathrm{t_0})/(\mathrm{a_1} - \mathrm{a_2} \cos(\omega \mathrm{t_0}))$

The sinusoidal term in this inequality can then be treated as the effective PJ for further analysis.

i. When $\sigma_{\mathrm{RJ}}/\sigma_{\mathrm{PJ_{eff}}} > 1$

$\mathrm{A_{eff}}$ and σ_{eff} will be:

$$\begin{aligned} \mathrm{A_{eff}} &= 0 \\ \sigma_{\mathrm{eff}} &= \sqrt{\frac{\mathrm{a_1}^2}{2}\left(1 + \frac{\mathrm{a_2}^2}{\mathrm{a_1}^2} - 2\frac{\mathrm{a_2}}{\mathrm{a_1}} \cos(\omega \mathrm{t_0})\right) + \sigma_{\mathrm{RJ}}^2}. \end{aligned} \tag{3.16}$$

ii. When $\sigma_{\mathrm{RJ}}/\sigma_{\mathrm{PJ_{eff}}} < 1$

$\mathrm{A_{eff}}$ and σ_{eff}, to be applied to Eq. 3.9, will be:

$$\begin{aligned} A_{eff} &= \left(1 - 0.9 \cdot \frac{\sigma_{RJ}}{\sqrt{a_1^2/2 + a_2^2/2 - a_1 a_2 \cos(\omega t_0)}}\right) \\ &\quad \times \sqrt{a_1^2 + a_2^2 - 2 a_1 a_2 \cos(\omega t_0)} \\ \sigma_{\mathrm{eff}} &= \sqrt{(1 + 0.84^2) \cdot \sigma_{\mathrm{RJ}}^2}. \end{aligned} \tag{3.17}$$

4. **Region 4**
Since the magnitude gain is negligible in this region, the PJ component is not tracked at all, as shown in Fig. 3.8. As neither PJ nor RJ components are tracked by the CDR circuit, the error boundaries can be expressed as:

$$|\mathrm{a_1} \sin(\omega \mathrm{t}) + \mathrm{n(t)}| \geq \mathrm{T}/2$$

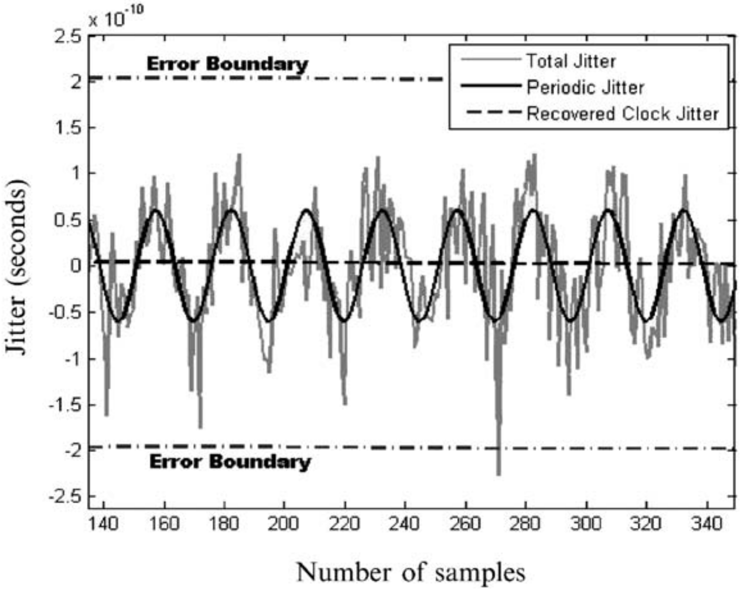


Fig. 3.8 The input jitter and the recovered clock jitter for Region 4

For this region, the effective PJ is the same as the injected PJ.

i. When $\sigma_{RJ}/\sigma_{PJ_{eff}} > 1$

$$A_{eff} = 0$$
$$\sigma_{eff}{}^2 = \frac{a_1{}^2}{2} + \sigma_{RJ}{}^2. \tag{3.18}$$

They can then be used for Eq. 3.9 to estimate the BER.

ii. When $\sigma_{RJ}/\sigma_{PJ_{eff}} < 1$

A_{eff} and σ_{eff} for Eq. 3.9 will be:

$$A_{eff} = \left(1 - 0.9 \cdot \frac{\sigma_{RJ}}{a_1/\sqrt{2}}\right) \cdot a_1$$
$$\sigma_{eff} = \sqrt{(1 + 0.84^2) \cdot \sigma_{RJ}{}^2}. \tag{3.19}$$

For all cases, we have derived A_{eff} and σ_{eff} for Eq. 3.9 to estimate the BER. As the frequency of the PJ changes, the values of A_{eff} and σ_{eff} will be changed, so is the BER. The above analysis shows how the frequency component of jitter affects the BER corresponding to the CDR circuit's jitter transfer characteristics.

For PRBS Pattern

For the PRBS pattern, the equations derived for A_{eff} and σ_{eff} in each region do not change. The only differences are: (1) The time delay (t_0) and the magnitude of the recovered clock jitter in Region 3 (a_2) will be calculated using Eq. 3.6 instead of Eq. 3.4. (2) The transition density (ρ_T) is 1/2. Thus the BER equation becomes:

$$BER_{PRBS} = Q\left(\frac{T/2 - A_{eff}}{\sigma_{eff}}\right). \tag{3.20}$$

3.3 BER Analysis Including Intrinsic Noise in the CDR Circuit

The jitter transfer analysis can be extended to consider the intrinsic noise of the CDR circuit. The main noise source of the CDR circuit is from the Voltage Controlled Oscillator (VCO), whose frequency- and time-domain properties can be specified as phase noise and jitter, respectively [31]. The CDR circuit behaves as a high-pass filter for the VCO noise in contrast to the low-pass filtering behavior for the input noise [23, 32, 33]. The transfer function between VCO phase noise (θ_n) and the output phase (θ_o) is

$$H_n(s) = \frac{\theta_o(s)}{\theta_n(s)} = 1 - H(s) = \frac{m\frac{2\xi}{\omega_n}s^3 + (m+1)s^2}{m\frac{2\xi}{\omega_n}s^3 + (m+1)s^2 + 2\xi\omega_n s + \omega_n{}^2} \tag{3.21}$$

where m, ξ, and ω_n are the same as Eq. 3.4. Figure 3.9 shows a CDR circuit's jitter transfer characteristic for the VCO noise.

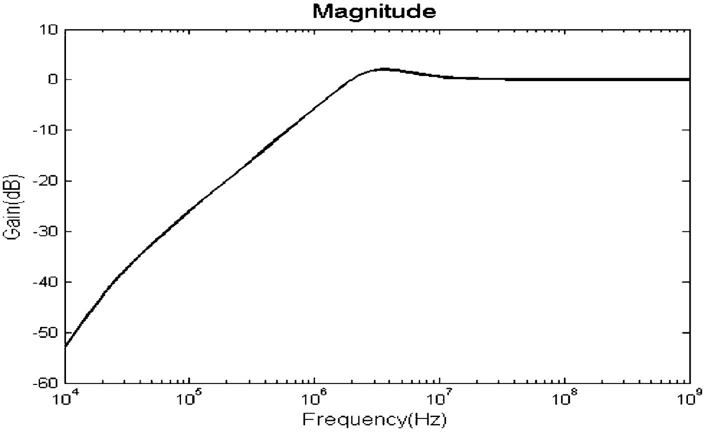


Fig. 3.9 The VCO jitter transfer characteristic

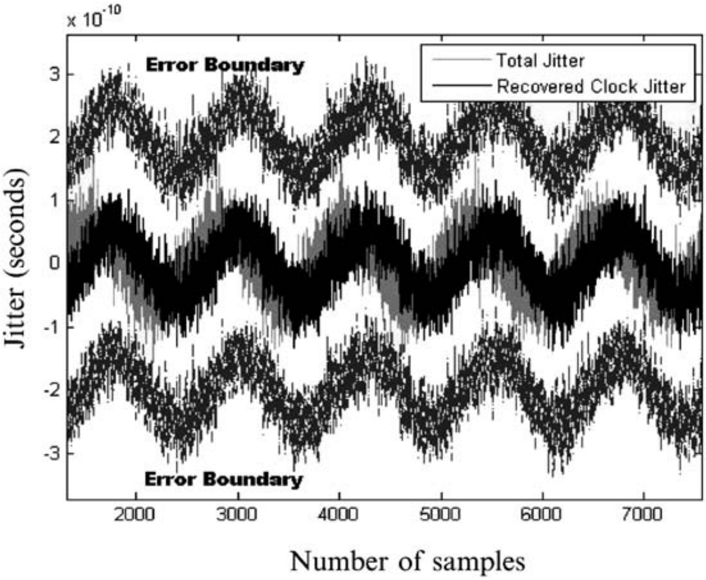


Fig. 3.10 The input jitter and the recovered clock jitter with VCO noise

The VCO noise is often assumed to be a white Gaussian [34]. As white noise has a significant high-frequency component, most of the VCO noise appears in the recovered clock jitter, as shown in Fig. 3.10.

Suppose there is VCO noise $n_2(t)$ with an rms value of σ_{VCO} in the presence of the input PJ $a_1\ \sin(\omega t)$ and RJ $n_1(t)$. As an example, if the input PJ is in Region 2, by modifying Inequalities 3.12, the error boundaries can be expressed as:

$$\begin{aligned} &a_1 \sin(\omega t) + n_1(t) \geq a_1 \sin(\omega(t - t_0)) + n_2(t) + T/2, \text{ or} \\ &a_1 \sin(\omega t) + n_1(t) \leq a_1 \sin(\omega(t - t_0)) + n_2(t) - T/2. \end{aligned} \tag{3.22}$$

By combining and simplifying the inequalities of Eq. 3.22, we arrive at the following inequality:

$$\left| a_1 \sqrt{2 - 2\cos(\omega t_0)} \sin(\omega t + \Theta_1) + n_1(t) - n_2(t) \right| \geq T/2.$$

From this inequality, the rms value of the effective RJ, which combines the effects of the input RJ and the VCO noise, is calculated as:

$$\sigma_{RJ\_eff} = \sqrt{\sigma_{RJ}{}^2 + \sigma_{VCO}{}^2}. \tag{3.23}$$

Note that these two noise sources are assumed to be independent. This σ_{RJ_eff}, instead of σ_{RJ}, can then be applied to Eqs. 3.14 and 3.15 to estimate the BER for Region 2. For the other regions, the BER can be calculated by applying σ_{RJ_eff} instead of σ_{RJ} to Eqs. 3.10), and 3.16–3.19.

3.4 Experimental Results

To validate the proposed estimation technique, we conducted both simulation and experiments based on hardware.

3.4.1 Simulation Results

A behavioral model of a CDR circuit was developed using MATLAB, which operates at 2.5 Gbps with loop parameters specified in Section 3.2.1. In order to reduce the simulation time, we use the cycle-domain model of the CDR circuit [35]. Data with various combinations of RJ and PJ were used as input to the CDR circuit. The BER was then measured by comparing the timing information of the input data and the recovered clock signal. Four different cases were simulated:

Case 1: Clock-like data with jitter satisfying the condition of $\sigma_{RJ}/\sigma_{PJ_{eff}} > 1$;
Case 2: Clock-like data with jitter satisfying the condition of $\sigma_{RJ}/\sigma_{PJ_{eff}} < 1$;
Case 3: PRBS with jitter satisfying the condition of $\sigma_{RJ}/\sigma_{PJ_{eff}} > 1$; and
Case 4: PRBS with jitter satisfying the condition of $\sigma_{RJ}/\sigma_{PJeff} < 1$

In each case, the amounts of PJ and RJ are fixed, and only the frequency of PJ is varied to cover all four regions.

The main difficulty with this validation process is the excessively long simulation time required to capture enough samples to measure a low BER. The measured BER approaches the actual BER as the total number of compared bits approaches infinity. However, if we measure the BER by comparing a finite number of bits, the value will vary depending on the size of the compared bits. The BER can be modeled using a Binomial distribution because there are only two possible outcomes (i.e. error or no error). In most cases, as we are dealing with a low BER and a large number of compared bits, the Poisson distribution can further approximate the error distribution [36].

Suppose α is the expected number of errors at a given BER and N_{bits} is the number of compared bits, then α will be expressed as:

$$\alpha = BER \cdot N_{bits}.$$

Then, the number of errors (N_{err}) has the distribution following the PDF of:

$$P_{poisson}(N_{err}) = \frac{\alpha^{N_{eff}}}{N_{err}!} \cdot e^{-\alpha}.$$

In order to obtain consistent BER measurement results, the ratio of the standard deviation of N_{err} (which is $\sqrt{\alpha}$) to the expected number of errors (α) should be sufficiently small. In our experiment, we captured about 1,000 errors for each case to make sure that this ratio is less than 0.05. By doing that, we can afford simulation only to the 10^{-7} BER level.

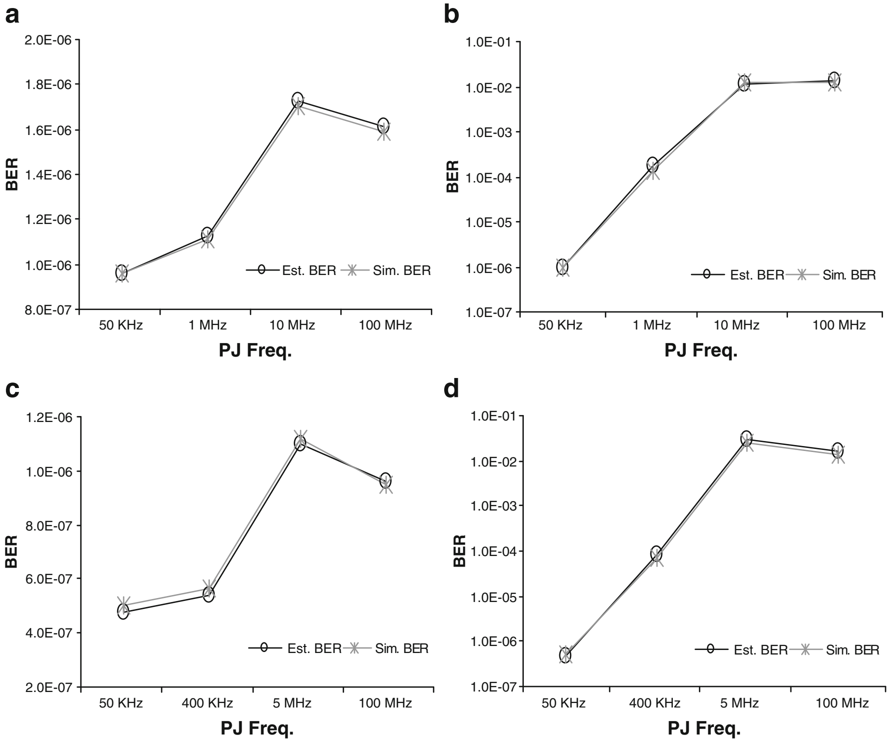


Fig. 3.11 BER simulation results for (**a**) Case 1, (**b**) Case 2, (**c**) Case 3, and (**d**) Case 4 responses, (**b**) for measuring phase responses

Figure 3.11 shows the simulation results for each case. Each graph plots the estimated BER and the simulated BER. For the clock-like pattern, four different frequencies of PJ were injected in each case, which are 50 KHz for Region 1, 1 MHz for Region 2, 10 MHz for Region 3, and 100 MHz for Region 4. The rms value of RJ is fixed at T/9.8, and the peak-to-peak amplitude of the PJ was fixed at 0.06T for Case 1 and 0.6T for Case 2. For the PRBS pattern, we chose 50 KHz for Region 1, 400 KHz for Region 2, 5 MHz for Region 3, and 100 MHz for Region 4. The amplitude of the PJ was set at 0.07T for Case 3 and 0.7T for Case 4. The rms value of RJ was fixed at T/9.8. The simulation results indicate that for all cases, the difference between the estimated BER and the simulated BER is very small.

3.4.2 Hardware Validation Results

We further conducted the experiments using the MAXIM 3873A CDR circuit, which operates at 2.488 Gbps with 2 MHz bandwidth and less than 0.1 dB jitter peaking [37]. This chip was chosen because its specifications are very close to those used in our simulation setup. We used Synthesys Research's BERTScope for jitter injection and for BER measurement.

3.4.2.1 Jitter Transfer Characteristics

We first characterize the CDR circuit's jitter transfer function using the experimental setup shown in Fig. 3.12. For measuring the jitter transfer function, the BERTScope

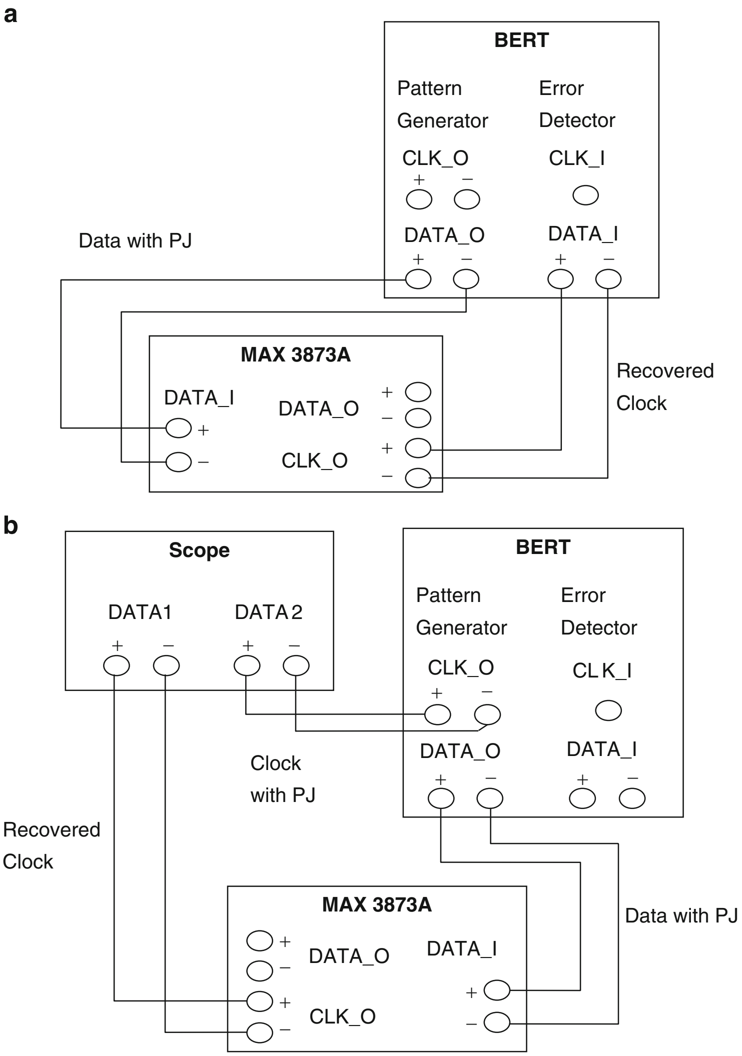


Fig. 3.12 Setup for characterizing the CDR circuit: (**a**) for measuring magnitude responses, (**b**) for measuring phase responses

has the ability to sweep the frequency of the injected PJ and, in turn, measure the magnitude difference between the injected PJ and the recovered clock jitter. However, it does not support phase difference measurements. Thus, Tektronix's Oscilloscope (TDS 6124C) equipped with jitter analysis software was used for the phase measurement. Their jitter analysis package can measure the Time Interval Error (TIE) of two inputs [38]. Thus, we compare the timing difference between the TIE of the Pattern Generator's clock with PJ and the TIE of the recovered clock to measure the phase responses of the CDR circuit. Fig. 3.12(b) shows the phase measurement setup.

Figure 3.13 shows the jitter transfer characteristics of the CDR circuit. Both the clock-like pattern and the $2^7 - 1$ PRBS pattern were used and their magnitude and the phase responses were measured.

