2007-03-08

Circuit and Modeling Solutions for High-Speed Chip-to-Chip Communication

Timothy Mowry Hollis
Brigham Young University - Provo

CIRCUIT AND MODELING SOLUTIONS FOR HIGH-SPEED CHIP-TO-CHIP COMMUNICATION

by

Timothy M. Hollis

A dissertation submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

Brigham Young University

April 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a dissertation submitted by

Timothy M. Hollis

This dissertation has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	David J. Comer, Chair
Date	Donald T. Comer
Date	Michael A. Jensen
Date	Michael D. Rice
Date	Karl F. Warnick

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the dissertation of Timothy M. Hollis in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

David J. Comer
Chair, Graduate Committee

Accepted for the Department

Michael J. Wirthlin
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of
Engineering and Technology

ABSTRACT

CIRCUIT AND MODELING SOLUTIONS FOR HIGH-SPEED CHIP-TO-CHIP COMMUNICATION

Timothy M. Hollis

Electrical and Computer Engineering

Doctor of Philosophy

This dissertation presents methods for modeling and mitigating voltage noise and timing jitter across high-speed chip-to-chip interconnects. Channel equalization and associated tuning schemes have been developed to target the distinct characteristics and signal degradation exhibited in the clock and data signals of multi-Gigabit/second digital communication links. Multiple methods for generating realistically degraded signals for the purpose of simulation are also presented and used to verify the proposed equalization and filtering topologies.

Specifically, a new technique for modeling high-speed jittery clocks in the frequency domain is presented and shown to reduce transient simulation time and memory requirements, while simultaneously improving the timing resolution and accuracy of the simulation by minimizing the dependence on the transient simulation time-step. The technique is further developed to provide unprecedented control over the timing characteristics of the generated signals, and is then extended to the generation of random data signals with definable jitter statistics. Through these techniques,

realistic clock and data waveforms are constructible, providing for the visualization of the combined effects of voltage and timing degradation, while at the same time tracking the phase relationship between the clock and data signals as they pass across their respective channels and through the receiving circuitry of the communication link.

New methods for the automated tuning of second-order continuous-time channel equalizers are proposed based on the simulated or measured single pulse and double pulse responses of the transmission channel. Using only one degree of freedom, the methods target the reduction of inter-symbol interference (ISI) as identified in the single and double pulses. Through tuning either the circuit quality factor (Q), the peaking frequency, or the frequency zero, the methods are shown to adapt to a variety of channel lengths and datarates from the same original equalizer transfer function, implying a good degree of generality, while offering a simple, yet effective, method for ISI reduction.

Finally, the design of an active 5 Gigahertz (GHz) bandpass filter, employed for high-speed clock conditioning, is presented and shown to address both random and deterministic components of the clock signal degradation. The bandpass transfer function is achieved through a combination of AC coupling and a resonant LC tank consisting of on-chip interleaved spiral inductors and a tunable capacitor array. Through adjusting the load capacitance in parallel with the inductors, the center frequency of the filter is tunable over a range of nearly 5GHz. The design targets a supply voltage of 1.2 volts and draws approximately 5.7 milliamps of current.

ACKNOWLEDGMENTS

I would like to start by thanking my wife Alisha, who has stood by me and supported me not just through the process of obtaining the PhD, but through nearly fifteen years of schooling. Even when, after completing a degree in Psychology, I made a U-turn and decided to pursue electronics and engineering (the second best decision of my life following marrying her), she was right behind, encouraging me, willing to make any sacrifice to help me succeed. I would also like to thank my children: Jeremiah, Emily, Samuel, Evelyn, and Isaac who have been very patient with me as I have tried to balance school, work, and being an involved dad. I would like to thank my parents and my sisters for their encouragement and would especially like to acknowledge my grandfather, an engineer himself, who convinced me that I would not be satisfied with anything less than the doctoral degree.

I would like to thank several professors from Brigham Young University, with particular thanks going to Dr. David Comer who has taught me more than just about engineering and carrying out research, but also about being a man of integrity. I have greatly appreciated his trust in me, which he has shown by always encouraging me to pursue inspiration. I wish to acknowledge Dr. Don Comer also for his mentoring. Dr. Comer frequently demonstrated how to think "outside the box" and in doing so often helped me identify unrealized nuances of my work. To the remaining members of my advisory committee, I am also grateful. To Dr Jensen I am grateful for the financial support he provided through the graduate program, which not only relieved my concerns over paying tuition, but also made it possible for me to present my research at international conference events. To Dr. Rice for his insights on particular areas of my study and guiding me to resources that broadened

my understanding of the problem. And finally to Dr. Warnick who, in his humble way, was often able to show me a more mature mathematical approach to what I was working on. I always left his office with a deeper understanding and appreciation for the fundamentals.

I would like to thank two professors from the University of Utah, Dr. Neil Cotter and Dr. Reid Harrison. Dr. Cotter took personal time to meet with me and guide me through some of the most important decisions I had to make as I neared the end of my undergraduate career. In a similar way, Dr Harrison gave me advice, without which I may not have had the opportunity at BYU that I did.