We measured the magnitude responses of two extra patterns – periodic 1100 and 11110000 patterns, which have lower transition densities – to assess the effect of the data's transition density on the jitter transfer characteristics. The results show

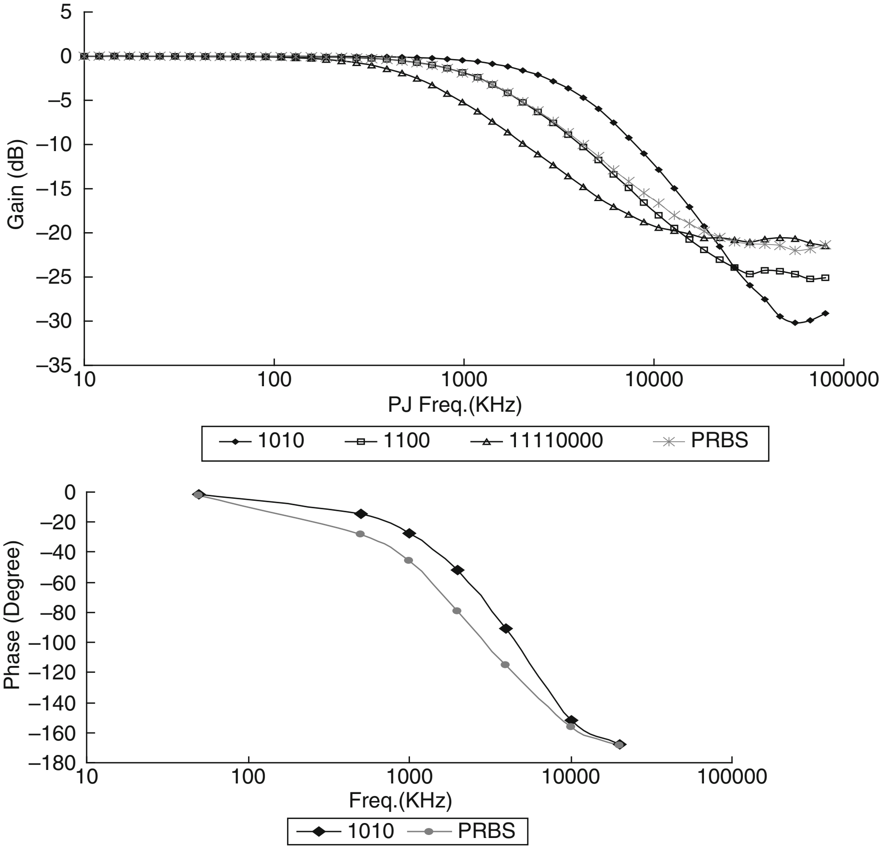


Fig. 3.13 Measurement results for the CDR circuit's jitter transfer characteristics

that the PRBS pattern and the 1100 pattern, whose transition density is equal and is half that of the clock-like pattern, have very similar transfer curves. In the high-frequency range, the PRBS pattern has a higher gain than that of the periodic 1100 pattern. This difference is because the DDJ leaks into the recovered clock jitter for the case of the PRBS pattern. This phenomenon does not make any difference in the low frequency range because the recovered clock jitter is much greater than the DDJ in this range. In addition, the periodic 11110000 pattern, which has a transition density of 1/4, has an even lower bandwidth. These results validated our finding that the jitter transfer characteristic has a strong correlation with the transition density, which was analyzed in Section 3.2.1.

3.4.2.2 BER Measurement Results

After characterizing the CDR circuit, we measured the BER by injecting both PJ and RJ into the data. Figure 3.14 shows the experimental setup. The BERTScope can inject the PJ up to 2.2T, from 1 KHZ to 10 MHz, and 0.5T, from 10 to 80 MHz. In addition, it can inject RJ up to 0.5T at 10^{-12} BER level, which means the rms value of maximum injectable RJ is limited to 0.036T (≈0.5T/14.069) [39]. Due to this limitation that only a small amount of RJ can be injected, we cannot afford to conduct the BER measurement experiments for Cases 1 and 3 (as the BER level would be significantly lower than 10^{-12}). Therefore, the experiments were conducted only for Cases 2 and 4.

We injected PJ at eight different frequencies, ranging from 50 KHz to 80 MHz, to cover all four frequency regions. We fixed the rms value of RJ at 0.036T for all cases. For the clock-like pattern, we injected PJ with two different peak-to-peak amplitudes, 0.5T and 0.45T, and sweeping the PJ's frequency. For the PRBS pattern, we injected PJ with 0.5T peak-to-peak amplitude and sweeping its frequency

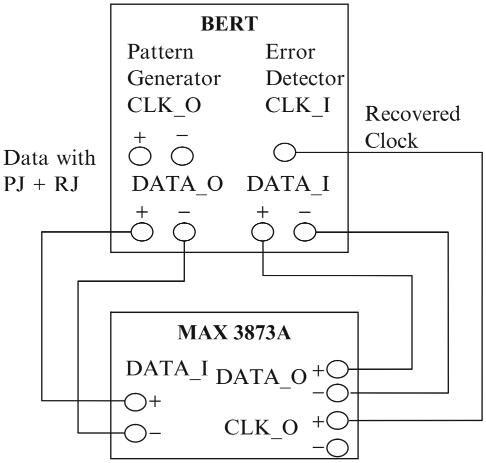


Fig. 3.14 Experimental setup for BER measurements

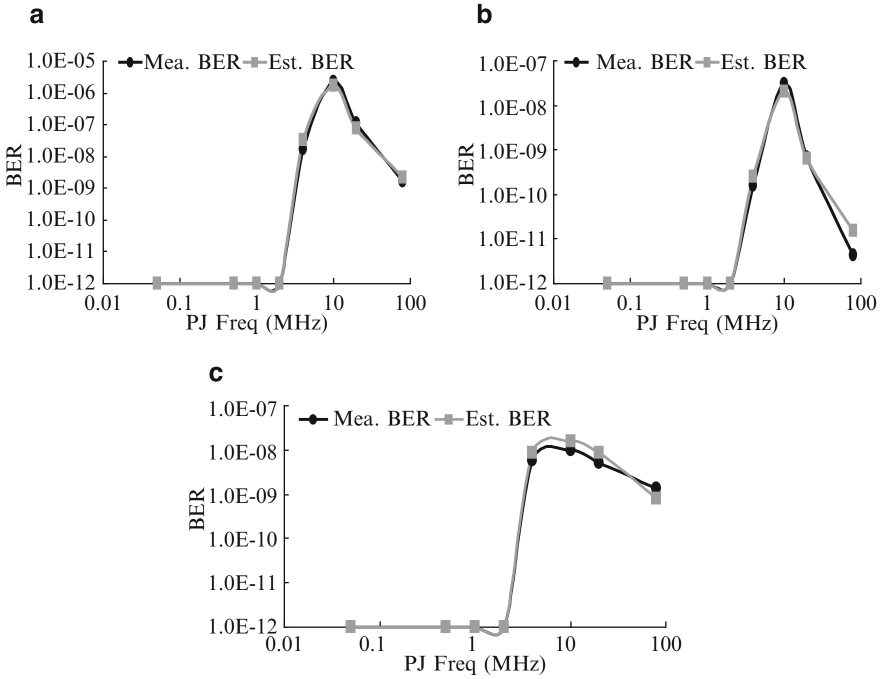


Fig. 3.15 BER measurement results for (**a**) clock-like pattern w/ 0.5T PJ, (**b**) clock-like pattern w/ 0.45T PJ, and (**c**) PRBS pattern w/ 0.5T PJ

too. Figure 3.15 shows the comparison between the measurement results and the estimation results which are calculated based on Eqs. 3.9 and 3.20. We made minor necessary modifications in our estimation: we used the effective rms value of RJ (Eqs. 3.23) instead of using the rms value of the input RJ to incorporate the CDR circuit's intrinsic noise, and we included the extra DJ value caused by the cable into A_{eff} for the PRBS pattern.

In the low frequency range, the expected BER is too low to be measured (i.e. less than 10^{-12}) within a reasonable amount of time due to the CDR circuit's tracking ability. Thus, if no error is captured within a time limit (in our experiment, the limit was set to 7 min $\approx 10^{12}$ b/(2.5 Gbps)), the BER is represented as 10^{-12} as in Fig. 3.15. For these cases, the estimated BER is indeed much lower than 10^{-12}. The results clearly show that the measured BER and the estimated BER match very well in almost all cases. When the measured BER level is around 10^{-12}, the difference is larger than those of other cases as shown in Fig. 3.15b. We suspect that this is because the injected RJ does not have the unbounded tails in real applications – thus the Gaussian model for the RJ may not accurate in the very low probability region. Further investigation is needed to more accurately model the RJ in this region.

The measurement results also clearly indicate that the BER peaks in Region 3 and drops when the PJ frequency reaches Region 4. This validates our analysis that the out of phase phenomenon between the injected PJ and the recovered clock jitter could cause significant degradation in BER performance. Note that this increase in BER is not caused by jitter peaking. This is evident because the device has a less than 0.1dB jitter peaking and the magnitude gain of the jitter transfer function in this frequency range ($\sim$10 MHz) is about -15 dB as shown in Fig. 3.13.

3.5 Summary and Future Work

Jitter has been used for measuring the quality of high-speed serial links, and several standards specify the jitter performance for these links. However, jitter alone would not fully reflect the overall system performance. The spectral information of the jitter and the magnitude and phase responses of the CDR circuit's jitter transfer function should be jointly considered to determine the system performance. In this chapter, we show how the frequencies of the input jitter and the CDR circuit's internal jitter affect the recovered clock jitter and the dependency of the BER on the characteristics of the CDR circuit. The experimental results demonstrate the validity of our analysis and the roles of these parameters in determining the BER.

As the data rates continue to increase, the DDJ, caused by inter-symbol interference, becomes more significant due to the bandwidth limitation of the electrical wires [40]. Various equalization techniques have been proposed to compensate for the DDJ [41–43]. We will extend our work to incorporate the DDJ and some key parameters extracted from the equalizer for the BER estimation.

Chapter 4
BER Estimation for Non-linear Clock and Data Recovery Circuit

In order to expedite the BER testing procedure, in Chapter 3, we developed a BER estimation technique for high-speed serial interfaces that incorporate a linear CDR circuit. However, this technique is not applicable to bang–bang (BB) CDR circuits, which have gained popularity due to their amenities to high-speed serial link applications. Because the BB phase detector in a CDR circuit behaves non-linearly with respect to the input jitter, its jitter transfer function varies significantly with respect to the jitter magnitude [44–46]. Such variations do not exist for a linear CDR circuit. Thus, the jitter transfer dependency on the jitter magnitude has to be considered to accurately estimate the BER for a BB CDR circuit.

Several recent papers studied the characteristics of BB CDR circuits [44–46]. In [44], the jitter transfer's dependency on the input jitter magnitude is first analyzed. Then, the jitter tolerance is examined and the maximum input jitter at a given frequency that a CDR loop can tolerate is predicted. This analysis is only applicable to PJ. Because RJ and DDJ are also present in practice, most standards require all these jitter components to be considered for jitter tolerance testing [4, 15]. Thus, jitter analysis considering only PJ would not be sufficient.

In this chapter, we propose a method for estimating the BER of a BB CDR circuit. We first extend the jitter transfer and tolerance analysis proposed in [44] for the case when the data consist of both PJ and RJ. Then, we derive the BER estimation equation based on this analysis. The equation also uses the jitter spectral information as input, which includes the rms value of the RJ and frequencies and amplitudes of the PJ components.

In the next section, we summarize the jitter analysis for a BB CDR circuit proposed in [44]. Section 4.2 describes the BER estimation technique for a BB CDR circuit when both PJ and RJ are present in the data. Section 4.3 presents the experimental setup and validation results. Section 4.4 concludes the chapter.

4.1 Jitter Analysis for BB CDR Circuits

The block diagram of a typical BB CDR circuit is shown in Fig. 4.1 [44]. It consists of a phase detector, a charge pump, a loop filter and a voltage-controlled oscillator

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_4,

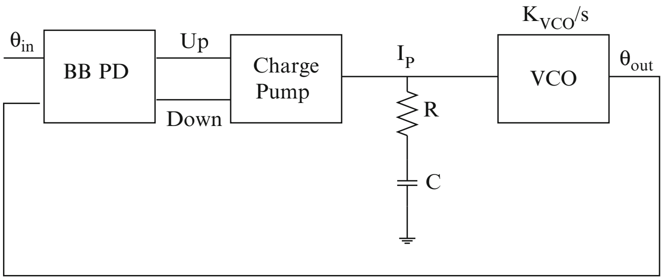
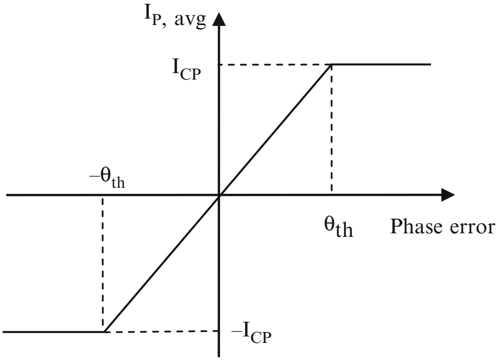


Fig. 4.1 Block diagram of a BB CDR circuit

Fig. 4.2 Average phase detector gain curve



(VCO). The only difference between a linear and a BB CDR circuit is the PD operation. The bang-bang PD discards the magnitude information of the phase error and only measures its polarity while a linear PD's output magnitude is proportional to the phase error.

4.1.1 Jitter Transfer Analysis

Jitter transfer is defined as the ratio of a CDR circuit's output jitter to the input jitter, which is a function of the input jitter frequency. Since the binary PD in a BB CDR circuit exhibits non-linear behavior to the input jitter, conventional linear PLL theory cannot be used for analysis. Figure 4.2 shows the characteristic of the BB phase detector to the input jitter, $\theta_{\text{in}} = \theta_{\text{in},p} \cdot \cos w_{\theta} t$. It operates in the linear region for a small phase error whereas operates in the non-linear region for a large phase error. The ideal characteristic of a BB phase detector should not have any linear operation region. However, the average PD gain is smoothed by the metastability of flipflops in the PD and random jitter in the input data and the VCO, and, thus, has a finite slope across a narrow range of the phase error [44].

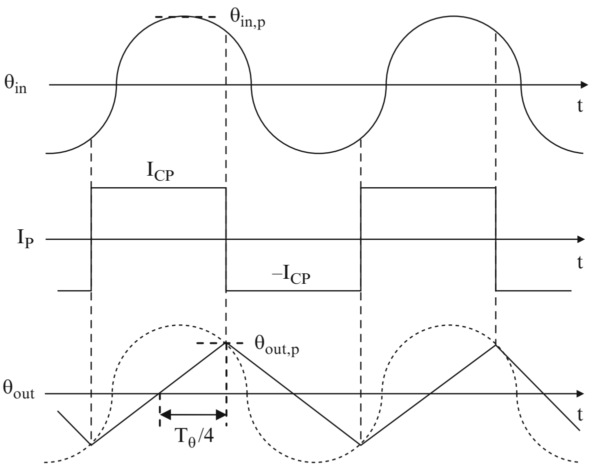


Fig. 4.3 Slewing in a BB CDR loop

At a low jitter frequency, the PD can still track the input jitter closely, and the gain is close to 1. As the jitter frequency increases, so does the phase error. This results in a larger current from the charge pump flowing into the loop filter. Since the available current beyond the linear region (i.e. I_{CP}) is constant, large and fast variation of the input jitter results in slewing. The slewing phenomenon is illustrated in Fig. 4.3. For illustration, we assume an extreme case in which the phase error changes the polarity in every half cycle of the jitter. The charge pump current, I_P, alternates between $+I_{CP}$ and $-I_{CP}$ as a result. Because the loop filter capacitor is typically large, the output of the loop filter tracks I_PR. The output phase thus follows a triangular waveform as shown in Fig. 4.3. The peak value of the output phase occurs after integration of the control voltage for a duration of $T_\theta/4$ ($T_\theta = 2\pi/w_\theta$); that is,

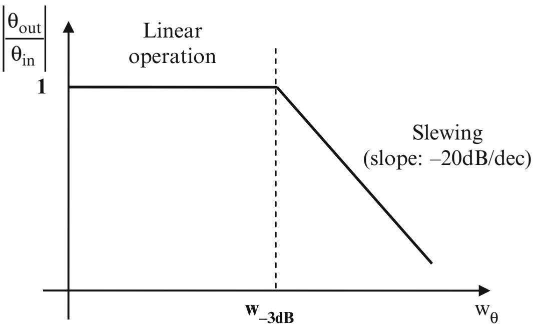
$$\theta_{out,p} = \frac{K_{VCO} I_P R T_\theta}{4}.$$

Thus, the jitter transfer function can be represented as

$$\left|\frac{\theta_{out,p}}{\theta_{in,p}}\right| = \frac{\pi K_{VCO} I_P R}{2\theta_{in,p} w_\theta}. \tag{4.1}$$

Equation 4.1 expresses the dependence of the jitter transfer upon the input jitter magnitude, $\theta_{in,p}$. In addition, the equation also reveals a −20 dB/dec slope in the slewing region. As w_θ decreases, slewing eventually vanishes and the jitter transfer approaches unity, as shown in Fig. 4.4. Extrapolation of the linear and slewing regions yields an approximate value for the −3 dB bandwidth of the jitter transfer, which represents the boundary between the linear operation region and the slewing region:

$$w_{-3dB} = \frac{\pi K_{VCO} I_P R}{2\theta_{in,p}} \tag{4.2}$$



$\left|\frac{\theta_{out}}{\theta_{in}}\right|$
1
Linear operation
Slewing (slope: –20dB/dec)
w_{-3dB}
w_θ

Fig. 4.4 Jitter transfer function of a BB CDR

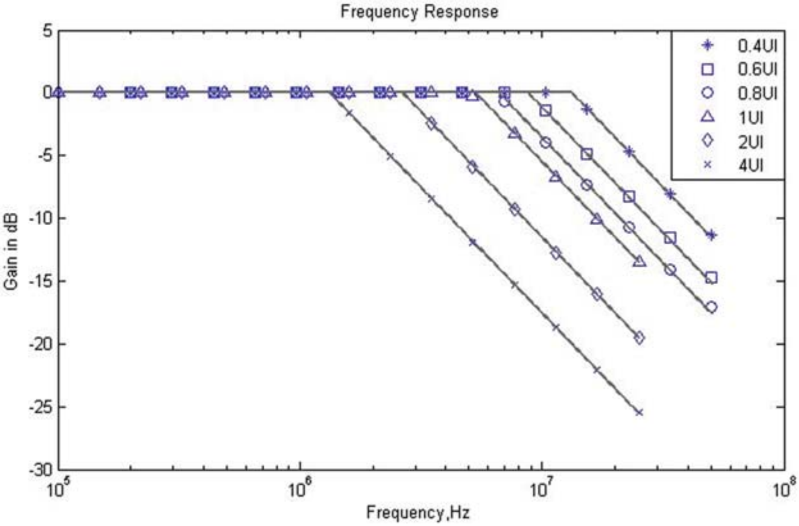


Fig. 4.5 Predicted and simulated jitter transfer

In order to validate the jitter transfer analysis, we constructed a behavioral model of a BB CDR circuit, which operates at 10 Gb/s, and conducted MATLAB simulations. Jitter gain was measured at a number of different frequencies between 100 KHz and 50 MHz with PJ magnitude varying from 0.4 to 4 UI. The solid lines in Fig. 4.5 represent the predicted jitter transfer functions at six different PJ magnitudes, each of which is divided into a unity gain region and a slewing region separated by the −3 dB bandwidth calculated based on Eq. 4.2. The symbols on the lines indicate the jitter transfer gains derived from simulations. For all cases, the simulation results match the theoretical jitter transfer curves very well.

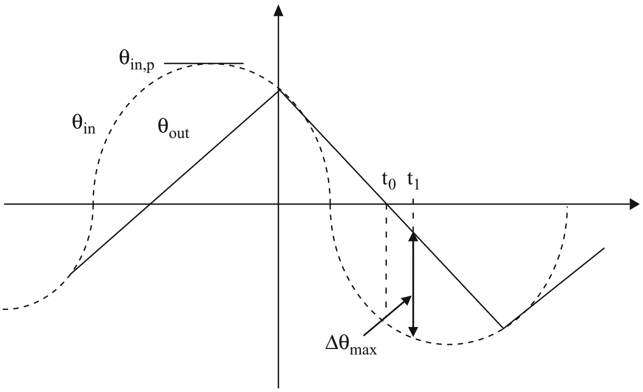


Fig. 4.6 Slewing and maximum phase error

4.1.2 Jitter Tolerance Analysis

Jitter tolerance is the process used to determine the magnitude of input PJ at which a CDR loop starts to introduce errors. As the phase error, $\theta_{in} - \theta_{out}$, approaches π (i.e. half of the unit interval [UI]), the BER increases rapidly. In this sub-section, we summarize the relationship between the input PJ and the maximum phase error. It is important to recognize that a BB CDR loop must slew if it incurs errors.

Figure 4.6 shows an example of the slewing for which the maximum phase error, $\Delta\theta_{max}$ occurs at t_1. The phase error at t_0 is used to approximate the maximum phase error because it is close enough to $\Delta\theta_{max}$ and much simpler to calculate. If θ_{out} slews for most of the period, t_0 is approximately equal to $T_\theta/4$. Denoting the input jitter as $\theta_{in,p}\cos(\omega_\theta t + \delta)$, the maximum phase error could be approximated as:

$$\Delta\theta_{max} \approx \Delta\theta(t_0) = \left|\theta_{in,p} \cdot \cos(\frac{\pi}{2} + \delta)\right|_{\Delta\theta_{max}} = \frac{\sqrt{4w_\theta^2\theta_{in,p}^2 - \pi^2K_{VCO}^2I_P^2R^2}}{2w_\theta} \quad (4.3)$$

The details of the derivation can be found in [44]. Based on Eq. 4.3, we can calculate the maximum tolerable input jitter by expressing $\theta_{in,p}$ in terms of w_θ when the maximum phase error, $\Delta\theta_{max}$, is equal to π.

4.2 BER Estimation

The behavior of a BB CDR loop is well understood when only PJ is present in the data. However, the analysis becomes much more complex when RJ is also present. In the last section, we have shown that the jitter transfer function of the BB CDR loop varies with respect to the PJ magnitude. In the presence of RJ, even with a fixed PJ magnitude, the jitter transfer function will vary for different amounts of RJ. In the following section, we discuss the variation of the jitter transfer function with respect to the rms value of RJ, followed by the derivation of the BER estimation technique.

4.2.1 Variation of Jitter Transfer Due to RJ

If the only jitter component in the data is PJ, the peak-to-peak magnitude of the jitter is fixed and, thus, the dependency of the jitter transfer on the jitter magnitude can be defined. However, when RJ is present in the data, the peak-to-peak magnitude of the jitter becomes difficult to define because the RJ is typically assumed to follow a Gaussian distribution and thus has unbounded characteristic.

In order to observe the RJ's effects on jitter transfer, we obtained jitter transfer gains through simulation by injecting both PJ and RJ. The rms value of the RJ is fixed at 0.1 UI. Three different peak-to-peak magnitudes of the PJ – 0.4, 0.6, and 0.8 UI – are injected with the RJ. Figure 4.7 shows the simulation results. The simulated jitter transfer gains (symbols in Fig. 4.7) were compared to the predicted jitter transfer functions (black dotted lines in Fig. 4.7) based on the analysis described in the last section. There are two distinctive discrepancies between them: the −3 dB bandwidth and the slope in the slewing region. The −3 dB bandwidth derived from simulation is lower than the prediction because the addition of the RJ results in an

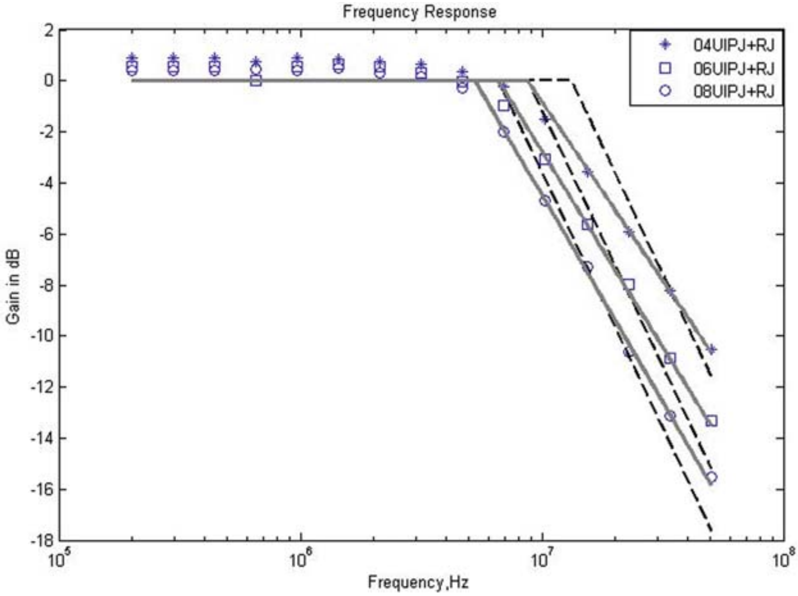


Fig. 4.7 Jitter transfer variations due to RJ

increase in jitter magnitude. To minimize this discrepancy, we apply the magnitude of the total jitter, $\theta_{in,TJ}$, which is defined as the sum of the PJ magnitude and the RJ's rms value (i.e. $\theta_{in,TJ} = \theta_{in,p} + \sigma_{RJ}$), to Eq. 4.2 instead of $\theta_{in,p}$ to take into account the RJ contribution to the jitter magnitude. While RJ has an unbounded characteristic, we use the rms value of RJ for calculating the jitter magnitude. Regarding the slope, the predicted functions, which have −20 dB/dec slope in the slewing region, have a steeper slope than those of the simulated curves. The solid gray lines in Fig. 4.7 are the best-fit lines for the simulated data in the slewing region.

It is interesting to note that the slope of the slewing region depends upon the ratio between PJ and RJ. When only PJ is present in the data, the recovered clock jitter increases monotonically during the first half of the jitter period, as illustrated in Fig. 4.3. The peak-to-peak magnitude of the recovered clock jitter decreases linearly as the frequency of the PJ increases, which results in a −20 dB/dec slope in the slewing region. However, in the presence of both PJ and RJ, the recovered clock jitter no longer increases monotonically. The peak-to-peak magnitude of the recovered clock jitter is less than that of the case in which only PJ is present, as shown in Fig. 4.8.

Therefore, the magnitude of the recovered clock jitter no longer decreases linearly in proportion to the increase in the jitter frequency. This results in a different slope in the slewing region for different PJ to RJ ratios. In order to quantify the slope variations due to the RJ, we conducted behavioral simulation for four different rms values of RJ – 0.025, 0.05, 0.1, and 0.15 UI. The PJ magnitude also varies from 0.4 to 2 UI, generating a wide range of different PJ and RJ combinations. Figure 4.9 shows the simulation results for the slope variations. The x-axis is defined as the ratio of the rms value of the PJ to the sum of the rms values of PJ and RJ (i.e. $\sigma_{PJ}/(\sigma_{PJ} + \sigma_{RJ})$). Because it is very difficult, if not impossible, to derive an

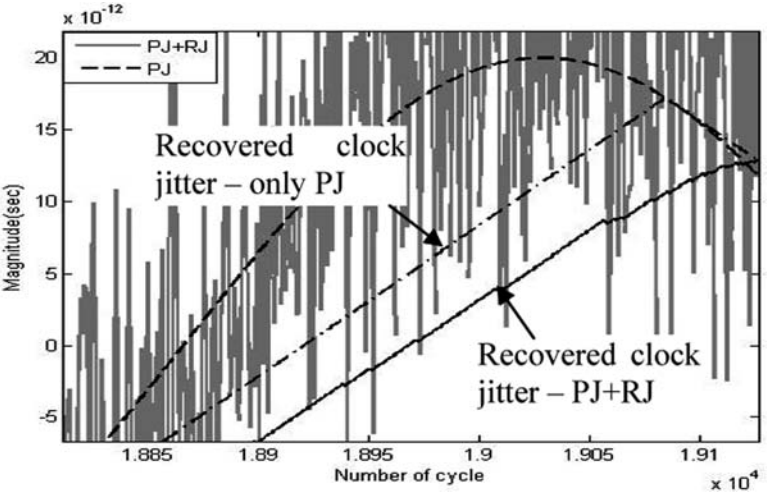


Fig. 4.8 Non-monotonic increase of recovered clock jitter

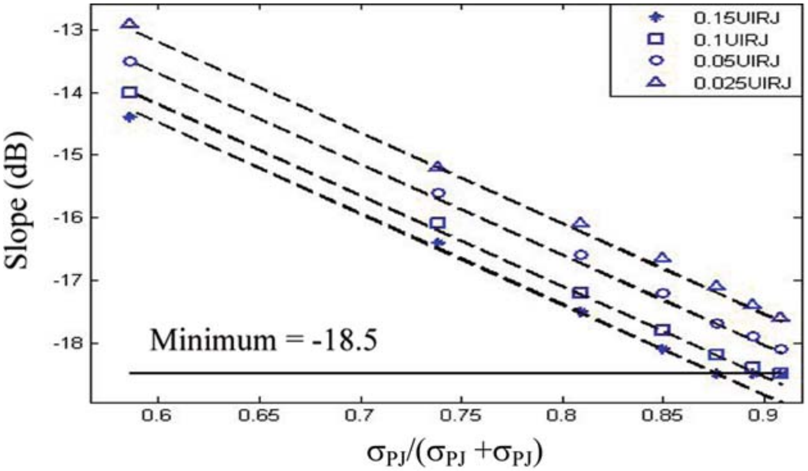


Fig. 4.9 Simulation results of slope variations

analytical relationship between the slope and the jitter magnitude, we empirically derive their relationship from the simulation results as follows:

$$\text{Slope}=\left[\begin{array}{ll} -14.5\cdot\left(\frac{\sigma_{PJ}}{\sigma_{PJ}+\sigma_{RJ}}\right)-5.5+0.5\cdot\log_2\left(\frac{0.1}{\sigma_{RJ}}\right), & \text{if Slope} > -18.5 \\ -18.5, & \text{otherwise} \end{array}\right.$$

4.2.2 BER Estimation

A bit error occurs when the magnitude of the phase error between the input jitter and the recovered clock jitter exceeds half of a UI. Thus, we can estimate the BER once we know the distribution of the phase error between the input jitter and the recovered clock jitter.

When only PJ is present in the data, the phase error distribution can be approximated using a Uniform distribution with a peak-to-peak value of $2^*\Delta\theta_{\max}$. $\Delta\theta_{\max}$ can be calculated from Eq. 4.3. This approximation correctly reflects the bounded property of the phase error distribution and is analytically convenient for BER estimation. When both RJ and PJ are present, the phase error distribution can be approximated by the convolution of a Uniform and a Gaussian distribution because most of RJ components are not tracked by the CDR circuit. Since the addition of the RJ changes the jitter transfer function, we need to modify Eq. 4.3 for calculating the magnitude of the Uniform distribution. To incorporate the bandwidth variation, we simply substitute $\theta_{\text{in,p}}$ in Eq. 4.3 by $\theta_{\text{in,TJ}}$. For the slope variation, we introduce a gain compensating factor that takes into account the gain difference in the slewing region, which is introduced by the addition of the RJ:

$$\mathrm{K_{slope}}(\mathrm{w}_\theta) = \left(\frac{\mathrm{w}_\theta}{\mathrm{w}_{-3\,\mathrm{dB}}}\right)^{\frac{\mathrm{Slope}+20}{20}}$$

We include $\mathrm{K_{slope}}(\mathrm{w}_\theta)$ in the denominator of Eq. 4.3 because the maximum phase error is inversely proportional to the jitter transfer gain. As a result, the maximum phase error becomes

$$\Delta\theta_{\mathrm{max\_TJ}} = \frac{\sqrt{4\mathrm{w}_\theta^2\theta_{\mathrm{in,TJ}}^2 - \pi^2\mathrm{K_{VCO}^2}\,\mathrm{I_P^2}\,\mathrm{R}^2}}{2\mathrm{w}_\theta\mathrm{K_{slope}}(\mathrm{w}_\theta)} \tag{4.4}$$

Knowing the magnitude of the Uniform distribution (i.e. $\Delta\theta_{\mathrm{max\_TJ}}$) and the rms value of the Gaussian distribution (i.e. σ_{RJ}), the PDF of the phase error can be calculated by the convolution of these two distributions:

$$\mathrm{P}_{\Delta\theta}(\mathrm{x}) = \frac{1}{2\mathrm{A_{TJ}}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{RJ}}}\left[\int_{\mathrm{x}-\mathrm{A_{TJ}}}^{\mathrm{x}+\mathrm{A_{TJ}}} \mathrm{e}^{-\mathrm{t}^2/2\sigma_{\mathrm{RJ}}^2}\,\mathrm{dt}\right],$$

where $\mathrm{A_{TJ}}$ is the maximum phase error in UI (i.e. $\mathrm{A_{TJ}} = \Delta\theta_{\mathrm{max\_TJ}}/(2\pi)$). In turn, the BER, which represents the probability that the phase error exceeds half of the UI, would be:

$$\begin{aligned}\mathrm{BER} = 1 - \frac{1}{2\mathrm{A_{TJ}}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{RJ}}}\Bigg[&(0.5 + \mathrm{A_{TJ}})\int_{-\infty}^{0.5+\mathrm{A_{TJ}}} \mathrm{e}^{-\mathrm{t}^2/2\sigma_{\mathrm{RJ}}^2}\,\mathrm{dt} + \sigma_{\mathrm{RJ}}^2\mathrm{e}^{-(0.5+\mathrm{A_{TJ}})^2/2\sigma_{\mathrm{RJ}}^2}\\ &- (0.5 - \mathrm{A_{TJ}})\int_{-\infty}^{0.5-\mathrm{A_{TJ}}} \mathrm{e}^{-\mathrm{t}^2/2\sigma_{\mathrm{RJ}}^2}\mathrm{dt} - \sigma_{\mathrm{RJ}}^2\mathrm{e}^{-(0.5-\mathrm{A_{TJ}})^2/2\sigma_{\mathrm{RJ}}^2}\Bigg]\end{aligned} \tag{4.5}$$

4.3 Experimental Setup and Results

4.3.1 Simulation Setup

We conducted MATLAB simulations to validate the BER estimation technique. The behavioral model of a BB CDR circuit operating at 10 Gb/s was used for the validation. The jitter transfer characteristics were shown in the previous sections. The main bottleneck of the validation process is the excessively long simulation time required to capture enough samples to measure a low BER. In order to reduce the simulation time, we used the cycle-domain model of the CDR circuit [35].

In the experiment, PJ and RJ are injected into the transmitted data, which is then applied to the input of the CDR circuit. The BER is measured by comparing the timing information of the input data to the recovered clock signal. The amplitudes and frequencies of the PJ and the rms values of the RJ were adjusted so that the resulting BER fell within the desired BER levels. Due to the limited simulation time, we could only afford to conduct simulations down to the 10^{-7} BER level. For each case, we conducted the simulation to the point at which 1,000 errors were captured. Then, we compared the error between the estimated BER (i.e. estimated using Eq. 4.5) and the simulated BER (i.e. measured from simulation).

4.3.2 Simulation Results

We chose two different rms values of RJ, 1/10 and 1/8 UI. The peak-to-peak PJ magnitudes were chosen at 0.4 UI (i.e. Case 1) and 0.8 UI (i.e. Case 2) for the 1/10 UI RJ case, and 0.4 UI (i.e. Case 3) and 0.6 UI (i.e. Case 4) for the 1/8 UI RJ case. These cases cover the BER range from 10^{-7} to 10^{-2}. Four different PJ frequencies were chosen for each case, one of which was set to be located in the linear operation region.

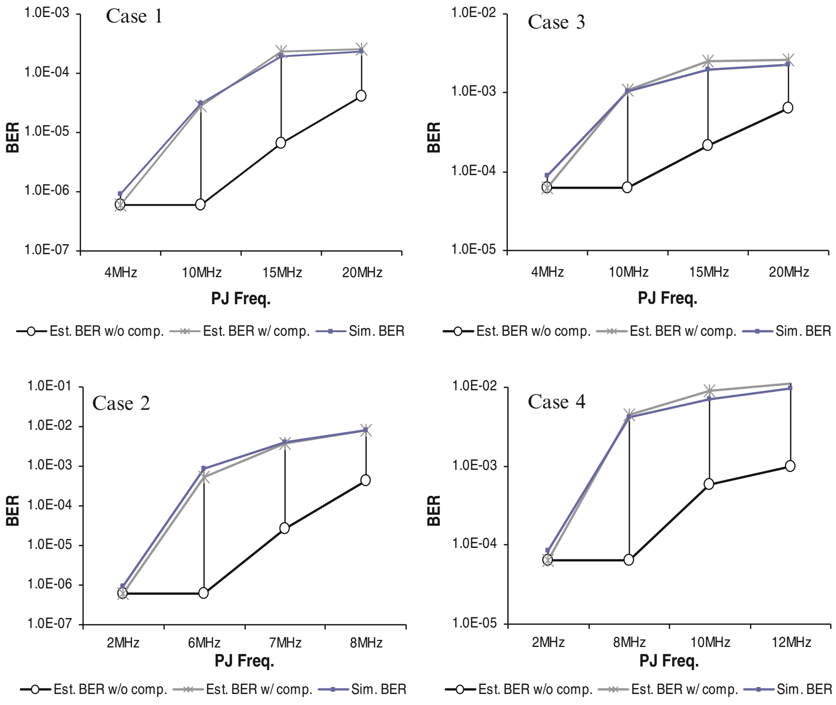


Fig. 4.10 BER simulation results

Figure 4.9 shows the simulation results of these cases. Each graph contains the simulated BER and two estimated BERs: one is the estimated BER without compensating the jitter transfer variations due to the RJ (i.e. Substituting Eq. 4.3 into Eq. 4.5) and the other is the estimated BER with compensation of these effects (i.e. Substituting Eq. 4.4 into Eq. 4.5). The simulation results indicate that for all cases, the simulated BER and the estimated BER, after compensating the jitter transfer variations, match pretty well (see Fig. 4.10).

4.4 Summary

In Chapter 3, we propose a BER estimation technique using the jitter spectrum information and the characteristics of the CDR circuit, which is implemented using a linear PD. In this chapter, we extend the technique using the same set of information for BB clock and data recovery circuits, which exhibit non-linear characteristics with respect to the input jitter. We show the dependency of a BB CDR loop's jitter transfer function on the magnitudes of PJ and RJ. We further propose an analytical technique to estimate the BER. The simulation results demonstrate the validity of our analysis. We will further conduct hardware validation, using suitable measurement equipment to assess the accuracy of the proposed estimation method.

Chapter 5
Gaps in Timing Margining Test

With the challenges of keeping testing costs down, DFT-based testing methods have been pursued [6–9, 47, 48]. The timing margining test is one of the widely adopted DFT methods by companies [8, 9]. Usually facilitated by a loopback on-die or off-die, the idea is to assess the margin in a given I/O's timing and make the pass/fail decision. This test coupled with additional DFT testing methods (e.g. DFT specific to clock data recovery circuitry as noted in [9]) enables computer product manufacturers to ignore the communications style testing, which involves expensive ATE equipment and long testing times. Still, there continues to be one nagging question, "Are we missing anything gross by relying heavily upon timing margining test?"

We decided to study this question and this chapter summarizes our results. The next section gives a primer on timing margining. The core of the chapter explains the testing method's gap, which include random jitter, non-linear clock recovery circuitry, jitter amplification, and duty cycle distortion.

5.1 Timing Margining Test Basics

Defect-based I/O screens using timing margining [8, 47] methodology has been embraced by industries as an alternative to traditional functional timing test. The basic goal of timing margining is to measure the amount of margin within a data eye. For a source synchronous interface such as Intel's Front-Side Data Bus, the distance between the strobe and data eye edges are measured. Ideally, the strobe would be centered on the data eye, and the right/left margin would be $^1/_2$ UI (Unit Interval, ideal data eye) each. In reality, because of jitter, circuit non-idealities, setup/hold time, trace length mismatch, and clock recovery inaccuracy, these margins are reduced and unequal.

Figure 5.1 shows a simplified diagram of a typical timing margining implementation for a source-synchronous data bus. In normal mode, the data is fed from the core and is transmitted to another agent. The strobe is also generated and the delay cell is set to 0. In this mode the strobe should be centered in the data eye. In timing margining mode, we use a loopback configuration which makes the strobe latch its own data back to the inbound section. There, it is compared to the sent data

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_5,

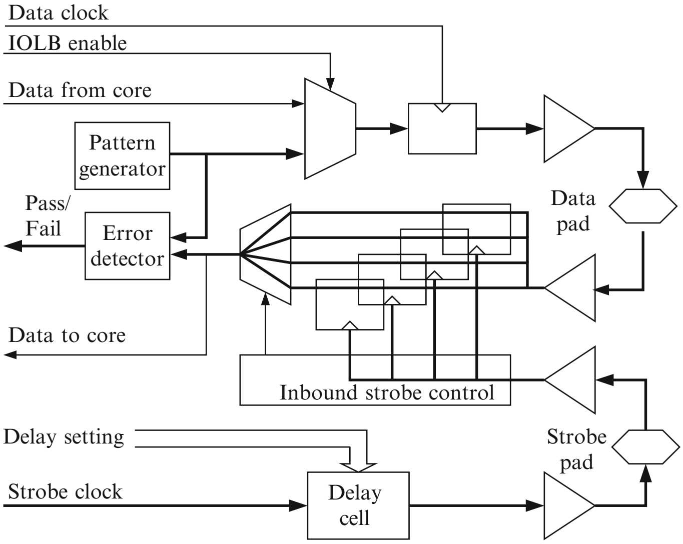


Fig. 5.1 Timing margining general approach for source synchronous pins

and a pass/fail decision is made. By moving the strobe with respect to the data, the delay cell allows a timing stress. In addition, this movement combined with monitoring the pass/fail result, the margin between the original strobe position and the data edge can be found. This assumes that the relationship between the strobe delay setting and the actual delay is known and reliable.