I would like to thank Dan Spangler, who while a leader of the Micron Foundation, spent his valuable time sharing with me insights from the technology industry and also providing advice that I felt very confident following. Within Micron, I would also like to specifically thank Brent Keeth for allowing me to build upon the work I began in school and for challenging me in ways that eventually lead to new levels of development in my personal research. In addition to Dan and Brent, I want to thank both Micron Technology, Inc. and the Micron Foundation for financial support and internship opportunities that have directly and indirectly contributed to this work. I have often been amazed by the level of personal encouragement that I have received from several sources at Micron.

I would like to thank the Intel Corporation also for financial support and an internship that served to focus my direction and the scope of my research very early in the process. I would like to specifically acknowledge Bryan Casper and Frank O'Mahony for their mentoring during and following my internship with the High Speed Signaling group at Intel. Bryan opened my eyes to new perspectives on jitter analysis while Frank took the time to pass along helpful analog circuit design techniques.

Finally, I would like to publically thank my Heavenly Father for sending inspiration in my times of need.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

As datarates approach and surpass multi-Gigabit/second (Gb/s) levels, the challenge of maintaining signal integrity across chip-to-chip interconnects grows due to the introduction of several analog phenomena which impact digital signals in the Gigahertz (GHz) frequency range. Fortunately, many of the parasitic effects of the inter-chip channel are not new and neither is the demand for performance and bandwidth. Over the past century several communication media ranging from the telegraph to fiber optics have been explored and employed to meet the requirements of society. In every case bandwidth limitations have been overcome, or at least mitigated, through the ingenuity of communication engineers, and it is through the leveraging of proven signal conditioning techniques that datarates have achieved their current levels.

Today the push toward ever higher operating speeds in consumer electronics is driven, in part, by growing software complexity. To maintain a given level of perceived performance, added complexity in the underlying software must be balanced or tracked by improvements in processing efficiency. That processing efficiency is not only a function of the clock frequency of the micro-processor unit (MPU), but is also highly dependent upon the available system memory and the rate at which the MPU, memory, and other peripheral components communicate.

One popular prediction of the anticipated growth in software complexity is attributed to Sun Microsystems' Greg Papadopoulos who stated that "the mass and volume of software, (i.e. LOC size[1], memory demands, and processor loading) increase

[1]LOC = Lines of Code

in an inverse natural logarithm relationship to the available processor resources," which, according to Moore's Law is anticipated to double every two years [1].

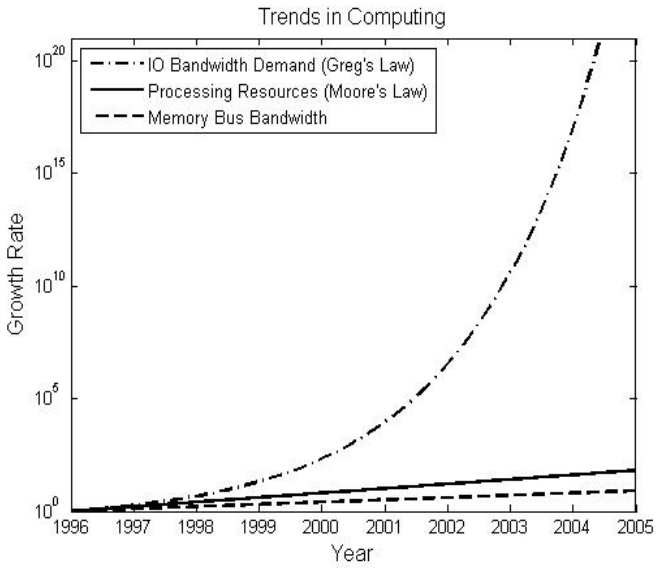


Figure 1.1: Trends in computing speed supply and demand.

In other words, even as MPU operating speeds and computational efficiency increase, the sheer complexity and mass of the associated software obscure much of the performance enhancement obtained at higher clock frequencies. Fig. 1.1 provides a visual comparison of Moore's Law and Greg's Law, clearly identifying the gap between the demand for increased computational power and the achieved growth in computational resources. High definition television, multi-Megapixel digital cameras, music and image file-sharing, as well as the rapid growth in the complexity and detail of graphics emerging from the $30 billion electronic gaming industry [2] are just a few examples of the growing computational load imposed on today's MPU.

To accommodate the market's insatiable appetite for bandwidth, MPUs are forced to share their computational burden with other application specific chips, including memory controllers, graphics processors, etc. Unfortunately the inter-chip

communication link has historically been the limiting factor or bottleneck in overall system performance [3], because while circuits on a single chip are capable of communicating at incredible speeds, communication between circuits located on separate chips is severely impeded by signal-degrading effects inherent in the chip-to-chip signal path. The third curve in Fig. 1.1 verifies this, as it tracks the growth in memory bus bandwidth over the corresponding time period. If Greg's Law may be considered a representation of the demand for inter-chip communication, then there is a terrible discrepancy between the demand for and the achieved inter-chip bandwidth.