The same concept can be applied to various single-ended I/O configurations such as common-clock, DDR, and CMOS, as well as serial interfaces such as Serial ATA, FBD, PCI Express. Most serial interfaces already have a phase interpolator (PI) in the receiver circuitry, which can be re-used to perform timing stress on the received data eye. Timing margining can also be applied in a non-loopback configuration by sending data from one chip to another chip, if a pattern comparison protocol, such as IBIST [49], has been implemented.

5.2 Gap Analysis in Timing Margining Test

The timing margining test has been successfully deployed for testing Front Side Bus at Intel [8, 47] and at other companies [9]. However, it is uncertain that the timing margining test can completely replace the conventional jitter testing methods, such as jitter measurement and jitter tolerance, for high-speed serial interfaces running at multiple Gbps. In this section, we investigate possible sources that might generate some gaps between the timing margining test and the conventional jitter test.

Table 5.1 Relationship between BER and Q_{BER}

Q_{BER}	BER	Q_{BER}	BER
3.7	10^{-4}	6.4	10^{-10}
4.8	10^{-6}	7.0	10^{-12}
5.6	10^{-8}	7.9	10^{-15}

5.2.1 Random Jitter

Jitter is commonly divided into DJ and RJ. One of the motivations for separating jitter into these classifications is to extrapolate system performance without direct, time-consuming measurement. DJ is bounded and represented by a peak-to-peak value, while RJ is unbounded and its peak-to-peak value highly depends on the measurement time. Since RJ is best described by a Gaussian distribution, it is commonly represented by an rms value. Then, total jitter (TJ) at a certain BER level is estimated using the peak-to-peak value of DJ (DJ_{p-p}) and the rms value of RJ (RJ_{rms}) as follows [27]:

$$TJ(BER) = DJ_{p-p} + 2 \cdot Q_{BER} \cdot RJ_{rms} \tag{5.1}$$

where Q_{BER} is determined by the targeted BER as shown in Table 5.1.

However, timing margining test treats DJ and RJ in the same manner, that is, it does not separate the DJ and the RJ. In addition, it measures the margin within a few data cycles – significantly less than 10^{12} cycles – even though most specifications require a BER of 10^{-12} or lower. Since the RJ is unbounded and its peak-to-peak value greatly depends upon the measurement time, the timing margin measurement from a small number of cycles likely underestimates the RJ at the 10^{-12} BER level. Such an underestimation would thus result in an optimistic BER estimation.

The bathtub plot is introduced to illustrate this phenomenon. The bathtub plot is generated by gradually stressing the sampling time – from −0.5 to 0.5 UI – and by measuring the corresponding BER. Figure 5.2 illustrates parts of two different bathtub plots (the x-axis varies from 0 to 0.5 UI): one shows when DJ is dominant (black curve), and the other shows when RJ is dominant (gray curve). At a 10^{-4} BER level, for example, the corresponding timing margin when RJ is dominant (2) is greater than that of the DJ dominant case (1). On the other hand, at a 10^{-12} BER level, the phenomenon is completely different. The eye is completely closed when RJ is dominant, but the eye is still open when DJ is dominant.

As expected from Eq. 5.1, the eye-opening or total jitter in the DJ dominant case has a very small variation depending on the BER level which also represents the number of captured samples used for error detection. On the other hand, the eye-opening for the RJ dominant case has a strong relationship with the targeted BER.

Thus, when RJ is only a small portion of the total jitter, the timing margin result from a few thousand samples is not much different from the result at the 10^{-12} BER level. This is the situation for the targeted systems with copper traces of known length. However, as the data rate keeps increasing and approaches the tens of gigabits per second range, a few picoseconds of RJ can have a significant impact on

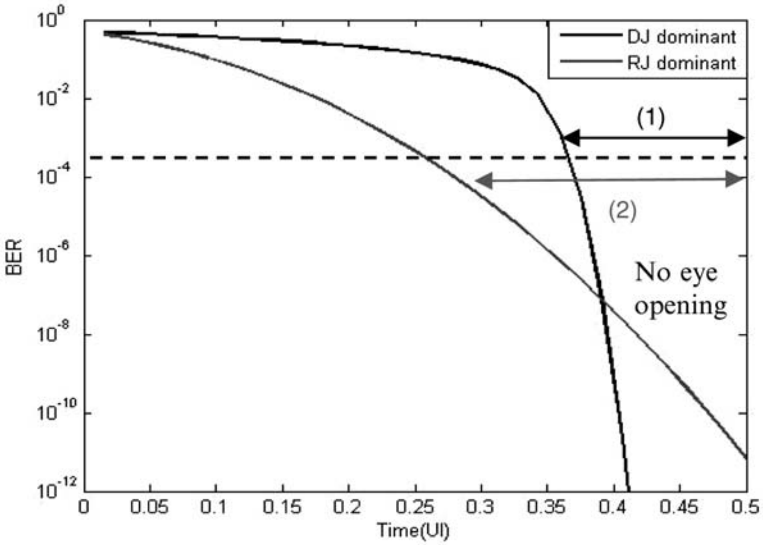


Fig. 5.2 Eye opening comparison

the timing margining result. In order to cope with this problem, timing margining can be performed at two or more different BER levels to separate the RJ and the DJ.

5.2.2 PLL-Based Clock Recovery with Non-linear Phase Detector

Non-linear behaviors of a system can also generate the gaps between the two testing methods. For example, we can observe either no or a very small amount of eye closure when the system has less than a certain amount of jitter. However, a slight increase in jitter can suddenly close the eye due to the non-linear characteristics of the system, instead of gradually closing it. In this situation, the timing margin is not correlated with the jitter margin. An example of this non-linear behavior is a PLL-based clock recovery circuit with a bang-bang phase detector (PD).

As discussed in Chapter 4, the bang-bang PD discards the magnitude information of the timing error and only measures the polarity of the timing error. Since the output of the bang-bang PD is constant regardless of the phase error, it can be greater for small errors, or less for large errors than the output of the linear PD as shown in Fig. 5.3. In other words, if the phase error is small ($\O_1$ in Fig. 5.3), the bang–bang PD has greater gain than the linear PD. On the other hand, if the phase error is large ($\O_3$ in Fig. 5.3), the bang–bang PD has less gain than the linear PD.

Since the effective gain of the bang–bang PD strongly depends on the phase error (i.e. input jitter magnitude), the loop dynamics of the clock recovery circuit with bang–bang PD also varies with the input jitter magnitude [44–46]. In other words,

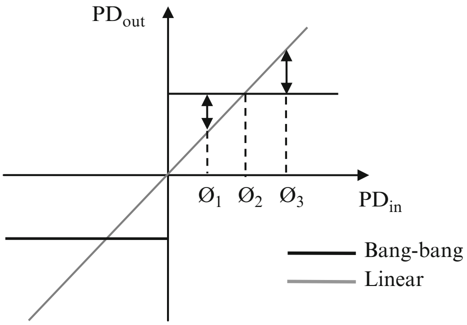


Fig. 5.3 Phase detector gain curves

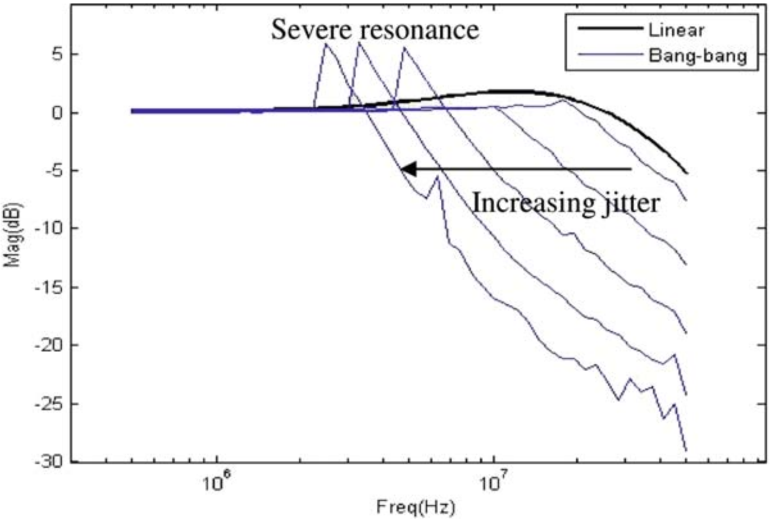


Fig. 5.4 Jitter transfer curves

the loop bandwidth decreases as the input jitter increases because of the decrease in the effective PD gain. In addition, jitter peaking suddenly increases when the jitter magnitude exceeds a certain value because the PD output can have a long string of either +1s or −1s, as shown in Fig. 5.4 [50]. This peaking causes severe resonances at certain frequencies, and thus the system can be broken abruptly once the jitter magnitude surpasses the threshold.

Bathtub curve simulations were conducted to validate this phenomenon. The behavioral models of PLL-based clock recovery circuits are implemented using MATLAB with either linear PD or bang–bang PD. The loop parameters were chosen to reflect the jitter transfer characteristics shown in Fig. 5.4. Then, we injected sinusoidal jitter with 2 MHz frequency into the circuits, and plotted the bathtub curves by gradually increasing the magnitude of the sinusoidal jitter. For the linear clock recovery circuit, we varied the magnitude of the jitter from 3 to 5 UI with a 0.4

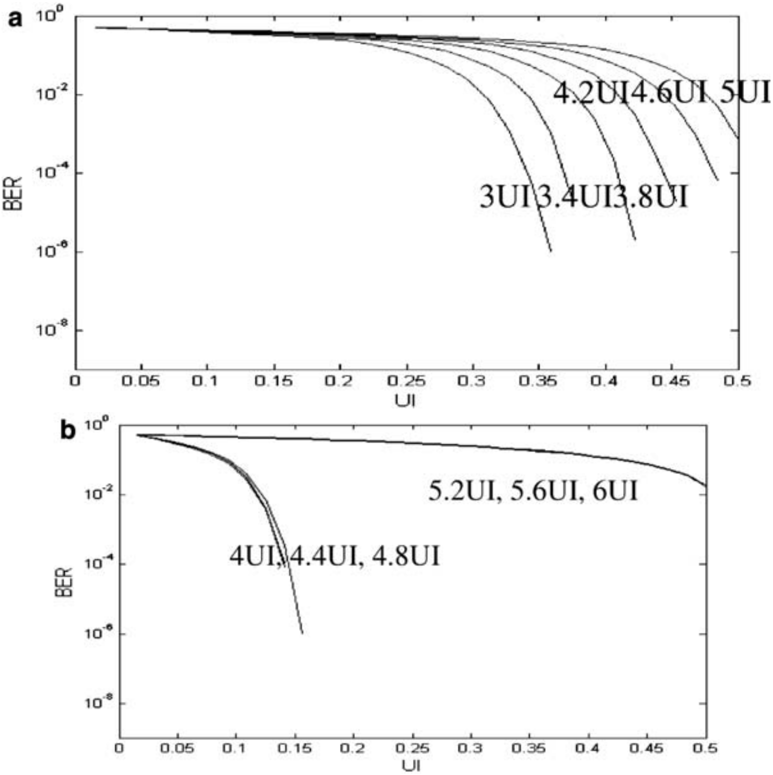


Fig. 5.5 Bathtub curve comparison: (**a**) for linear clock recovery circuit, (**b**) for non-linear clock recovery circuit

UI step size. For the non-linear clock recovery circuit, the magnitude of the jitter is varied from 4 to 6 UI with the same step size.

Figure 5.5 represents how the bathtub curves vary with the jitter magnitude for both the linear and the non-linear clock recovery circuits. The non-linear case suddenly closes the eye when the jitter exceeds a certain level, while the other closes the eye linearly as jitter increases.

These results indicate that for non-linear clock recovery circuits, the timing margining test alone might not detect how close the system is to the breaking point. In other words, even though the eye-opening passes within a high volume manufacturing (HVM) testing environment, the system might not work in real applications if the jitter level is very close to the breaking point. Thus, jitter injection might be required on top of the timing margining test to address this problem, especially when the system has a non-linear clock recovery unit.

5.2.3 Jitter Amplification

As a HVM testing environment is different from a platform environment, the test criteria is usually determined more stringently with respect to the device specification by having some guardbanding. Typical timing margining test conditions are cleaner than those in the platform because there are no power/substrate noises from other devices in the test environment and the loopback channel is shorter. If the jitter caused by additional sources is simply added to the intrinsic jitter of the device under test (DUT), guardbanding to the test specification can guarantee the operation of the device in the platform. However, if jitter is amplified through some circuitry or channel, the simple guardbanding might not always ensure the proper screening of bad devices in HVM testing.

Jitter amplification mainly occurs due to the bandwidth limitation of devices (e.g. clock buffer, delay cell, or channel). Specifically, the gain difference between the signal and the jitter component causes jitter amplification. Frequency domain analysis can provide insight on jitter amplification. The clock waveform with jitter, $\phi(t)$, can be modeled by the phase modulated sinusoid [51]:

$$s(t) = A_c \cos(2\pi f_c t + \varphi(t))$$

When the magnitude of the jitter is small, the above equation can be approximated as follows:

$$s(t) = A_c \cos(2\pi f_c t) + A_c \sin(2\pi f_c t) \cdot \varphi(t)$$

If we denote the jitter as $\varphi(t) = A_j \cdot \cos(2\pi f_j t)$, s(t) become

$$s(t) = A_c \cos(2\pi f_c t) + \frac{A_c A_j}{2} \left\{ \sin(2\pi(f_c + f_j)t) + \sin(2\pi(f_c - f_j)t) \right\} \qquad (5.2)$$

Figure 5.6 represents the frequency spectrum of the jitter components based on Eq. 5.2. As can be seen, the low frequency component of the jitter is located close

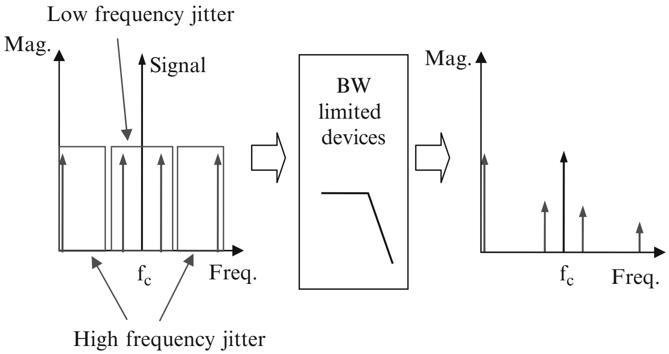


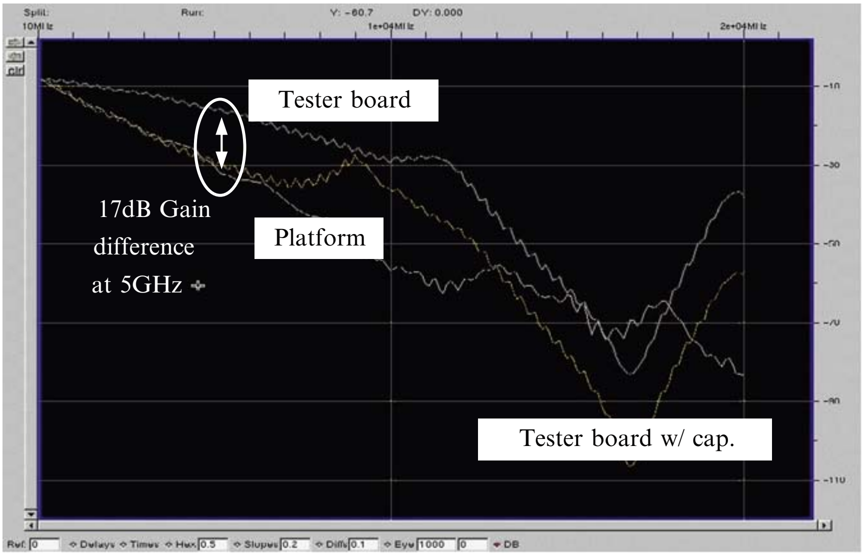
Fig. 5.6 Frequency spectrum of jitter

to the signal, while the high frequency component of the jitter is located relatively farther away. Thus, when the signal passes through a bandwidth limited device, the signal and the jitter component are attenuated differently. The high frequency jitter in particular, which is located in the low frequency region in the spectrum, is not attenuated as much as the signal component. This relative gain difference causes jitter amplification, which can be expressed as Eq. 5.3 for a linear channel H(f) [52].

$$\mathrm{Jitter\_Amp} = \frac{(\mathrm{H}(\mathrm{f_c} + \mathrm{f_j}) + \mathrm{H}(\mathrm{f_c} - \mathrm{f_j}))}{2\mathrm{H}(\mathrm{f_c})} \tag{5.3}$$

There are several jitter sources in the system, such as power supply noise, thermal noise, and the duty cycle distortion (DCD). The power supply noise, which usually relates to package resonance frequency, is typically on the order of a hundred megahertz. Since random jitter is distributed in the wide frequency range, most of the jitter is attenuated with the signal. Among the jitter sources, DCD has the highest frequency component, whose frequency is the same as the clock signal. Thus, the DCD results in the greatest jitter amplification.

In order to quantify the jitter amplification difference between the testing and the platform environments, we compared the frequency responses of the tester board channel and the platform channel as shown in Fig. 5.7. The tester board channel has −13 dB gain at 5 GHz, while the platform channel has −30 dB gain at the same frequency. If we calculate the jitter amplification factor based on Eq. 5.3 for a 5 GHz clock, the tester channel amplifies the DCD 2.4 times while the platform channel amplifies it 16 times. This huge difference in jitter amplification can cause a significant gap in the timing margining between the two different environments. For the



Y: -60.7 DY: 0.000

Ref: 0 ◇ Delays ◇ Times ◇ Hex 0.5 ◇ Slopes 0.2 ◇ Diffs 0.1 ◇ Eye 1000 0 ◆ DB

Fig. 5.7 Frequency responses of channels

data loop-back channel, the jitter injection filter can be used to emulate the system channel [22]. This can be extended to the clock channel for source-synchronous interfaces to mimic the jitter amplification effect.

5.2.4 Duty Cycle Distortion in Clock

In general, since data is widely spread out through the frequency range from DC to half of the data rate, the expected jitter amplification through the data would be less than that of the clock. In addition, most of the deterministic jitter components in the data, such as inter symbol interference (ISI) and DCD, will be captured in the timing margining test because it directly closes the eye-opening. However, DCD in the clock does not directly close the eye-opening and thus it might not be captured by the timing margining test. We investigated the impact of DCD in the clock on the eye-opening and the timing margining results.

If the clock has DCD, the clock recovery circuit in a receiver, which is commonly implemented by a delay-locked loop (DLL)/PLL, may or may not detect the DCD depending on the architecture of the phase detector. Since many PD architectures used in DLLs/PLLs are designed to be insensitive to the DCD [53, 54], the DCD in the clock is directly transferred to the output of each delay cell. In order to control the phase offset for the timing margining test, the output of each delay cell is connected to the PI. Then, the PI chooses two adjacent phases to generate the main clock by mixing the phases. The number of delay cells and the resolution of the PI greatly depend on architectures. In our analysis, the DLL is assumed to have four delay cells to generate eight different phases with 45° separation as shown in Fig. 5.8. In addition, the phase mixer is assumed to generate seven steps within a 45° range, which gives a 6.4°(=45°/7) phase resolution.

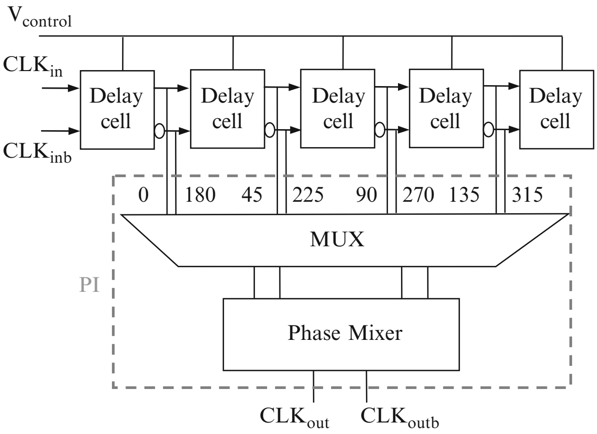


Fig. 5.8 Delay cells and phase interpolator

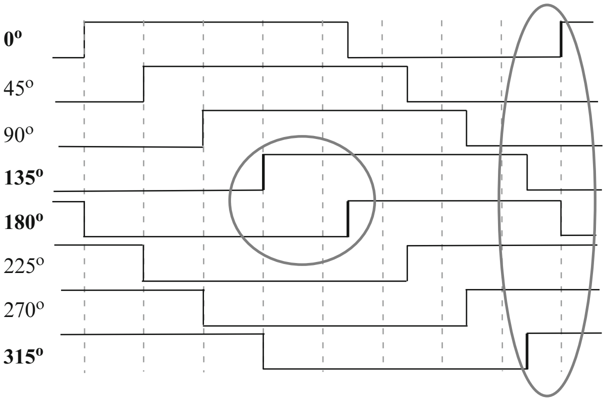


Fig. 5.9 Timing diagram of DLL output with DCD

When the clock has DCD, all eight phases from the delay cells have the same amount of DCD because the phase detector does not detect the DCD. However, the phase separations between adjacent phases are not all the same, as shown in Fig. 5.9. In other words, the phase separations between true (complementary) signals are not affected by the DCD. On the other hand, the phase separations between a true signal and a complementary signal (e.g. 135° and 180°, 315° and 0°) are affected by the DCD. One of them (135° and 180°) has a greater phase separation, while another one (315° and 0°) has less phase separation.

Therefore, if the PI chooses a true signal and a complementary signal for the clock generation, the step size will be narrower/wider than the ideal. This non-ideal step size can affect the timing margining results, as shown in Fig. 5.10. We analyzed this effect for three different cases:

Case 1: The region which has wider phase separation is used to measure the timing margin.

Case 2: The region which has narrower phase separation is used to measure the timing margin.

Case 3: Non-ideal regions are located in the jittery region (i.e. only ideal regions are used to measure the timing margin).

For Case 1 (shown in Fig. 5.10a), the timing margin result is decreased compared to the case when the clock does not have the DCD, while it is increased for Case 2, as shown in Fig. 5.10b. For Case 3 (shown in Fig. 5.10c), the timing margining result is not affected by the DCD. Since the timing margining results can vary case by case, the DCD in the clock may not be detected by the timing margining test.

In addition, the testing can give wider eye-opening results than the original eye-opening. In some cases (i.e. Case 2), this might result in passing bad devices in HVM testing. In order to address this problem, the duty cycle detection/correction circuitry should be required to minimize the duty cycle error in the clock. This is becoming more common place.

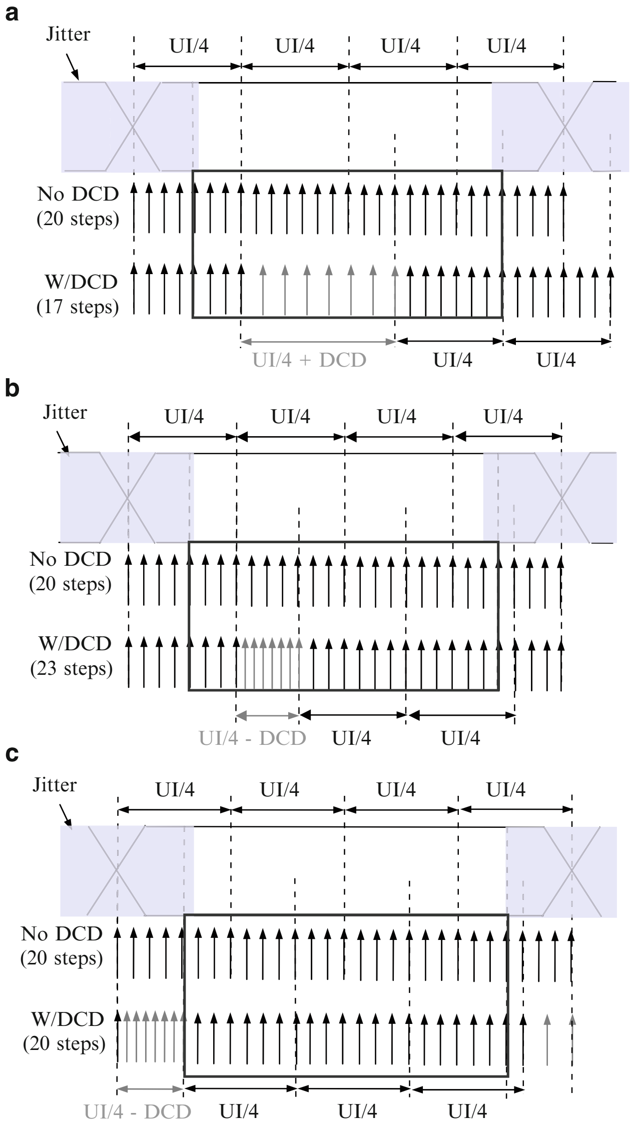


Fig. 5.10 Comparisons of the timing margining measurement: (**a**) Case 1, (**b**) Case 2, and (**c**) Case 3

5.3 Summary and Future Work

Realizing that switching from one test method to another may result in gaps; we explored possible misses by relying upon timing margining test alone. By comparing against standard jitter measurements, we determined that RJ dominant systems are not covered by a single timing margining measurement. However, we note that computer products are dominated by DJ and we could augment the current testing approach by taking two or more measurements for different BER levels. Depending upon the type of data clock recovery circuitry, there are test vulnerabilities. Non-linear CDR circuitry, which is based on bang-bang PLL implementations, can have abrupt changes in data eyes with increasing jitter which would not be covered by timing margin test. Hence, their inherent sensitivity would require additional test coverage. While jitter injection is one means to address, an area for future work is to develop a DFT-based methodology for non-linear CDR circuitries. With respect to jitter amplification, the noted differences between the end-use system and the test system results in potential gaps for which others have suggested techniques [49]. Finally DCD can adversely affect the timing margining measurement itself. A preventive measure such as duty cycle detection and correction circuitry could be implemented.

Chapter 6
An Accurate Jitter Estimation Technique

One of the popular and efficient approaches for testing a high-speed transceiver is based on the loopback principle. Even though the BER can be measured in the loopback mode using a pattern generator and an error detector in a transceiver, only the I/O performance margin (e.g. timing margin) is guaranteed in production testing due to excessively long testing time for low BER measurement [8, 9]. However, as we illustrated in Chapter 5, such a test might not detect some chips that fail the BER specification when the RJ accounts for a non-trivial fraction of the TJ.

In this chapter, we introduce a new technique for estimating TJ, which can be incorporated into the existing loopback-based test method without a significant increase in testing time. An estimation technique based on the dual-Dirac model has been widely used. This technique intends to quickly predict the TJ at low BER levels based on jitter or BER measurement data taken at higher BER levels. However, the technique has a few problems: (1) The estimation accuracy decreases as the deterministic jitter (DJ) characteristics become increasingly complex due to ISI and crosstalk. (2) The estimation accuracy is extremely sensitive to the BER region which the prediction is based upon. In order to cope with these problems, we propose a high-order polynomial fitting technique that can accurately estimate the TJ down to a 10^{-12} BER level using the information from a higher BER region.

In the next section, we first describe the DJ characteristics due to ISI and crosstalk, which are the major contributors to the DJ for multi-Gbps links. Section 6.2 presents the limitations of the estimation technique based on the dual-Dirac model, followed by the description of the proposed technique. Section 6.3 concludes the chapter.

6.1 Characteristics of DJ

DJ can be divided into DDJ and PJ. PJ can be easily characterized by sinusoidal functions. Modeling DDJ is much more difficult because it is highly data-pattern dependent. In this section, we analyze the characteristics of DDJ by assuming the application of commonly used PRBS patterns. ISI and crosstalk are two major sources of DDJ. The bandwidth limitation of a channel generates significant ISI

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*, Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_6,

due to an increased data rate. In addition, increasing the interconnect density makes the crosstalk between lanes a significant contributor to DDJ. In the following, we describe how to characterize both ISI- and crosstalk-induced jitter.

6.1.1 ISI-Induced Jitter

Channel bandwidth limitation causes ISI. Channel loss due to skin-effect and dielectric loss causes signal dispersion, and thus the signal is distorted by its own delayed versions. In other words, the signal is affected not only by the current bit but also by a number of previous bits. For each signal transition, the previous bits shift the signal amplitude, which, in turn, changes the relative time in which the signal crosses a decision threshold. The amount of timing deviation is determined by the data sequence and the channel bandwidth. The channel bandwidth affects the memory depth of the channel, which determines the number of previous bits that would have an impact on the threshold crossing time. If the bandwidth is low, more bits affect the crossing time. As described in [55], if only the penultimate bit has a non-negligible effect, there are two distinct sets of sequences of interest – 010 and 101, for which the signal transition would arrive early, and 001 and 110, for which the signal transition would arrive late. Thus, there are only two delta functions in the jitter histogram. If the signal speed increases and the relative channel bandwidth decreases, more previous bits affect the crossing time. Then, the DDJ jitter histogram includes more delta functions. Since a PRBS pattern has a certain property, we can easily derive the DDJ characteristic resulting from the ISI. Figure 6.1a illustrates this property of a 4-bit PRBS pattern: for each memory depth, the PRBS pattern has the same number of occurrences for each distinct set of sequences. Because of this property, it results in the same probability for each delta function in the jitter histogram. If the bandwidth limited channel causes a memory depth of n bits, the jitter histogram would have 2^{n} delta functions, each of which has an identical probability. Thus, ISI-induced jitter can be approximated as a uniform distribution, as illustrated in Fig. 6.1b.

6.1.2 Crosstalk-Induced Jitter

Crosstalk is the energy coupling from one trace to another, which could be caused by either capacitive or inductive coupling between adjacent lanes. Figure 6.2a illustrates three different crosstalk modes between the coupled transmission lines [56]: odd, superposition, and even modes. In the odd mode, the data transition in the aggressor is opposite to that in the victim. In the even mode, the data transitions are the same in both the aggressor and victim. In the superposition mode, the aggressor does not have any data transition, and thus there is no crosstalk. If there is only one a

4-bit PRBS pattern: 000111101011001

Memory depth	Patterns	# of occurrences
1	**0**01 / **1**10	4
	101 / **0**10	4
2	**00**01 / **11**10	2
	0101 / **10**10	2
	1001 / **01**10	2
	1101 / **00**10	2

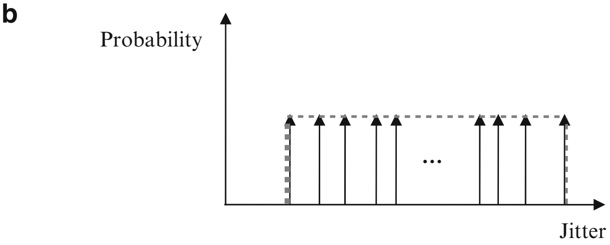


Fig. 6.1 Characteristics of ISI induced jitter: (**a**) edge pattern probability for a 4-bit PRBS pattern, (**b**) jitter distribution due to ISI resulting from an n-bit PRBS pattern

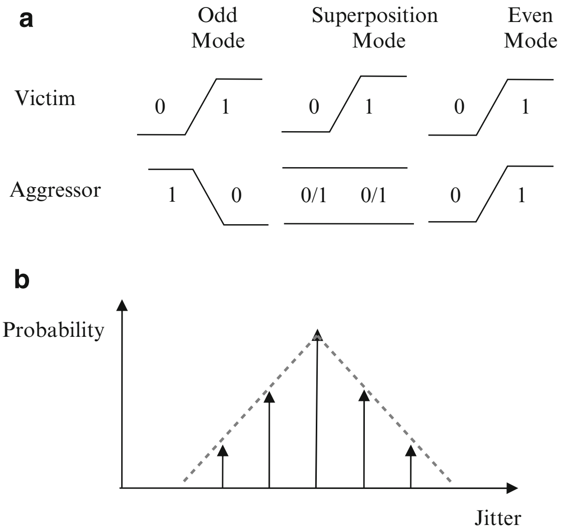


Fig. 6.2 Characteristics of crosstalk induced jitter: (**a**) different modes of crosstalk, (**b**) jitter distribution due to crosstalk

aggressor line in a system, the PDF of the jitter due to crosstalk consists of three delta functions, represented in [56] as:

$$\mathrm{PDF_{Xtalk}} = \frac{1}{4}\delta(\mathrm{J_{Xtalk}}) + \frac{1}{2}\delta(0) + \frac{1}{4}\delta(-\mathrm{J_{Xtalk}}), \tag{6.1}$$

where $\mathrm{J_{Xtalk}}$ is the jitter amount due to crosstalk. The probability of having no crosstalk (i.e. the superposition mode) is twice the probability of getting speed-up (the even mode) or slow-down (the odd mode). The analysis can be easily extended for a case in which a victim trace is sandwiched between two aggressor traces. The jitter PDF can be derived by the convolution of Eq. 6.1

$$\mathrm{PDF_{Xtalk}} = \frac{1}{16}\delta(2\mathrm{J_{Xtalk}}) + \frac{1}{4}\delta(\mathrm{J_{Xtalk}}) + \frac{3}{8}\delta(0) + \frac{1}{4}\delta(-\mathrm{J_{Xtalk}}) + \frac{1}{16}\delta(-2\mathrm{J_{Xtalk}})$$

The resulting PDF has five delta functions, as shown in Fig. 6.2b. In general, the jitter histogram due to crosstalk has a higher probability around the middle (i.e. no or little jitter) and the probability decreases as jitter increases [57]. Therefore, we can approximate the jitter histogram as a triangular distribution.

6.2 Total Jitter Estimation

6.2.1 Estimation Based on Dual-Dirac Model

The dual-Dirac model has been widely adopted to separate DJ and RJ for estimating TJ at low BER levels. In this model, DJ is approximated by two delta functions and RJ is assumed to have a Gaussian distribution that is represented by an rms value.

In order to easily separate DJ and RJ from the measured bathtub curve, the vertical axis of the plot is changed from the BER to the Q-scale. The advantage of using the Q-scale is that a Gaussian jitter distribution becomes a straight line in the Q domain. Therefore, a simple first-order line fitting can separate RJ from DJ when an appropriate fitting region is chosen. The relationship between the Q-function and the BER can be represented as [27]:

$$\mathrm{Q(x)} = \sqrt{2}\,\mathrm{erfc}^{-1}(\mathrm{BER(x)}), \tag{6.2}$$

where the complementary error function is given by

$$\mathrm{erfc(x)} = \frac{1}{\sqrt{2\pi\sigma}}\int_{x}^{\infty}\exp(-\frac{t^2}{2\sigma^2})\mathrm{dt}$$

By measuring the BER at different positions in the eye and deriving their corresponding Q values using Eq. 6.2, the bathtub curve can be plotted using the Q-scale.

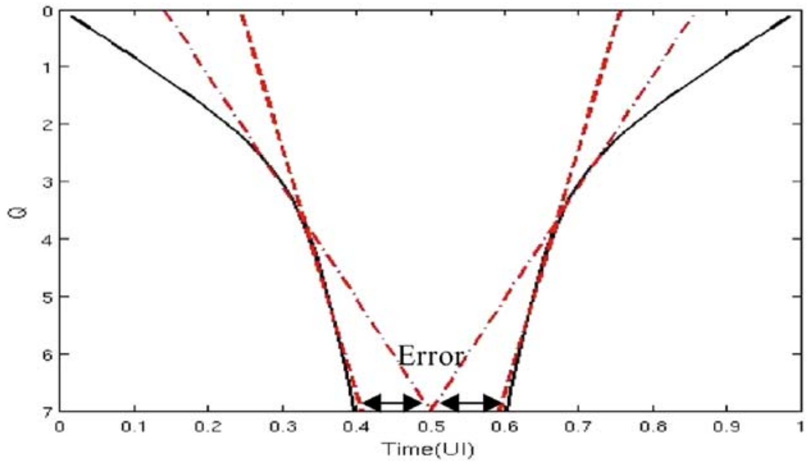
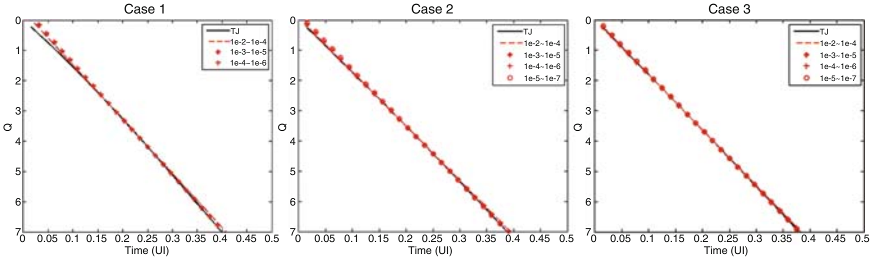


Fig. 6.3 The bathtub curve and TJ estimation

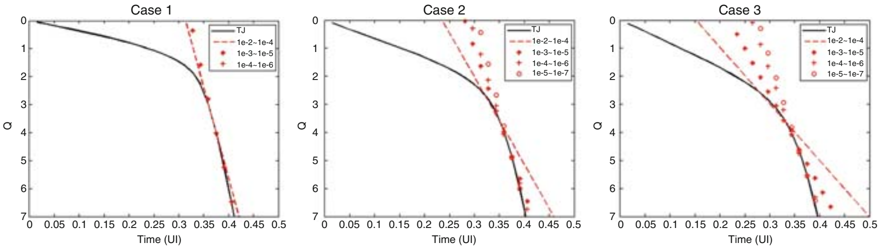
Figure 6.3 shows the bathtub curve in the Q-domain. The solid line gives the distribution from an actual measurement. The dashed lines represent the estimation results using the dual-Dirac model from two different fitting regions. As the figure indicates, the estimation accuracy is extremely sensitive to the choice of fitting region. Choosing a lower BER region for fitting results in a more accurate estimation. This is because we only use the tail of the distribution, which closely follows the true Gaussian distribution for fitting. However, measuring error rates at low BER levels demands longer testing time and, in turn, greater testing cost.

We conducted MATLAB simulations to find out the appropriate fitting region for an accurate estimation of TJ at a 10^{-12} BER level (i.e. Q $=$ 7). However, generating a bathtub curve down to the 10^{-12} BER level is almost impossible in the simulation environment due to the excessive simulation time. Instead, we derived closed form equations for the BER distributions. We generated the distributions for three different DJ cases: (1) Case 1: DJ has only an ISI component, and thus follows a uniform distribution. (2) Case 2: DJ has only a crosstalk component, and thus follows a triangular distribution. (3) Case 3: DJ has both ISI and crosstalk components, and thus its distribution results from the convolution of uniform and triangular distributions. Then, we applied the first-order line fitting by choosing different BER regions to estimate the TJ, and compared the results with the actual TJ at a 10^{-12} BER level. The step size for generating the bathtub curve was set to 1/64 UI (the PI in a modern transceiver can generate 64 steps within a UI [9, 58]). We chose four different BER ranges as the fitting regions $-10^{-2} - 10^{-4}$, $10^{-3} - 10^{-5}$, $10^{-4} - 10^{-6}$, and $10^{-5} - 10^{-7}$. We conducted experiments for the three DJ cases. Figure 6.4 shows the results when RJ is dominant (i.e., RJ has 0.05 UI rms, DJ has 0.07 UI peak-to-peak). Figure 6.5 shows the results when DJ is dominant (i.e., RJ has 0.01 UI rms, DJ has 0.35 UI peak-to-peak). The tables list the differences between the estimated TJ and the actual TJ.



Distribution type	$10^{-2} \sim 10^{-4}$	$10^{-3} \sim 10^{-5}$	$10^{-4} \sim 10^{-6}$	$10^{-5} \sim 10^{-7}$
Case 1 (uniform)	2.25 %	1.17 %	0.76 %	-
Case 2 (triangular)	1.62 %	0.99 %	0.61 %	0.4 %
Case 3 (uniform+triangular)	1.35 %	1.17 %	0.91 %	0.78 %

Fig. 6.4 TJ estimation results for RJ dominant case



Distribution type	$10^{-2} \sim 10^{-4}$	$10^{-3} \sim 10^{-5}$	$10^{-4} \sim 10^{-6}$	$10^{-5} \sim 10^{-7}$
Case 1 (uniform)	2.3 %	0.53 %	0.53 %	-
Case 2 (triangular)	13.94 %	3.38 %	1.88 %	0.37 %
Case 3 (uniform+triangular)	25.91 %	9.17 %	2.15 %	1.15 %

Fig. 6.5 TJ estimation results for DJ dominant case

When RJ is dominant, the estimation error for different fitting regions does not vary too much. However, when DJ is dominant, the amount of error is very sensitive to the fitting region. In addition, the amount of error in DJ Cases 2 and 3 is greater than that of Case 1. Thus, for Case 1, measuring the BER down to the 10^{-5} level is sufficient for estimating the TJ at the 10^{-12} level (with less than 1% error). For Cases 2 and 3, measuring the BER to at least a 10^{-7} level is required for sufficiently accurate estimation.