Yet, obstacles facing digital communication engineers are not limited to the derivation of signal conditioning circuitry to counter the impact of limited channel bandwidth, but also include the task of developing models and methodologies suitable for capturing and characterizing the newly encountered signal degradation as well as for analyzing and verifying proposed signal conditioning solutions. The cost of initial development and design prototyping has grown so great that the methodology of design iteration is no longer acceptable; rather, designs must function with the first pass. Failing to emulate the true operating conditions, including signal integrity, guarantees failure at multi-Gb/s rates. Conversely, when circuits are exercised in the presence of realistic degradation, success in simulation becomes a better predictor of success within the system.

The challenge associated with simulating channel-affected signals is highly correlated to the characteristics of the degradation. As will be discussed in greater detail, signals in any transmission medium experience both random and deterministic degradation. Random degradation, in the form of random Gaussian distributed voltage noise and timing noise or jitter stemming from several sources, requires statistical quantification. Similarly, deterministic voltage noise and jitter linked to power supply noise, inter-channel crosstalk, impedance discontinuities, component variance, and at high frequencies the response of the channel, result in a variety of observable characteristics, from periodicity to uncorrelated-bounded randomness. To model these noise components correctly requires the ability to designate their probability during the noise generation stage and consequently inject or superimpose these effects onto

the underlying signals and power supplies in a way reflecting what occurs in the actual system.

To date, industry standard simulators do not provide the level of noise and jitter generation control needed to accurately model a realistic communication link. While some of the more advanced, and hence expensive, tools provide for an accurate generation of Gaussian distributed noise and jitter, no simulator in existence allows for the derivation of signals exhibiting the random, periodic, and aperiodic jitter encountered in the real world. [2]

A second challenge in simulating realistic signaling environments is tied to the underlying statistical assumption that a sufficient number of samples of the behavior to be characterized are available. As such, it is becoming necessary to include more and more cycles with each simulation. At the same time, the relative size of each individual noise and jitter component is very small with respect to the overall signal swing and symbol period or unit interval (UI), implying that fine voltage and timing resolution are also necessary. When fine simulated resolution is coupled with a greater number of simulated cycles, the result is an enormous amount of data and prohibitively lengthy simulation times. It is not uncommon for transistor-level transient (time-based) simulations to run for hours or even days. It is also not uncommon for such simulations to fail after several hours due to a lack of memory resources. And in some circumstances, these incredibly long simulations finish successfully, yet the results are not viewable due to the enormous amount of data output by the simulator and the limited capacity of industry standard waveform viewers.

To speed design-to-market time, the growing trend is to compartmentalize system circuitry during the verification process. Rather than simulate the full system at the transistor level, smaller circuit blocks are characterized in Spice-based simulators and then those characteristics are used to construct behavioral models that may be included in simulations at higher levels of abstraction [5]. This methodology is

[2] Agilent Technology's Advanced Design System (ADS) provides a square-wave clock with Gaussian distributed random jitter for transient simulation. This jittery clock source may also be used to trigger a random data source, thereby adding random jitter to the data signal. While the simulated jitter closely approximates a true Gaussian distribution, other jitter components commonly encountered in fabricated circuits are not directly realizable in ADS (periodic jitter, etc.) [4].

very effective when implemented carefully, but has the potential for providing unrealistic performance predictions, as much of the nonlinear circuit behaviors are lost in the translation from transistor-based circuits to behavioral circuits.

In addition to breaking the system down into more manageable blocks, it is not uncommon for voltage and timing noise to be evaluated independently. One of the weaknesses in this approach is that it fails to capture the interaction of voltage noise and timing jitter. As will be shown, voltage and timing noise exhibit a synergistic relationship, wherein each leads to the other and together they combine to limit performance wherever they are encountered.

In this dissertation both the need for enhanced signal conditioning circuitry and the need for improved verification methodologies are addressed. The main contributions of this work to the prior art include:

1. The development of a signal modeling methodology based on Fourier theory which allows for the generation of both periodic clock and random data signals with nearly unconstrained, yet completely controllable, voltage and timing noise characteristics. Because the techniques derive true signals, with both voltage and timing dimensions, the full interaction of voltage and timing noise may be simulated leading to new levels of realism during system verification and signal integrity analysis.[3]

2. The development of an alternative signal waveform generation technique which overcomes some of the limitations of the Fourier-based approach at the cost of some flexibility.

3. The development of self-calibration algorithms for continuous-time data channel equalization targeting the suppression of inter-symbol interference (ISI), the novelty of which is in the simplicity and effectiveness of the techniques, which take repeated samples of the channel's single pulse and double pulse responses

[3]A patent application entitled "Generation and Manipulation of Signals for Circuit and System Verification," was filed on October 14, 2006. Two additional patents have been approved for filing covering the jitter phase control provided by the proposed signal generation technique and an extension of the technique to incorporate finite impulse response pre-filtering of the generated signals.

and tune the frequency response of the equalizer to effectively reduce ISI with only one degree of freedom.[4]

4. The design and implementation of a fully differential 5 GHz bandpass filter with associated center frequency tuning circuitry for reducing clock jitter in source-synchronous serial communications.[5]

In the chapters that follow, more adequate motivation for the development of novel modeling and noise suppression techniques will be provided. Chapter 2 begins by presenting common high-speed electrical signaling topologies and goes on to describe the signal degradation common to such interconnects. It discusses the sources of degradation and then further separates the observable noise into voltage and time-domain components with their many sub-components. With the foundation provided in Chapter 2, Chapter 3 discusses many of the challenges associated with generating waveforms exhibiting realistic noise in a way suitable for and compatible with time-domain simulation. Chapter 3 also discusses the growing problem of simulator efficiency. Chapter 4 goes on to present a new method for generating jittery clock and data signals. In the case of the clock generation, the techniques proposed also facilitate efficient high-speed clock channel simulation. Chapter 5 goes on to discuss existing techniques for mitigating voltage and timing noise imposed by band-limited clock and data channels. Chapter 6 takes a continuous-time equalizer topology and presents new methods of self-calibration which tune the equalizer's frequency response using only one degree of freedom, based on one of two simple algorithms operating on the single pulse and double pulse responses of the channel. Chapter 7 presents a fully differential, LC-based, tunable bandpass filter designed to reduce both random and deterministic degradation of forwarded clock signals. And finally, Chapter 8 summarizes the contributions of this work and suggests paths for continued research in the areas presented.

[4]A patent is being being drafted presently by Micron Technology, Incorporated, covering facets of the proposed equalizer calibration algorithms.

[5]A patent application covering the bandpass filter design and one of the center frequency tuning schemes was filed by the Intel Corporation on December 30, 2005, entitled "Forwarded Clock Filtering."

Chapter 2

High-Speed Interconnects - Topologies and Limitations

Before discussing the several performance limiting phenomena encountered in the high-speed PC board-based communication link, it is helpful to become familiar with the standard link architectures.

2.1 Common Interconnect Topologies

Today's high-speed chip-to-chip communication is dominated by two interconnect topologies, which are both shown in Fig. 2.1. The upper window presents a high level diagram of the source-synchronous link, wherein a reference clock signal, initially in "sync" or phase with the data, is forwarded to the receiver in parallel with the data across a dedicated channel. At the receiving end, this clock, or one derived from it, is used to sample the data waveform during the data detection and recovery process. By routing the clock and data signal paths close together it is hoped that system and environmental noise will impact both signals equivalently. In addition to the close proximity of the clock and data signals, the respective paths are also carefully matched, in terms of length, to insure that commonly experienced noise will remain correlated and cancel out when the forwarded-clock is used to sample and capture the transmitted data. The lower window presents the clock-data-recovery (CDR) architecture, wherein the clock is not forwarded along with the data, but rather is encoded into the data and extracted prior to data detection within the receiver.

For several reasons, true source-synchronous operation is becoming more and more difficult to implement. First of all, the demand for increased aggregate inter-chip bandwidth has been met, in part, through an increase in parallelism or

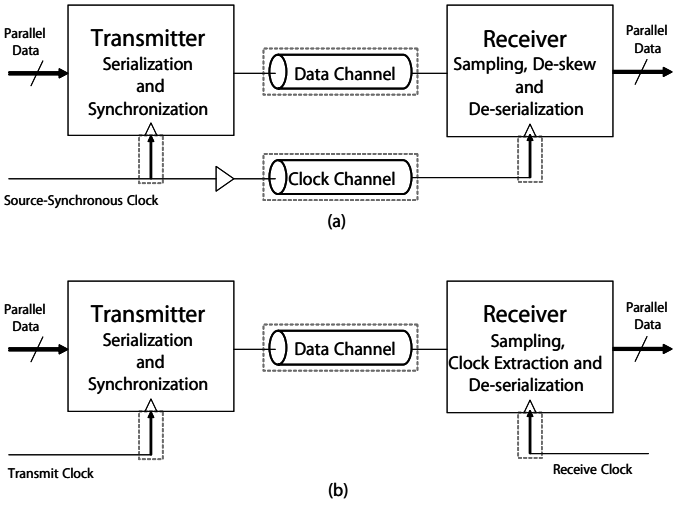


Figure 2.1: Simplified diagrams of source-synchronous (top) and clock-data-recovery (bottom) interconnect topologies.

the number of chip-to-chip connections. As a result it is nearly impossible to match routing lengths identically when the simultaneous push to lower production cost limits the number of available PC board layers onto which the signal paths may be laid out. Incidentally, it is this challenge of route matching between chips that has steered signaling standards from the true parallel link to a set of parallel running serial links, as serial communication is less sensitive to propagation delay mismatch between parallel signals.