Since the triangular distribution can be represented by the convolution of two uniform distributions, the jitter distribution for Case 3 can also be interpreted as the convolution of three uniform distributions. As a result, the jitter distribution for Case 3 is more similar to the Gaussian distribution, according to the central limit theorem. As the DJ characteristics become closer to those of the Gaussian distribution, it becomes more difficult to separate the DJ and the RJ. Therefore, if there are more noise sources in the system, the estimation error will further increase and measuring BER at levels lower than 10^{-7} would become necessary.

Besides the accuracy limitation, the estimation technique has to discard the measurement data at higher BER levels in order to fit the straight line for the RJ. After discarding these data, if the number of BER data points within the chosen fitting range is not sufficient, the estimation accuracy might be further degraded. In order to guarantee enough samples within the fitting region, this technique would require a PI with a finer resolution.

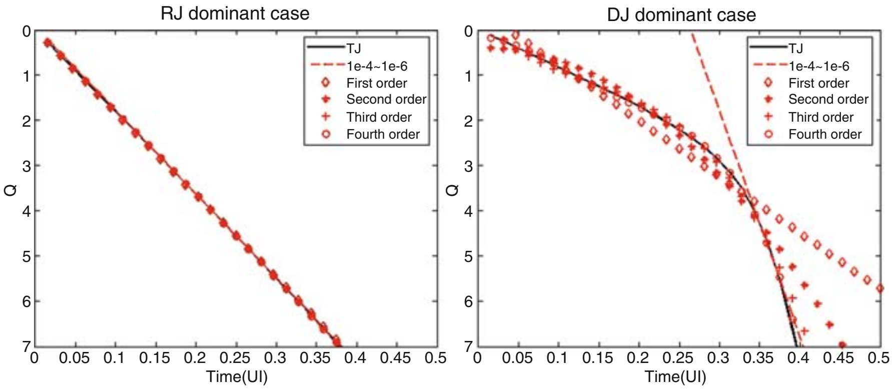
6.2.2 High-Order Polynomial Fitting

In order to address the limitations of the dual-Dirac-model-based technique, we propose to use high-order polynomial fitting to estimate TJ. No high BER samples would be discarded and all BER data down to the 10^{-6} BER region are used for fitting. We believe 10^{-6} BER is a good choice as the lower bound for fitting because this BER level can be measured very quickly in most multi-gigabit per second links. (e.g., at a 5 Gbps data rate, to measure BER around the 10^{-6} level capturing 10^{8} samples takes only 0.02 s). We applied the first, second, third, and fourth order polynomial fittings to both RJ and DJ dominant cases, assuming DJ has both ISI and crosstalk. The RJ and DJ values for each case are identical to those of the previous case. The estimation errors are compared with the dual-Dirac-model-based line fitting results for which BERs in the range of $10^{-4} - 10^{-6}$ are used for fitting. Since it uses the BER information within the window of $10^{-4} - 10^{-6}$, we call it a window-based first-order fitting. This is in contrast to the first order fitting which uses all BER information down to 10^{-6}.

The simulation results are shown in Fig. 6.6. As the polynomial order increases, the error decreases. The fourth order fitting gives less than 1% estimation error for all cases.

6.2.3 Accuracy Versus Number of Samples for Fitting

Our early analysis concluded that the TJ estimation using fourth-order polynomial fitting results in very accurate results. In this section, we explore the number of measurement points within the fitting region required to maintain the estimation accuracy. In the earlier discussion, we assumed that the PI can generate 64 steps within a UI. Thus, in the DJ dominant case, there are 24 points from the transition



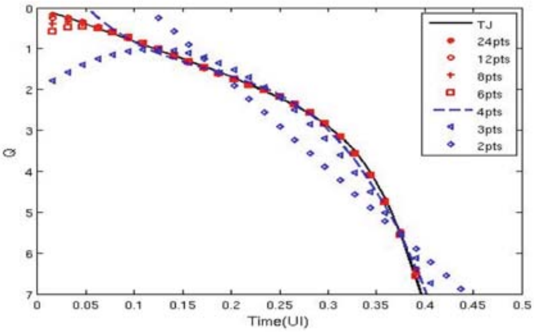
	Window based	1st order	2nd order	3rd order	4th order
RJ dominant	0.91 %	1.53 %	0.67 %	0.39 %	0.54 %
DJ dominant	2.15 %	> 30 %	14.68 %	4.17 %	0.83 %

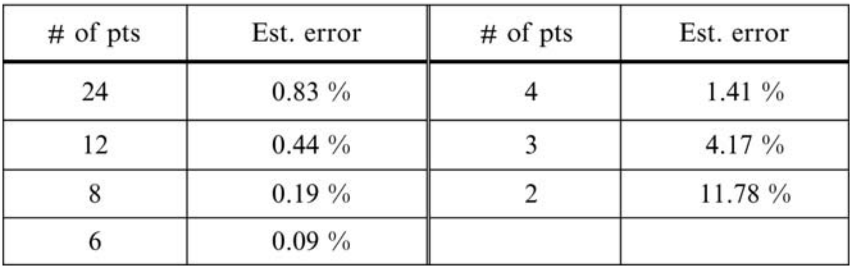
Fig. 6.6 TJ estimation results based on high-order polynomial fitting

edge to the 10^{-6} BER region. We reduced the number of BER measurements for fitting to 12, 8, 6, 4, 3, and 2 by evenly skipping some PI steps. For each case, we estimated the TJ using fourth-order fitting and compared the estimation error with the original case which used all 24 points. Figure 6.7 shows the simulation results. The results indicate that four points would be sufficient for estimating TJ in order to achieve an error rate of less than 1.5%.

6.3 Summary

As BER requirements become more and more stringent, the industry demands new testing methods, that are both efficient and accurate, to guarantee the product meets the BER specification at a low test cost. In this chapter, we first explored the applicability of the dual-Dirac-model-based technique for estimating TJ to the widely used loopback-based I/O margining test. Our conclusion is that the accuracy of the former technique is highly sensitive to the choice of the fitting region. Also, as DJ behaves more and more like a Gaussian distribution, this technique requires data to be measured at lower BER levels to ensure reasonable accuracy. This requirement implies longer testing time and, in turn, greater test cost. We propose the use of high order polynomial fitting to improve both the efficiency and accuracy of the estimation. This fitting strategy alleviates the need to measure the BER at a very low level,





# of pts	Est. error	# of pts	Est. error
24	0.83 %	4	1.41 %
12	0.44 %	3	4.17 %
8	0.19 %	2	11.78 %
6	0.09 %		

Fig. 6.7 TJ estimation results depending on number of samples

and thus reduces the testing time. It also enables the use of all BER data points for fitting to improve the estimation accuracy. The experimental results show that, for a system affected by both RJ and DJ, fourth-order fitting using four BER measurements, which are evenly spread from the transition region to the 10^{-6} BER region, can accurately estimate TJ at the 10^{-12} BER level.

Chapter 7
A Two-Tone Test Method for Continuous-Time Adaptive Equalizers

With the increasing demand of higher bandwidth, the data rate of I/Os is approaching the tens of gigahertz range. While the continuing advancement of process technology enables an I/O chip to run at such frequencies, the bandwidth of the communication channels, including cables and legacy backplanes, has become the limiting factor.

The bandwidth limitation of the channel causes ISI. Various equalization techniques, which multiply the inverse response of the channel to flatten out the overall frequency response, have been developed to compensate for this channel effect. In addition, the channel characteristics may not be known in advance and might be time-variant [59]. To cope with such problems, several adaptation algorithms have also been developed to adjust the overall system response depending on the channel conditions.

The equalizer can be implemented either in the transmitter or in the receiver. The implementation of the transmitter equalizer is relatively easier than that of the receiver equalizer because the required Finite Impulse Response (FIR) filter deals with the digital data at the transmitter side, rather than the received analog data at the receiver side [60,61]. However, since channel information is not easily available at the transmitter, it is difficult to apply the adaptive technique at the transmitter.

The approaches of equalization at the receiver can be divided into two categories: discrete-time equalization and continuous-time equalization. A discrete-time equalizer, which is based on the FIR filter, can take advantage of various digital adaptive algorithms [41,42,62,63]. However, since equalization is based on the samples captured by the receiver's recovered clock, there exists a cross-dependence between the equalizer and the clock recovery circuit. As the data rate increases, the power consumption would increase dramatically due to the large number of taps implemented in this type of equalizer [64]. On the other hand, a continuous-time equalizer does not require a sampling clock and thus the equalizer would work independent of the clock recovery circuit. Continuous-time equalizers have been investigated for low power and high speed applications, and promising performance has been reported [64–69].

While equalizer design has been studied for a long time and a number of novel architectures have recently been developed, high-quality and cost-effective production test methods for these equalizers are not yet well developed. The most popular

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_7,

means of testing equalizers is to measure the eye-diagram using either an external scope or on-chip measurement circuitry [70]. These methods test the equalizer by simply comparing the eye-openings before and after the equalizer. However, the eye-opening results measured in the test environment would not match those in the real system if the channels in the test board and those in the real applications are different. In addition, measuring the eye-diagram in the multi-gigahertz range requires either expensive equipment for external measurement or a significant amount of internal circuitry for on-chip measurement. In [9], a cost-effective test method for adaptive equalizers is proposed. However, this technique has a limitation: it could not properly test the adaptation loop in the equalizer because the low frequency gain of the equalizer is programmed before testing.

In this chapter, we propose a novel test method for a continuous-time adaptive equalizer. Our technique can cost-effectively test both the equalization filter and the adaptation loop. The advantages of the proposed technique are as follows: (1) the test stimuli can be easily generated and applied, which need not be stressed using different channels. (2) The test output is a DC signal and thus can be easily measured. (3) The extra on-chip circuitry needed for supporting the technique requires only very small area overhead and does not result in any performance degradation.

In the next section, we describe the basic operation of a continuous-time adaptive equalizer. Section 7.2 describes the proposed two-tone test method. Section 7.3 presents the experimental setup and results for the validation of the proposed technique. Section 7.4 concludes the paper and discusses future work.

7.1 Continuous-Time Adaptive Equalizer

The block diagram of a conventional continuous-time adaptive equalizer is shown in Fig. 7.1. The equalization filter either boosts the high-frequency components or attenuates the low-frequency components of the received input signal to compensate

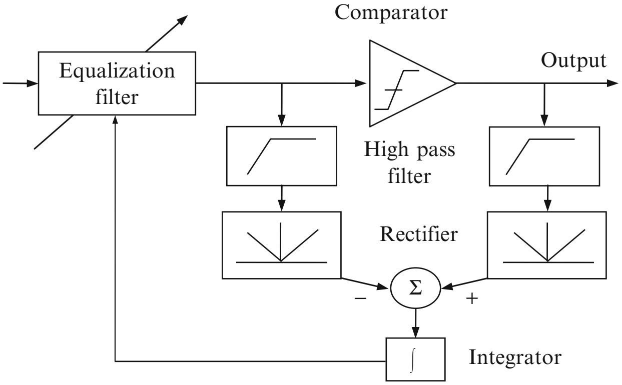


Fig. 7.1 Block diagram of a conventional continuous-time adaptive equalizer

for high-frequency loss due to the channel. The adaptive servo loop, which adjusts the compensation gain of the equalization filter, determines the control voltage by comparing the comparator's input and the output signals. In practice, it is difficult to design a comparator that can generate a clean waveform for comparisons at very high frequencies. Several new approaches for adaptation [67–69] have been proposed to address this problem. These new methods use the power spectrum derived from the output signal of the equalization filter for the adaptation, as shown in Fig. 7.2. Since the power spectrum of a random signal can be described by a sinc^2 function, the high-frequency loss can be detected by comparing the power densities of two different frequency ranges. Three different methods have been proposed to compare the power spectrum of the random signal. Two band-pass filters are used in [67] to compare the power of two specific frequencies. In [68], one low-pass filter and one high-pass filter are used to compare the power between the low-frequency and high-frequency portions of the signal. In [69], only one low-pass filter is used and the entire signal's power is compared to that of the low-frequency portion of the signal. Figure 7.3 illustrates these three different power spectrum comparison architectures. Our proposed test method is applicable to the modified continuous-time adaptive equalizer (i.e. the type shown in Fig. 7.2) under any of these three architectures.

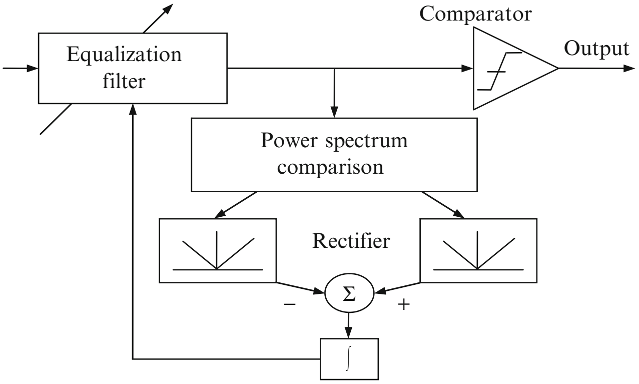


Fig. 7.2 Block diagram of a modified continuous-time adaptive equalizer

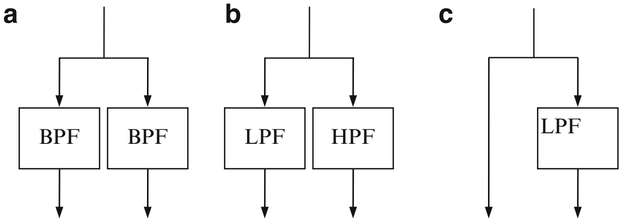


Fig. 7.3 Three different architectures for power spectrum comparison

7.2 Proposed Two-Tone Test Method

7.2.1 Description of the Test Method

The proposed test technique directly uses a two sinusoidal-tone signal as test stimulus instead of a data pattern stressed through the channel as is commonly used for validating and characterizing the equalizers. The frequency for one of the two sinusoidal tones, denoted as f_L, falls within the stop-band region; the frequency of the other tone, denoted as f_H, falls within the pass-band region. Figure 7.4 illustrates the frequency response of an adaptive equalizer that can compensate for the channel loss up to α_0 dB, and the frequency bands of two sinusoidal tones. In order to mimic the channel response, this technique repeatedly applies the two-tone signal and gradually varies the magnitude ratio of f_H and f_L. Specifically, we gradually increase the magnitude of f_L while fixing the magnitude of f_H. Such test stimuli mimic different relative losses of the high-frequency components caused by the channel. While this could also be accomplished by gradually reducing the magnitude of f_H and fixing the magnitude of f_L, it is much easier in practical implementations to adjust the magnitude of the low-frequency component.

With the test stimuli, we measure the rms value at the output of the equalizer. The adaptive servo loop attempts to maintain the ratio of f_L to f_H to the expected level based on the sinc^2 function. Thus, as illustrated in Fig. 7.5, if the test stimulus's f_L magnitude is within the range of R1 and, therefore, the magnitude ratio of f_L to f_H is within the range of the equalizer's maximum compensation gain (i.e. $G_{EQ\_Max} = 10^{\alpha_0/20}$), the rms value of the equalizer output should be a constant. When the magnitude of f_L is increased to the point that the magnitude ratio of f_L to f_H exceeds the maximum compensation gain of the equalizer (as indicated in Fig. 7.5, where the magnitude of f_L is in the range of R2), the rms value at the equalizer output will start to increase. Figure 7.6 illustrates how the rms value at the

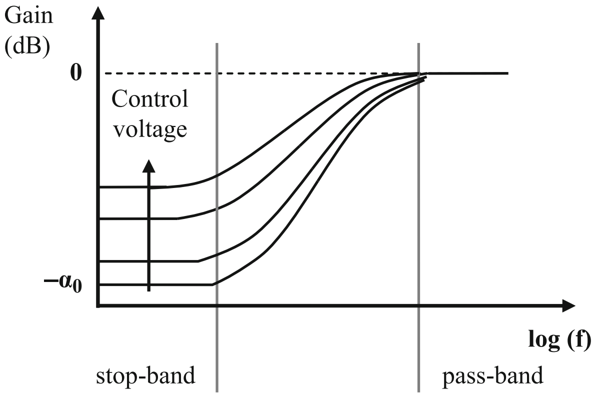


Fig. 7.4 Frequency response of an adaptive equalizer

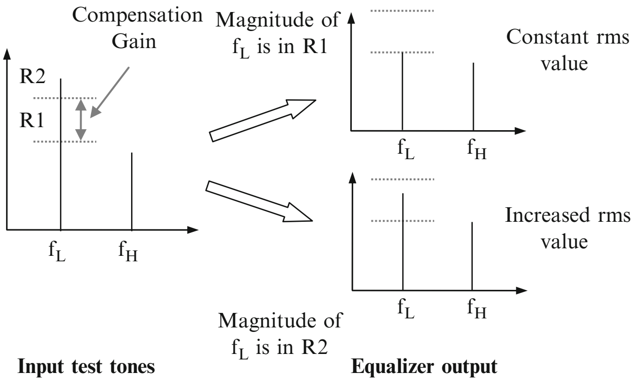
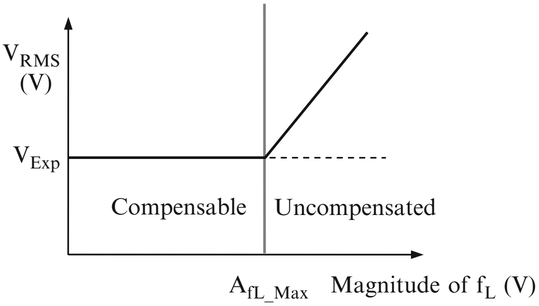


Fig. 7.5 Magnitude variations of two sinusoidal signals

Fig. 7.6 Output rms value vs f_L's magnitude



equalizer output would vary with respect to the magnitude of the f_L (with a fixed magnitude of f_H). Therefore, we can test the equalizer's maximum compensation gain, which is a key design specification of the equalizer, by identifying the magnitude of f_L when the rms value starts to deviate from the expected value, denoted as A_{fL_Max} in Fig. 7.6. The adaptive loop also can be tested by the constant rms values up to the A_{fL_Max}. Note that the conventional eye-diagram method may not easily test these specifications because: (1) designing the test board channel to exactly match the expected high-frequency loss is difficult. (2) A number of different channel lengths are required for testing the adaptive loop, and this may not be feasible for production testing.

7.2.2 Implementation of the Test Method

We applied the proposed test technique to a continuous-time adaptive equalizer recently presented in [69, 72] whose block diagram is shown in Fig. 7.7. It uses a passive equalization filter whose control voltage is adjusted by the servo loop, based on the power spectrum of the data (i.e. the type shown in Fig. 7.3c). For this type of

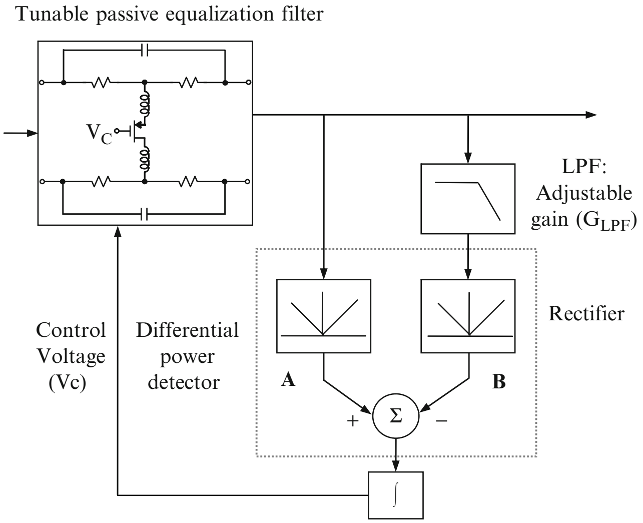


Fig. 7.7 Block diagram of the adaptive equalizer

equalizer, since the rectifiers already exist in the architecture, we only need to add an LPF to implement the rms detector.

There are two possible locations for measuring the rms value: either node A or node B (denoted in Fig. 7.7). The rms measurement at node A would be the combined rms value of f_L and f_H which is the same as the equalizer's output rms value. The rms measurement at node B would also result in a curve similar to that seen in Fig. 7.6 because the magnitude of f_H remains constant and only the magnitude of f_L varies.

We denote f_L's magnitude after the equalization filter as A_{fL_Out} and f_H's magnitude after the equalization filter as A_{fH}. The adaptive servo loop attempts to make the power of the two paths equal. That is:

$$A_{fL\_Out}^2 \times G_{LPF}^2 = A_{fL\_Out}^2 + A_{fH}^2$$

Therefore, the expected ratio of A_{fH} to A_{fL_Out}, denoted by K_{Exp}, becomes:

$$K_{Exp} = \frac{A_{fH}}{A_{fL\_Out}} = \sqrt{G_{LPF}^2 - 1} \qquad (7.1)$$

where G_{LPF} is the LPF voltage gain (see Fig. 7.7).

Note that the f_H's magnitude before the equalization filter is the same as A_{fH} because the gain of the equalization filter is close to 1 in the pass-band.

If we denote f_L's magnitude before the equalization filter as A_{fL_In}, we can then derive the expected rms value (V_{Exp}) as follows:

1. When A_{fL_In} is in R1 (as shown in Fig. 7.5)

The rms values at node A and node B would be:

$$V_{RMS(A)} = \sqrt{(A^2_{fL\_Out} + A^2_{fH})/2} = G_{LPF}A_{fH}/\sqrt{2G^2_{LPF} - 2}$$

$$V_{RMS(B)} = G_{LPF}A_{fL\_Out}/\sqrt{2} = G_{LPF}A_{fH}/\sqrt{2G^2_{LPF} - 2}$$

Note that we modify both of the equations with respect to A_{fH} using Eq. 7.1. Then, the rms values for both cases are the same as expected because A_{fH} is a constant. Therefore, the expected rms value would be

$$V_{Exp} = G_{LPF}A_{fH}/\sqrt{2G^2_{LPF} - 2} \tag{7.2}$$

2. When A_{fL_In} is in R2 (as shown in Fig. 7.5)
   The rms value at A and B would be:

$$V_{RMS(A)} = \sqrt{(A^2_{fL\_Out} + A^2_{fH})/2} = \frac{\sqrt{A^2_{fL\_In} + A^2_{fH}G^2_{EQ\_Mzx}}}{\sqrt{2}G_{EQ\_Max}}$$

$$V_{RMS(B)} = G_{LPF}A_{fL\_Out}/\sqrt{2} = \frac{G_{LPF}A_{fL\_In}}{\sqrt{2}G_{EQ\_Max}} \tag{7.3}$$

In this region, since the equalizer cannot fully attenuate the low-frequency component to the desired level, the rms value would be greater than the expected level and would thus increase as A_{fL_In} increases. However, the increasing rate would be different for these two locations. The increasing rate of the rms value measured at B would be greater because it is multiplied by G_{LPF}, as shown in Eq. 7.3.

The magnitude of f_L at which the rms value starts to deviate from the expected value can also be represented using Eq. 7.1:

$$A_{fL\_Max} = \frac{A_{fH}}{K_{Exp}} \cdot G_{EQ\_Max} \tag{7.4}$$

Therefore, by measuring the A_{fL_Max}, we can test both the equalization filter and the adaptive servo loop. Any defects and/or errors that cause variations to G_{EQ_Max} and K_{Exp} would be detected.

7.3 Experimental Results

To validate the proposed technique, we conducted both behavioral-level simulation using MATLAB and transistor-level simulation using Cadence Spectre. The design parameters of the equalizer used for the experiment are:

$G_{EQ\_Max} = 17\,\text{dB} = 7$
$G_{LPF} = 3.1$
$f_H = 5\,\text{GHz}$ and $A_{fH} = 80\,\text{mV}$
$f_L = 100\,\text{MHz}$ and $A_{fL\_In} = 80 - 340\,\text{mV}$

Based on these parameters, the V_{Exp} and A_{fL_Max} can be calculated using Eqs. 7.3 and 7.4:

$$V_{RMS} = 60\,\text{mV}$$
$$A_{fL\_Max} = 191\,\text{mV}$$

7.3.1 MATLAB Simulation Results

Two sinusoidal signals are injected into the behavioral model of the equalizer and the rms values are measured at A and B. For each location, the simulation is repeated by gradually increasing the magnitude of the low-frequency component (A_{fL_In}).

Figure 7.8 shows the simulation results for both cases. The rms values for both cases are constant up to 180 mV and start to increase around 200 mV. The expected

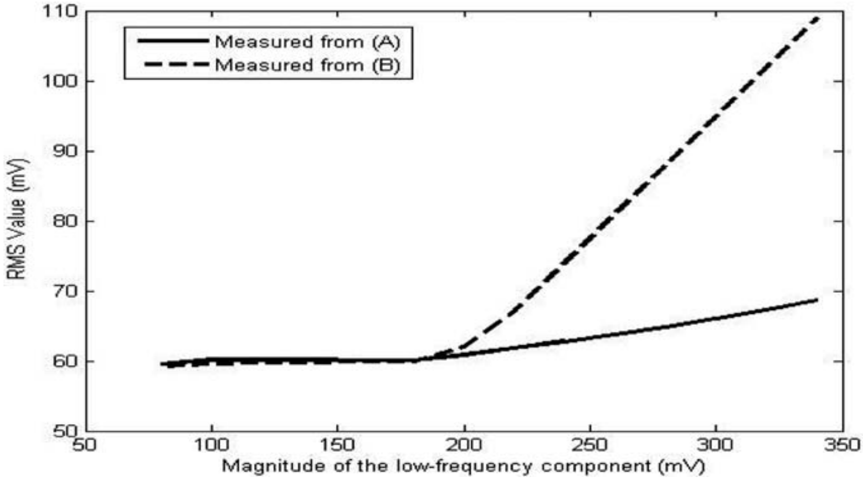


Fig. 7.8 MATLAB simulation results

A_{fL_Max} is about 190 mV. As the step size of A_{fL_In} in our experiment was 20 mV, the rms value increases only slightly at 200 mV but increases rapidly starting from 220 mV. This result indicates that the proposed method could indeed identify the device's A_{fL_Max} fairly accurately. However, the increasing rate for measurement at A is small the difference of the rms values between steps is only a few millivolts. This difference would be too small to be detected in real applications, particularly in the presence of various noise sources. For our test case, the expected ratio of A_{fH} to A_{fL_Out} is about 3. Therefore, the rms value of A_{fH}, which is constant, would be significantly larger than that of A_{fL_Out}. Thus, there would be a small variation to the total rms value measured at A. Based on this observation, we implemented the rms detector at B for transistor-level simulation. However, for designs with a small ratio of A_{fH} to A_{fL_Out}, the rms detector placed at A could be a viable solution, too.

We injected a couple of parametric faults into the equalizer for simulation and examined the rms value at B to check whether the faults are detectable using this technique. Specifically, we changed the maximum compensation gain of the equalizer to (1) 13 dB and (2) 10 dB from 17 dB. Note that defects/errors that cause the equalizer to malfunction would cause changes to its maximum compensation gains.

Based on Eq. 7.4, if the maximum compensation gain is reduced, the rms value would start deviating from the expected value at a lower magnitude. Figure 7.9 shows the simulation results of the three cases (fault-free circuit and faulty circuits (1) and (2)). The maximum compensation gain's variation can be easily identified by detecting the change to A_{fL_Max}.

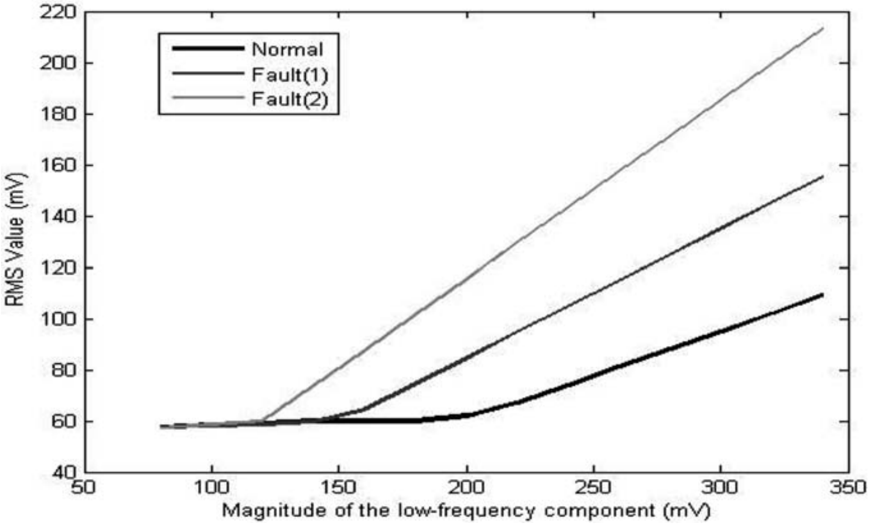


Fig. 7.9 Fault simulation results

7.3.2 Transistor-Level Simulation Results

The continuous-time adaptive equalizer shown in Fig. 7.7 has been designed in a 0.13-μm CMOS process. The design details of the tunable passive equalization filter was reported in [69]. The die photo and the measured filter response are shown in Fig. 7.10. The filter can compensate for a loss up to 17 dB at 5 GHz with a control voltage ranging from 0.1 to 0.6 V.

The rms detector is implemented inside the differential power detector in the servo loop (see Fig. 7.7). We reuse the rectifier in the servo loop and only add a simple RC low-pass filter at the output of the rectifier. To avoid degrading the

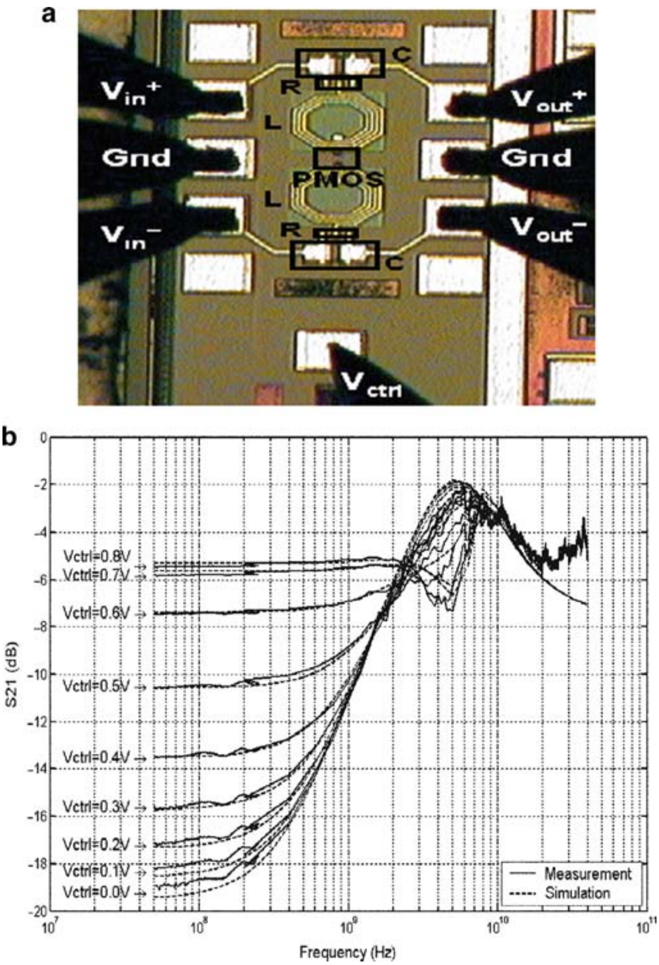


Fig. 7.10 Die photo and the filter response: (**a**) chip micrograph of the 0.13-μm filter prototype, (**b**) measurement vs simulation results of the filter's differential S21

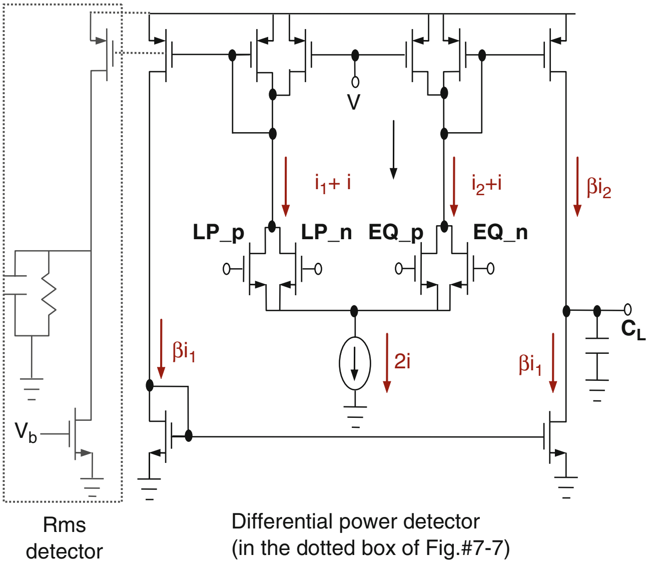


Fig. 7.11 Schematic of the differential power detector with rms detector

servo loop's performance, a current mirror is used to copy the output of the rectifier. Figure 7.11 shows the schematic of the differential power detector and the rms detector.

Figure 7.12a shows the rms values measured before the simple RC LPF. Figure 7.12b shows the DC output of the rms detector. For both figures, we compare the nominal case with two defective cases. For the defective cases, the gain of the servo loop's LPF (G_{LPF}) was changed to 3.75 (Fault A) and 4.5 (Fault B), respectively. This, in turn, also results in a different A_{fL_Max}. The rms values start to deviate at a lower A_{fL_In} for the defective cases as expected from Eq. 7.4. Therefore, any variation in the adaptive loop also can be easily identified by the change to A_{fL_Max}.

Compared to the MATLAB simulation results, the transistor-level simulation shows that the output rms values are not constant, even within the equalizer's compensation gain range. This is because the gain of the equalization filter at 5 GHz slightly varies depending on the control voltage, which in turn contributes to the variation in the expected rms value. However, since the difference in slope between the two regions is obvious, we could still clearly identify A_{fL_Max}.

7.4 Summary and Future Work

We proposed an efficient method for testing the continuous-time adaptive equalizer that uses a simple two-tone signal as a test stimulus and an rms detector for on-chip measurement. This technique can detect defects either in the equalization filter or

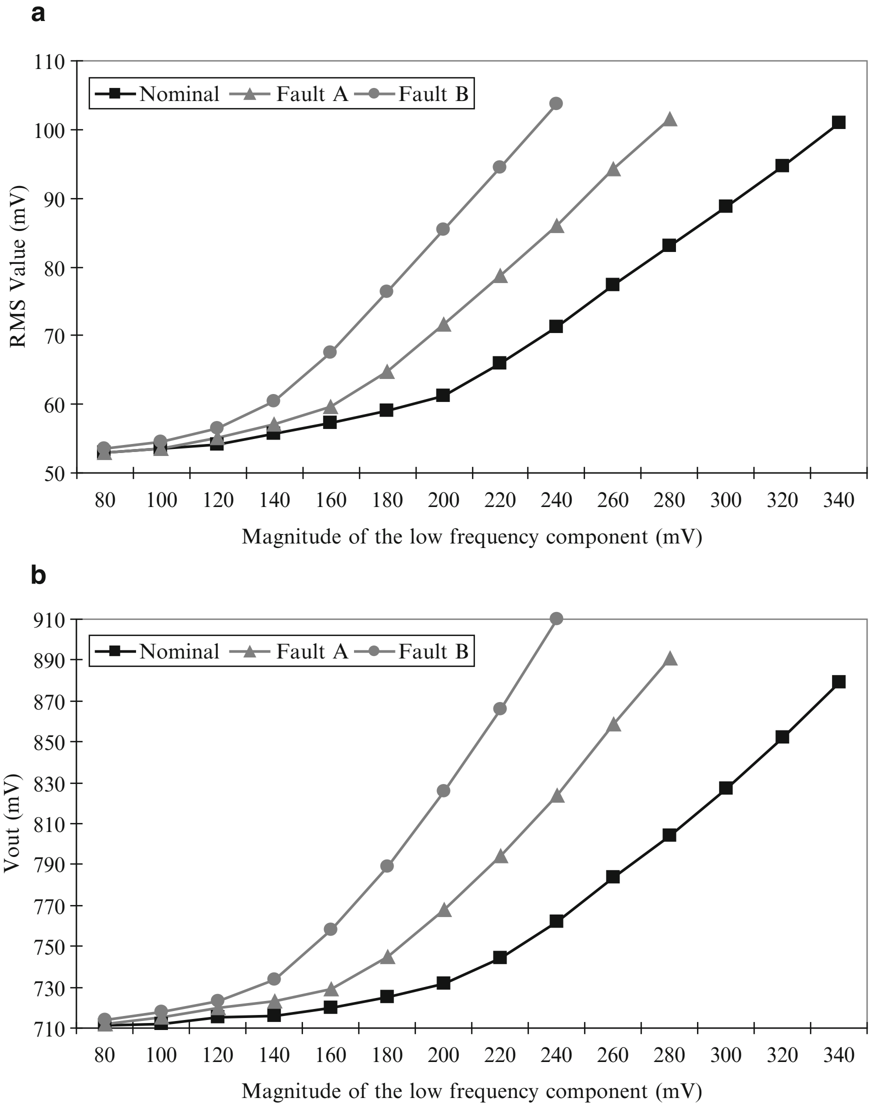


Fig. 7.12 Spectre simulation results: (**a**) measured rms value before the RC LPF, (**b**) DC output of the rms detector

in the adaptive servo loop that might not be easily detectable by the conventional eye-diagram method. We validated the idea by simulation using a recently proposed continuous-time adaptive equalizer. The behavioral and the transistor-level simulation results demonstrate the validity of our test technique.

In this chapter, we focused the discussion on testing a stand-alone continuous-time adaptive equalizer. This technique could be extended for the adaptive equalizers built in a high-speed transceiver. For such an extension, the high-frequency sinusoidal component would be replaced by a clock-like pattern generated by the transmitter in the loop-back test mode that has been the most popular means of testing high-speed transceivers [9, 71]. In addition, the low-frequency sinusoidal component would be added from external equipment. To further remove the need for external equipment in the loopback test, we are investigating a cost-effective, on-chip, low-frequency sinusoidal noise injection method to support this technique.

Chapter 8
Conclusions

This research has investigated efficient test methodologies for high-speed serial links. We have shown in Chapter 2 that the comparator-based undersampling technique can efficiently measure the jitter using either ATE or on-chip circuitry due to its simplicity and high bandwidth. The measurement results with the prototype board demonstrate the accuracy of the technique compared to the conventional histogram method.

To expedite the BER testing and correlate the measured jitter information to the BER, we use the BER estimation techniques proposed in Chapters 3 and 4. These techniques utilize the jitter spectral information and the characteristics of the CDR circuit for BER estimation. The high-speed serial links that incorporate the linear CDR circuit are first analyzed based on the linear PLL theory. Experimental results comparing the estimated BER and the BERT-measured BER on a 2.5 Gbps commercial CDR circuit demonstrate the high accuracy of the proposed technique. In Chapter 4, the BER estimation technique is extended to the serial links with a BB CDR circuit. Due to the heavily non-linear nature of the BB CDR circuit, the jitter transfer characteristics are analyzed depending on the jitter magnitude. Then, the error occurring mechanism of the non-linear loop is analyzed for the BER estimation.

Loopback-based margining test has been widely adopted by industry because it can test most I/O functionality without requiring expensive ATE. However, it cannot fully replace conventional jitter test due to the random jitter, the non-linear characteristics of a CDR circuit, jitter amplification, and the DCD in clock signal. In order to cope with the RJ underestimation problem in timing margining test, we propose the total jitter estimation technique in Chapter 6. The experimental results show that fourth-order fitting using only four BER measurements, which are evenly spread from the transition region to the 10^{-6} BER region, can accurately estimate the timing margin at the 10^{-12} level.

Adaptive equalizers have been widely used in most advanced transceivers to compensate for channel loss. In Chapter 7, we propose a two-tone test method for continuous-time adaptive equalizer, which can test both the equalizing filter and adaptation loop with only a very small area overhead and no performance degradation.

With an ever increasing demand for high bandwidth in communication systems, recent transceivers include more functionality in their designs which results in more

D. Hong and K.-T. Cheng, *Efficient Test Methodologies for High-Speed Serial Links*,
Lecture Notes in Electrical Engineering 51, DOI 10.1007/978-90-481-3443-4_8,

testing challenges. Some transceivers include a phase interpolator to generate the optimal sampling clock and to conduct the timing margining test [9,47,58,73,74], which might not be fully covered by the conventional jitter test or the margining test. Thus, developing DFT techniques for phase interpolator is a possible area for research in serial links testing. In addition, ADC/DACs have been utilized in high-speed transceivers either for equalization [75,76] or for compensating for process variations [74,77]. Developing efficient test methodologies for these circuitries also would be fruitful future research area.