In addition to the required off-chip matched routing, there is often some degree of on-chip clock and data routing that must be matched just as carefully. This stems from the fact that many data signals are forced to share a common reference clock. Due to the growing cost of pins on the IC package, a single clock is often associated with 8-32 data lanes. When this occurs, the clock must be distributed across the receiving port, which introduces latency in the clock path and potentially de-correlates noise that was still common to the clock and data signals at the receiving end of the off-chip channel as the result of careful off-chip routing. In order to

guarantee that the clock and data signals arrive at the point of data capture simultaneously, it is thus necessary to extend on-chip data wiring to match the propagation delay incurred through the on-chip clock distribution network. The problem with this approach is not so much the complication of having to match clock and data paths, but rather the limited achievable bandwidth of input data buffers. The signal attenuation resulting from the channel, pin, and pad capacitance requires that signals be amplified before being routed any further on chip, but designing an input buffer to provide amplification at multi-GHz frequencies is nearly impossible in standard CMOS technology. Still, the Joint Electron Device Engineering Council or JEDEC has determined that this approach provides the best performance while still meeting area and power requirements, and in so doing incorporated the source-synchronous interconnect with data input buffering for on-chip matched routing into the specification of the most recent memory standard, DDR3, which is intended to operate up to 1.6 Gb/s [6].

To avoid the input buffer dilemma, a growing trend is to capture the data right at the pad, or right as it enters the chip. The difficulty with this technique is that it still requires that a centralized clock signal be distributed across the input port, and in so doing guarantees a path mismatch equal in length to that of the clock distribution network. This is typically resolved by introducing a delay-locked loop (DLL) or a phase-locked loop (PLL) into the clock path. The DLL or PLL is then used to compensate for the inherent path mismatch by realigning the timing of the clock and data signals at the point of data capture. While this topology is often referred to as source-synchronous, due to the forwarded-clock, it is more correct to refer to it as meso-synchronous, as the clock and data paths are not strictly matched. A more detailed analysis of meso-synchronous links will be treated in the next chapter.

As was mentioned, in the CDR system shown in the lower window of Fig. 2.1, the transmitted data is still launched onto the channel in the same way, triggered by the transmit clock, but in this case, the clock is not forwarded to the other chip. Rather, the clock is embedded into the data bitstream through encoding at the transmitter, and is extracted at the far end for use in the data recovery process.

By embedding the clock into the datastream, correlation between the two signals is guaranteed at the cost of added receive-side complexity. While CDR topologies are finding increased popularity within the realm of electrical signaling, they are more often encountered in optical systems where simply laying out a parallel trace for a forwarded-clock is not possible [7].

2.2 Signal Degradation

As datarates increase, chip-to-chip signaling grows more challenging. Even in the ideal case (e.g. no signal degradation), the decreasing cycle time or UI demands faster circuit operation. At some point, even an ideally received data symbol will become impossible to detect correctly when the available sampling window falls below the setup-and-hold time required by the receiver. Noise, or distortion, only exacerbates the issue. Interestingly, it was the inherent immunity of digital communication systems to noise that made them so attractive in the first place. But as lowpass channel filters and reshapes the sharp edges of high-speed digital signals, the struggle to overcome noise and salvage performance becomes an analog design problem.

Fortunately, analog signal conditioning techniques are fairly mature, as analog communication has always been more sensitive to noise. Yet implementation of theoretically derived noise mitigation schemes is often not straightforward and many techniques must be altered through innovation to be useful at the high datarates presently targeted. In addition, new (previously inconsequential) noise is emerging directly as a result of higher frequency operation.

And while some signal conditioning techniques may address multiple noise components, the distinct nature of the various noise sources commonly encountered in baseband digital communications demands individualized solutions if optimal noise suppression is to be obtained. Similarly, to realistically represent the variety of noise components encountered in the typical inter-chip channel environment, it is critical to account for several unique characteristics including correlation or non-correlation to the signal swing and frequency, statistical characteristics, spectral content, etc. Thus before any solutions may be developed, whether addressing noise suppression

or simply noise modeling, it is first necessary to be familiar with the characteristics of the specific degradation to be addressed.

Noise, which in the broadest sense is manifested as deviations in the characteristics of a signal from ideal, must be considered in two dimensions: voltage noise or distortion along the vertical or amplitudinal axis and timing noise or jitter along the horizontal axis. Amplitudinal deviations in a given signal from ideal levels will be combined under the term voltage noise or simply noise through the remainder of this work. Similarly, deviations in the timing of significant signal events (e.g. transitions, etc.) from ideal are likewise lumped under the term timing noise or jitter. A common way to observe such cycle-to-cycle variation is through superimposing several consecutive cycles of simulated or measured waveforms to generate an eye diagram (see Fig. 2.2). Then by taking a vertical cross section of the eye at a specific point in time, the variations between the many levels at which the signal crosses that point in time are considered the voltage noise experienced over the captured cycles. Similarly, a horizontal cross section, typically taken at the level mid-way between the high and low binary levels of the signal identifies the signal jitter as the varied time points at which the transitioning signal passes through the threshold.