Appendix A
Extracting Effective PJ and RJ Components from Jitter Histogram

This appendix describes how to extract effective sinusoidal and random components from the jitter histogram. The jitter histogram, which represents the jitter Probability Density Function (PDF), is commonly used for characterizing the jitter. In order to estimate the BER when both sinusoidal and random components are present and contributing to errors, the PDF of the sum of these components should be known. The PDF of the sum of two random variables is the convolution of the individual PDFs. However, there is no closed-form expression for the convolution of a sinusoidal PDF and a Gaussian PDF. Therefore, we can approximate the total jitter PDF using the double delta model. Figure A.1 shows the convolution of a sinusoidal distribution and a Gaussian distribution. Figure A.1a illustrates the case when the random term is dominant, and Fig. A.1b shows the case when the sinusoidal term is dominant.

When RJ is dominant, the distribution follows the Gaussian distribution. On the other hand, when the sinusoidal term is dominant, the jitter distribution has two peaks.

Given a jitter histogram, in order to separate the jitter components, we can assume the tail part of the given distribution is mostly determined by the random component. Then, fitting it with a Gaussian curve would result in the rms value of the random component, as illustrated in Fig. A.2. The distance between the means of these two fitted Gaussians, $2A_{\text{eff}}$, gives the effective peak-to-peak amplitude of the PJ, which can be treated as the distance between two Dirac delta functions [14]. Note that this value is different from the injected peak-to-peak amplitude of the PJ [27].

We ran MATLAB simulations to quantify the variations of A_{eff} and σ_{eff} as a function of the ratio of the amount of RJ to the amount of the PJ. First, we computed the convolution of a sinusoidal PDF and a Gaussian PDF by varying the rms value of the random component with a fixed amount of PJ. Using the tail fitting algorithm described in [14], we calculated the A_{eff} and σ_{eff}. Figure A.3 shows the simulation results as a function of the ratio of the rms value of RJ, σ_{RJ}, to the rms value of PJ, σ_{PJ}. Figure A.3(a) shows the RJ rms error normalized with respect to σ_{PJ}, which is defined as:

$$\sigma_{\text{RJ\_err}} = \sqrt{{\sigma_{\text{eff}}}^2 - {\sigma_{\text{RJ}}}^2}$$

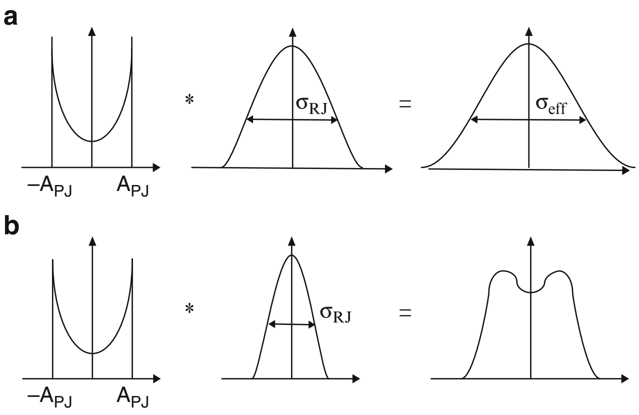


Fig. A.1 The convolution of two PDFs: (**a**) when random distribution is dominant, (**b**) when sinusoidal distribution is dominant

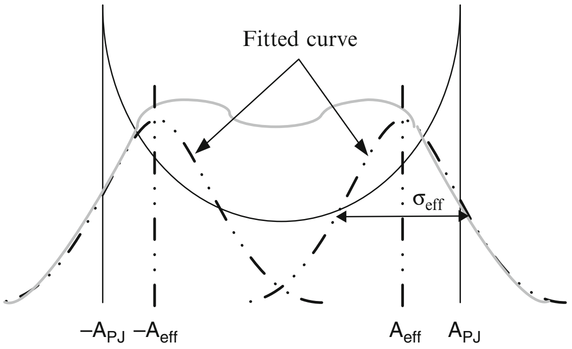


Fig. A.2 Tail fitting to the total jitter distribution

Figure A.3(b) shows the ratio of A_{eff} to A_{PJ} with respect to the ratio of σ_{RJ}/σ_{PJ}. Based on the Fig. A.3, the characteristics of σ_{eff} and A_{eff} can divide into two cases depending on the σ_{RJ}/σ_{PJ}.

1. When $\sigma_{RJ}/\sigma_{PJ} > 1$ (the case illustrated in Fig. A.1a)

In this case, the ratio of the RJ rms error to the rms value of PJ approaches one, which means the effective rms value is increased by the amount of the rms value of the PJ. In addition, A_{eff} is almost zero. Thus, the total jitter can be treated as pure random jitter (i.e. $A_{eff} = 0$), which has the following rms value:

$$\sigma_{eff} = \sqrt{{\sigma_{RJ}}^2 + {\sigma_{PJ}}^2}. \tag{A.1}$$

2. When $\sigma_{RJ}/\sigma_{PJ} < 1$ (the case illustrated in Fig. A.1(b))

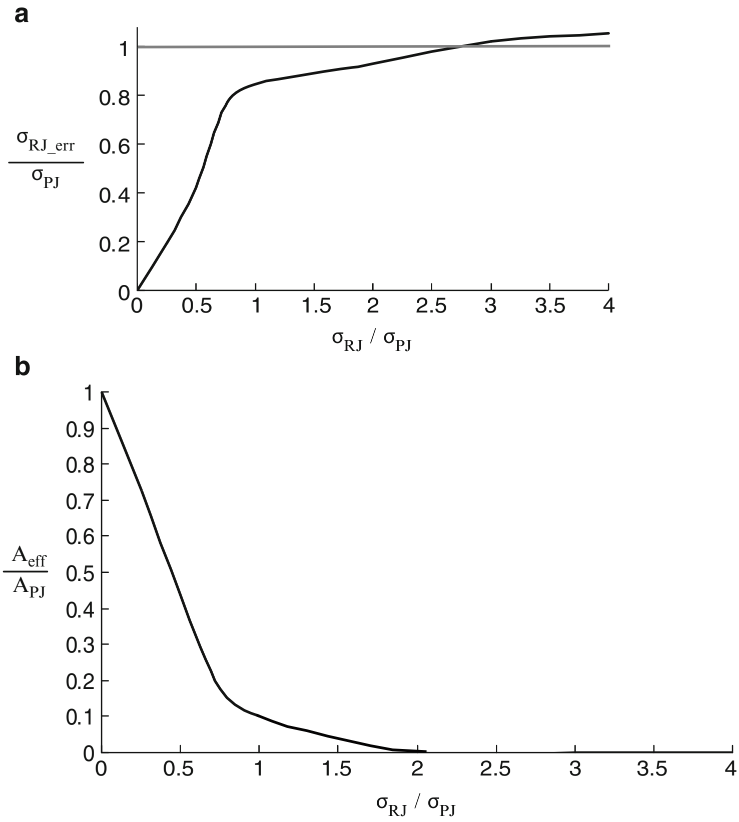


Fig. A.3 Simulation results of tail fitting: (**a**) RJ rms error, (**b**) effective PJ amplitude

Both of $\sigma_{RJ\_err}/\sigma_{PJ}$ and A_{eff}/A_{PJ} have an almost linear relationship with σ_{RJ}/σ_{PJ}. Thus, based on the two curves in Fig. A.3, we can easily derive the following equations to calculate A_{eff} and σ_{eff}:

$$A_{eff} = \left(1 - 0.9 \cdot \frac{\sigma_{RJ}}{\sigma_{PJ}}\right) \cdot A_{PJ}. \tag{A.2}$$

$$\sigma_{eff} = \sqrt{{\sigma_{RJ}}^2 + (0.84 \cdot \sigma_{RJ})^2}. \tag{A.3}$$

References

1. Partin J, Li M (2002) Comparison and correlation of signal integrity measurement techniques. DesignCon
2. Ou N et al. (July–Aug. 2004) Jitter models for the design and test of Gbps-speed serial interconnects, IEEE Des Test Comp 21(4):302–313
3. Tektronix (2003) A Guide to Understanding and Characterizing Timing Jitter. Available at www.tektronix.com/jitter
4. National Committee for Information Technology Standardization (NCITS) (2003) T11.2/ Project 1316-DT/ Rev 10.0, Fibre Channel-Methodologies for Jitter and Signal Quality Specification-MJSQ, 10 March 2003
5. Sunter S, Roy A (Sep 1999) BIST for Phase-Locked Loops in Digital Applications. In: Proceedings of International Test Conference, pp 532–540
6. Sunter S, Roy A (July–Aug 2004) On-chip digital jitter measurement, from megahertz to gigahertz. IEEE Des Test Comp 21(4):314–321
7. Chan AH, Roberts GW (Jan 2004) A jitter characterization system using component-invariant vernier delay line. IEEE Trans Very Large Scale Integr (VLSI) Sys 12(1):79–95
8. Mak TM et al. (July–Aug 2004) Testing Gbps interfaces without a gigahertz tester. IEEE Des Test Comp 21(4):278–286
9. Robertson I et al. (Nov 2005) Testing high-speed, large scale implementation of SerDes I/Os on chips used in throughput computing systems. In: Proceedings of International Test Conference, pp 1–8
10. Laquai B et al. (Oct 2001) Testing gigabit multilane serdes interfaces with passive jitter injection filters. In: Proceedings of International Test Conference, pp 297–304
11. Reiss A, Saksena G (Feb 17, 2004) Capturing and evaluating high speed data streams, US Patent 6,694,462
12. Lyons T, Sullivan S (May 2001) Measuring jitter on high speed serial data streams, Teradyne Users Group (TUG) presentation
13. Dalal W, Rosenthal D (Oct 1998) Measuring jitter of high speed data channels using undersampling techniques. In: Proceedings of International Test Conference, pp 814–818
14. Li M et al. (Oct 1999) A new method for jitter decomposition through its distribution tail fitting. In: Proceedings of International Test Conference, pp 788–794
15. Cai Y et al. (Oct 2002) Jitter testing for multi-gigabit backplane SerDes – techniques to decompose and combine various types of jitter. In: Proceedings of International Test Conference, pp 700–708
16. Yamaguchi T et al. (June 2003) Extraction of instantaneous and RMS sinusoidal jitter using an analytic signal method. IEEE Trans Circuit Syst 50(6):288–298
17. Cai Y et al. (Oct 2002) Digital serial communication device testing and its implication on automatic test equipment architecture. In: Proceedings of International Test Conference, pp 600–609
18. PCI-SIG (2003) PCI Express Base Specification Revision 1.0a, 15 April 2003
19. IEEE Std. 802.3ae Part3, Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Specifications, IEEE Computer Society, 2002

20. Li M, Wilstrup J (Oct 2003) Paradigm shift for jitter and noise in design and test >1 gb/s communication systems. In: Proceedings of International Conference on Computer Design, pp 467–472
21. Ong CK et al. (Jan 2004) Jitter Spectral Extraction for Multi-gigahertz Signal. In: Proceedings of Asia and South Pacific Design Automation Conference pp 298–303
22. Cai Y et al. (Jan–Feb 2002) Jitter testing for gigabit serial communication transceivers, IEEE Des Test Comp. 19:66–74
23. Razavi B (2002) Design of integrated circuits for optical communications, McGraw-Hill, New York
24. Lee K. et al. (Feb 1998) A Jitter-Tolerant 4.5 Gb/s CMOS interconnect for digital display. In: Proceedings of International Solid-State Circuits Conference, pp 310–311
25. Chang K et al. (Feb 2005) Clocking and circuit design for a parallel i/o on a first-generation CELL processor. In: Proceedings of International Solid-State Circuits Conference, pp 526–527
26. Leon-Garcia A (1994) Probability and random processes for electrical engineering. Addison Wesley, Reading, MA
27. Agilent Technologies, Jitter Analysis: The dual-Dirac Model, RJ/DJ, and Q-Scale, White Paper. Available at www.agilent.com
28. Jiang H et al. (Aug 2002) Optimal loop parameter design of charge pump PLLs for jitter transfer characteristic optimization. In: Proceedings of Midwest Symposium on Circuits and Systems, vol 1, pp 344–347
29. Maxim Integrated Products (2004) Pattern Creator/Converter software User Manual, Application Note: HFAN-9.5.0, Available at www.maxim-ic.com
30. Oppenheim AV, Willsky AS (1983) Signals and systems. Prentice Hall, Englewood Cliffs, NJ
31. Hajimiri A (Feb 2001) Noise in phase-locked loops. In: Proceedings of Southwest Symposium on Mixed-Signal Design, pp 1–6
32. Lee D (Nov 2002) Analysis of jitter in phase-locked loops. IEEE Trans Circuit Syst II 49(11):704–711
33. Moon U et al. (May 2002) Spectral analysis of time-domain phase jitter measurements. IEEE Trans Circuit Syst II 49(5):321–327
34. McNeill J (June 1997) Jitter in ring oscillators. IEEE J Solid-State Circuit 32(6):870–979
35. James NK (Aug 2000) Cycle-domain simulation for phase-locked loops. In: Proceedings of Southwest Symposium on Mixed-Signal Design. pp 77–82
36. Muller M et al. (Feb 2005) Total jitter measurement at low probability levels. Using Optimized BERT Scan Method. Designcon
37. Maxim Integrated Products (2007), Low-Power, Compact 2.5 Gbps/2.7 Gbps Clock-Recovery and Data-Retiming IC. Available at www.maxim-ic.com
38. Tektronix (2006) TDS JIT3-Jitter Analysis and Timing Software. Available at www.tek.com
39. SyntheSys Research Inc. (2006) BERTScope and BERTScopeS Signal Integrity Analyzers. Available at www.bertscope.com
40. Stojanovic V, Horowitz M (Sept 2003) Modeling and Analysis of High-Speed Links. IEEE proceedings of Custom Integrated Circuits Conference. pp 589–594
41. Jaussi JE et al. (Jan 2005) 8-Gb/s source-synchronous I/O link with adaptive receiver equalization, offset cancellation, and clock de-skew. IEEE J Solid State Circuit 40(1):80–88
42. Stofanovic V et al. (April 2005) Autonomous dual mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery. IEEE J Solid State Circuit 40(4):1012–1026
43. Payne R et al. (Dec 2005) A 6.25-Gb/s binary transceiver in 0.13-um CMOS for serial data transmission across high loss legacy backplane channels. IEEE J Solid State Circuit 40(12):2646–2657
44. Lee J et al. (Sept 2004) Analysis and modeling of bang–bang clock and data recovery circuits. IEEE J Solid State Circuit 39(9):1571–1580
45. Walker RC (2003) Designing bang–bang plls for clock and data recovery in serial data transmission systems. In: Razavi B (ed) Phase-locking in high-performance systems, IEEE Press, New York, pp 34–45
46. Choi Y et al. (Nov 2003) Jitter transfer analysis of tracked oversampling techniques for multigigabit clock and data recovery. IEEE Trans Circuit Syst II 50(11):775–783

47. Provost B et al. (Oct 2004) PAC IO loopback design for high speed /spl mu/processor IO test. In: Proceedings of International Test Conference, pp 23–30
48. Sunter S, Roy A (Oct 2005) Structural tests for jitter tolerance in SerDes receivers. In: Proceedings of International Test Conference, pp 1–10
49. Nejedlo JJ (Oct 2003) IBIST^{TM} (Interconnect Built-In Self-Test) architecture and methodology for PCI Express: Intel's next-generation test and validation methodology for performance IO. In: Proceedings of International Test Conference, pp 784–793
50. Kim J (2002) Design of CMOS adaptive-supply serial links, Ph.D. thesis, Stanford University, California
51. Shimanachi M (Oct 2001) An approach to consistent jitter modeling for various jitter aspects and measurement methods. In: Proceedings of International Test Conference, pp 845–857
52. Balamurugan G, Shanbhag N (Aug 2003) Modeling and mitigation of jitter in high-speed source-synchronous inter-chip communication systems. In: Proceedings of Asilomar Conference on Signals, Systems and Computers, pp 1681–1687
53. Sidiropoulos S, Horowitz MA, semidigital A (Nov 1997) Dual delay-locked loop. IEEE J Solid State Circuit 32(11):1683–1692
54. Lee TH, Bulzacchelli JF (Dec 1992) A 155-MHz clock recovery delay- and phase-locked loop. IEEE J Solid State Circuit 27(12):1736–1746
55. Buckwalter JF, Hajimiri A (Mar 2006) Analysis and equalization of data-dependent jitter. IEEE J Solid State Circuit 41(3):607–620
56. Buckwalter JF, Hajimiri A (Mar 2006) Cancellation of Crosstalk_Induced jitter. IEEE J Solid State Circuit 41(3):621–632
57. Kuo A et al. (Oct 2005) Crosstalk bounded uncorrelated jitter (BUJ) for high-speed interconnects. IEEE Trans Instrum Measure 54(5):1800–1810
58. Casper B et al. (Feb 2006) A 20 Gb/s forwarded clock transceiver in 90 nm CMOS. In: Proceedings of International Solid-State Circuits Conference, pp 263–272
59. Zerbe J (2005) Comparison of adaptive and non-adaptive equalization methods in high-performance backplanes DesignCon
60. Liu J, Lin X (2004) Equalization in high-speed communication systems. IEEE Circuits Systems Magazine, Second quarter, pp 4–17
61. Dally WJ, Poulton J (Jan–Feb 1997) Transmitter equalization for 4-Gbps signaling. IEEE Microelec 17:48–56
62. Beukema T et al. (Dec 2005) A 6.4 Gb/s CMOS SerDes Core with feed-forward and decision-feedback equalization. IEEE J Solid State Circuit 40(12):2633–2645
63. Kim J et al. (Feb 2005) A four-channel 3.125-Gb/s/ch CMOS serial-link transceiver with a mixed-mode adaptive equalizer. IEEE J Solid State Circuit 40(2):462–471
64. Sun R et al. (Sept 2005) A low-power, 20-Gb/s continuous-time adaptive passive equalizer. In: Proceedings of International Symposium on Circuits and Systems, pp 920–923
65. Choi J et al. (Mar 2004) A 0.18-um CMOS 3.5-Gb/s continuous-time adaptive cable equalizer using enhanced low-frequency gain control method. IEEE J Solid State Circuit 39(3):419–425
66. Ahang G et al. (Feb 2004) A BiCMOS 10 Gb/s Adaptive cable equalizer. In: Proceedings of International Solid-State Circuits Conference, pp 92–93
67. Maxim Integrated Products (2003) Designing a simple, wide-band and low-power equalizer for FR4 copper link, DesignCon
68. Lee J (Feb 2006) A 20-Gb/s adaptive equalizer in 0.13 um CMOS technology. In: Proceedings of International Solid-State Circuits Conference, pp 92–93
69. Sun R et al. (June 2006) A tunable passive filter for low-power high-speed equalizers. In: Proceedings of VLSI Circuit Symposium, pp 198–201
70. Jaussi JE et al. (Dec 2003) An 8 Gb/s simultaneous bidirectional link with on-die waveform capture. IEEE J Solid State Circuit 38(12):2111–2120
71. Tripp M et al. (Oct 2003) Elimination of traditional functional testing of interface timings at Intel. In: Proceedings of International Test Conference, pp 1014–1022
72. Sun R (Dec 2005) A low-power, 20-Gb/s continuous-time adaptive passive equalizer, MS thesis, Carnegie Mellon University

73. Kromer C et al. (Dec 2006) A 25-Gb/s CDR in 90-nm CMOS for high-density interconnects. IEEE J Solid State Circuit 41(12):2921–2929
74. Hidaka Y et al. (Feb 2007) A 4-Channel 3.1/10.3 Gb/s transceiver macro with a pattern-tolerant adaptive equalizer. In: Proceedings of Solid-State Circuits Conference, pp 442–445
75. Momtaz A et al. (Apr 2007) A fully integrated 10-Gb/s receiver with adaptive optical dispersion equalizer in 0.13-um CMOS. IEEE J Solid State Circuit 42(4):872–880
76. Harwood M et al. (Feb 2006) A 12.5-Gb/s SerDes in 65 nm CMOS using a baud-rate adc with digital receiver equalization and clock recovery. In: Proceedings of Solid-State Circuits Conference, pp 436–438
77. Palmer R et al. (Feb 2007) A 14 mW 6.25 Gb/s transceiver in 90 nm CMOS for serial chip-to-chip communications. In: Proceedings of Solid-State Circuits Conference, pp 440–442