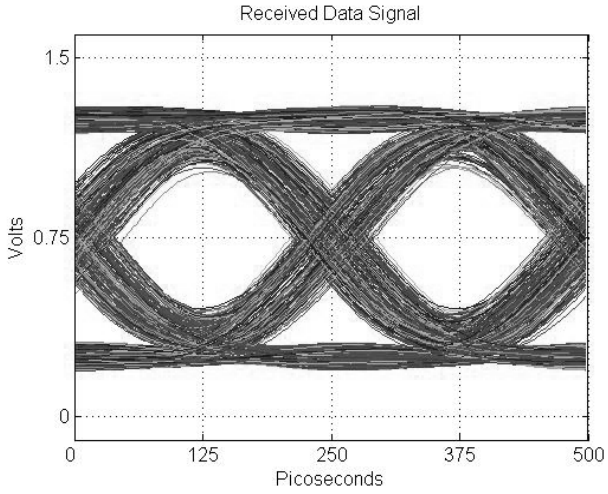


Figure 2.2: Example data eye diagram.

Both voltage noise and jitter are made up of several contributing factors, and as will be shown later, while noise and jitter may be injected into the signal independently, by the time the signal has passed through the next system block in the communication link, the noise and jitter exhibit a strong correlation. Over the next several pages, both noise and jitter will be decomposed and the sources of the individual components will be identified.

2.2.1 Voltage Noise

Voltage noise sources may be separated into two categories: proportional noise sources and fixed noise sources. Proportional noise sources exhibit a dependence on the signal swing while fixed noise sources are considered independent of the signal. To understand the implications of this statement it is necessary to introduce the term signal-to-noise ratio (SNR). SNR quantifies the ratio of the signal power to the observed noise power. Not only does it provide an intuitive description of the quality of a given communication link, but it can be used directly to predict both the achievable bit-error-rate (BER), or the number of bits that may be transmitted error-free, and capacity of the link when the channel bandwidth is known [8, 9]

To calculate the SNR of a particular link, it is first necessary to identify all of the contributing noise sources and separate them into the two categories just mentioned. Then by following the procedure found in [8], the SNR and corresponding BER may be computed. First, a value representing the total independent or random noise V_N is computed through combining the rms levels of all uncorrelated noise sources through the expression:

$$V_N = \sqrt{\sum_i V_{Ni}^2} \tag{2.1}$$

where V_{Ni}^2 is the variance of the ith contributing source. The next step is to compute the signal power. For the purposes of calculating the BER, it is useful to let the signal value include all deterministic noise sources. Thus the signal level is found as:

$$V_S = \frac{\Delta V}{2} - V_D \tag{2.2}$$

where ΔV is the peak-to-peak signal swing and V_D is the peak bounded noise level. This value may be generated through the overly pessimistic summation of the peak-to-peak levels of all deterministic noise sources or through a more elegant technique referred to as "peak distortion analysis" [10, 11]. By combining (2.1) and (2.2), the expression:

$$VSNR = \frac{V_S}{V_N} \tag{2.3}$$

may be used to calculate the voltage SNR. When considering noise as the only source of signal degradation, the expression:

$$P_{error} = \exp\left(-\frac{VSNR^2}{2}\right) \tag{2.4}$$

may in turn be used to compute the probability of error. Then based on the known bandwidth of the channel, Shannon's Theorem [12, 13] predicts that the link capacity is found through:

$$C = BW \log_2\Big(1 + SNR\Big) \tag{2.5}$$

where BW equals the channel's 3 dB bandwidth and SNR is considered here in terms of power rather than voltage.

Based on these expressions, it is clear that the performance of a communication link is highly dependent on the SNR, both in terms of achievable BER and capacity. Thus when the noise power grows while the signal power remains constant, the link performance is expected to degrade. However, when the noise is independent of the signal characteristics, then the SNR may be improved by increasing the signal power or swing. Conversely, when the noise is proportional to the signal, then increasing signal power simultaneously increases the noise power and the SNR, in theory, remains constant or may even decrease.

Two of the more pervasive proportional noise sources common to high-speed digital links are crosstalk (both inductive and capacitive) and simultaneous switching output (SSO) noise. SSO noise corresponds to the coupling of noise between transmit drivers. This noise is not necessarily crosstalk, by the standard definition, but rather results from imperfect power distribution. Ideally an unlimited amount of current is available to the circuits on-chip through a zero-resistance, zero-inductance supply network. In reality, the available current is finite and the supply network exhibits low resistivity and low inductance at best. The result of these nonidealities is that when relatively high-power driver circuits draw current from the power distribution, the resulting spikes in current generate short term voltage drops across the finite resistance between supply-line nodes, resulting in reduced bias conditions for neighboring drivers.

While crosstalk and SSO noise significantly contribute to the degradation of high-speed links, they are not the emphasis of this work. They have, however, been covered extensively in the literature [14, 15, 16, 17]. In addition, while crosstalk is often suppressed through careful layout and routing techniques, special circuits have also been developed to reduce its impact on performance [14, 15, 9]. SSO noise has also been addressed, with most approaches based on modifying driver topologies to reduce slew-rates and high/sharp current draw from the supply [18, 19, 20]. SSO noise and crosstalk may also be reduced through special data encoding as well as a technique known as data bus inversion (DBI). DBI consists of inverting all or some of the parallel data bits prior to transmission in accordance with an algorithm determined to lower the potential noise. Such algorithms may be based on minimizing the number of parallel transitioning bits or may simply seek to reduce the number of transmitted *ones* or *zeros* for power conservation.[1] In either case, an additional signal must be added to the bus to indicate that bus inversion has taken place. The additional cost of the DBI implementation and parallel interconnect must be weighed with the noise-suppressing ability of the technique.

[1] An alternative DBI algorithm based on balancing the number of simultaneously transmitted *ones* and *zeros* across the bus has been approved for patent filing.

In addition to the proportional noise sources, the two most common noise components which exhibit little dependence on the signal swing are random noise and inter-symbol interference (ISI).

Random noise is the result of random effects such as the random thermal motion of electrons in resistors (thermal noise) or the random fluctuations in current due to the granularity of electron current flow (shot noise). Its random nature makes it easily approximated with a Gaussian probability density function (pdf). This type of noise has probably been studied more than any other. As such, it is only mentioned here, but a more comprehensive treatment is found in [21].

ISI is a phenomenon associated with both the transmission environment and the transmitted signal characteristics, though not signal power as just discussed. Strictly speaking, ISI is the result of overlapping transmitted symbols in the bitstream. This symbol overlap may be due to the close proximity of the symbols in time or it may simply be the overlapping of a forward going symbol with some residual signal reflection. The severity of the distortion is determined by the signal pattern and frequency.

The key to ISI and other deterministic signal degradation is that, by definition, it is predictable and potentially reversible. Three keys to mitigating deterministic degradation are the use of channel equalization techniques to compensate for high frequency losses (a focus of this thesis), better channel termination practices, and the minimization of discontinuities along the chip-to-chip signal path.

For the same reason that the number of routing layers on the board are limited, namely due to cost, the quality of the board material is also often sacrificed to increase the profit margin of the end product. As a result, almost all digital board-based communication is implemented across copper traces on FR4 (flame retardant) fiberglass PC boards. As will be shown, the combination of the copper trace and the FR4 medium imposes two forms of high frequency signal loss, which both attenuate the signal amplitude and spread the transmitted symbol energy in time. These two phenomena are known as the "skin effect" and dielectric loss. The bandwidth

constraints associated with these two effects are accounted for by the following two transfer function expressions, as presented in [9]. The first expressions:

$$H_{skin}(f) = e^{-(1+j)\ l\sqrt{\pi\mu\sigma f}} \tag{2.6}$$

describes the skin effect, or the crowding of current near the surface of the copper conductor at high frequencies. As the current moves out from the conductor's center to its edges, the current density decreases in the core of the conductor, and as a result the copper appears more resistive. According to the expression, this effect is proportional to the square root of the frequency f, the permeability μ, and the conductivity σ of the conductor. Finally the impact of the skin effect grows more noticeable with the length of the transmission path, as referred to in the expression by the parameter l. The second expression:

$$H_{dielectric}(f) = e^{-l\sqrt{\epsilon_r}\ f/c\ tan\ \delta} \tag{2.7}$$

refers to the frequency dependent losses associated with the dielectric properties of the board. In this case, the effect is again proportional to the length l of the channel, but now is also inversely proportional to the wavelength $\lambda = c/f$ of the signal, where c corresponds to the speed of light and f is the signal frequency. And finally, the dielectric loss is proportional to the tangential loss factor of the material $tan\ \delta$. In addition, dielectric losses are also proportional to the square root of the dielectric constant ϵ_r. While these two forms of signal loss are dependent on the physical makeup of the channel and medium (e.g., dielectric thickness, trace thickness and width, trace routing layer, etc.), the skin effect is consistently observed at lower frequencies (1-3 GHz), while above 3 GHz dielectric losses dominate the filtering of the signal.

A good source covering the impact of FR4 on signal integrity is found in [22]. This paper delineates the nonidealities of the PC board medium, discusses how the frequency dependent characteristics of the board impact both analog and digital signals, and then compares standard FR4 with many alternative and more expensive

board materials, in terms of the specific material parameters discussed. Unfortunately, the high cost of more signal friendly materials makes them unacceptable for high volume commodity production.

Interestingly, it is not the signal attenuation, resulting from the skin effect and dielectric losses, which pose the greatest challenge to high-speed digital signaling. Rather, it is nonuniform group delay that causes the greatest distortion. Group delay, defined as the derivative of the phase response of a system with respect to frequency, describes the relative propagation velocities of signals at distinct frequencies. Because the propagation time across the inter-chip channel is frequency dependent, and because digital signals are broadband by nature, spectral components of the transmitted digital pulse arrive at the receiving end of the channel at different times producing a smearing of the pulse, very different from the typical RC-filtered pulse response. This factor is not captured explicitly by equations (2.6) and (2.7), but rather is hidden within the not so constant dielectric constant ϵ_r. As will be shown shortly, the pulse spreading that occurs in high frequency digital signaling can extend over several UI causing symbols to interfere with one another [23].

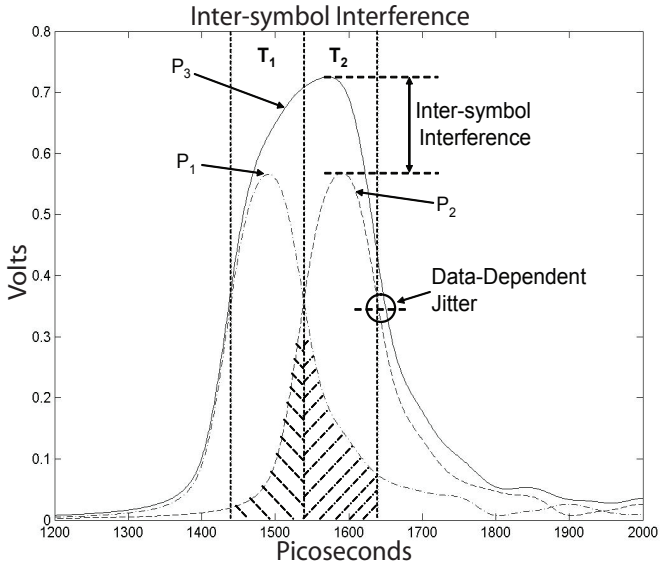


Figure 2.3: Illustration of the impact of ISI on signal amplitude and transition timing.

When sent across an ideal (lossless) channel, all of the energy in a transmitted pulse will be contained within a single time cell or UI. On the other hand, as was just discussed, when a square pulse is transmitted across a channel exhibiting nonuniform group delay, it tends to spread across multiple time cells, as shown in Fig. 2.3. Here P_1 is the simulated 10 Gb/s pulse response of a six inch copper trace on FR4. P_2 represents the same pulse, delayed by one UI. The larger pulse, P_3, is the waveform that results when P_1 and P_2 are sent across the same channel with no intermediate delay, a common occurrence in nonreturn-to-zero (NRZ) signaling.

As Fig. 2.3 shows, a significant portion of P_1 overlaps the cursor, or center sample, of P_2. Likewise, a similar amount of P_2 overlaps the cursor of P_1. This overlap results in the combined waveform, shown here as P_3, in which the bit value sampled at the center of interval T_2 will be larger than that at the center of T_1. The contribution that pulse P_1 makes to the overall value during time T_2 along with the contribution made by pulse P_2 to the overall value during time T_1 is an example of ISI, with the continued voltage accumulation experienced by pulse P_3 indicating the presence of ISI.

In fact, it may be predicted from Fig. 2.3 that the addition of a third consecutive pulse would result in an even larger value sampled during T_3 (the interval immediately following T_2). This is recognized by observing that the post-cursor or tail of pulse P_3 is larger than the tails associated with the individual pulses P_1 and P_2. Thus, the contribution of P_3 to the trailing pulse will be even greater than the previous contribution made by P_1 to P_2 from which P_3 was generated. Consequently the average value of the waveform tends to accumulate with each consecutive pulse.

Fig. 2.4 illustrates this concept. The average of the simulated unequalized curve, corresponding to received data without any signal conditioning, clearly shifts from low to high as the majority of the binary data values change from *zeros* to *ones*. One of the problems associated with such a dynamic shift in the average of the signal is that it eliminates the successful application of a single detection threshold. According to the simulation shown in Fig. 2.4, if the detection threshold were fixed at 0.5 volts, then the true value of the *ones* located near 300 ps, 550 ps, and 700 ps,

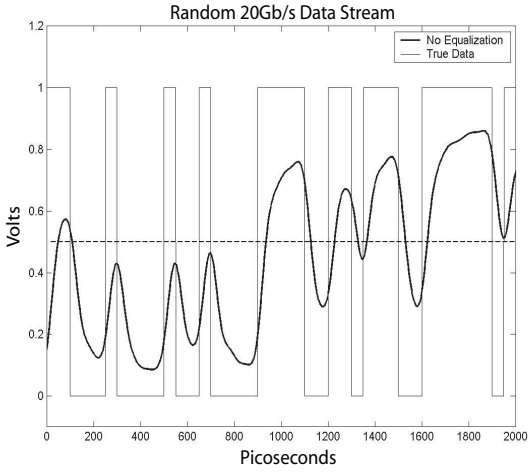


Figure 2.4: Comparison of transmitted data and the corresponding unequalized received data.

as well as the *zero* at 1950 ps, would not be detected as the signal never crosses the threshold during those intervals. And clearly there is no constant level to which the threshold may be adjusted to enable error free detection.

An additional illustration of the detrimental effects of ISI is shown in Fig. 2.5, where the unequalized 20 Gb/s pulse response shown in the upper window produces the completely closed eye found below. The single pulse response is delayed by one UI and included to illustrate the accumulation of ISI through comparing the relative sizes of the single and double pulse tails. The shaded area between the tails represents an accumulation of ISI, and provides the foundation for one of the channel equalizer self-calibration schemes proposed in a later chapter.

Now digital communication by way of electrical signaling is not the only system environment plagued by ISI. In fact, techniques for mitigating ISI have been developed over decades through several parallel efforts ranging from telephony to magnetic storage. Even in the low-loss environment of optical communications, ISI has played a dominant role in limiting bandwidth, and such is the case in all dispersive communication channels regardless of the transmission medium. When Lucky first proposed an adaptive equalization topology in 1965, it was in an effort to surpass

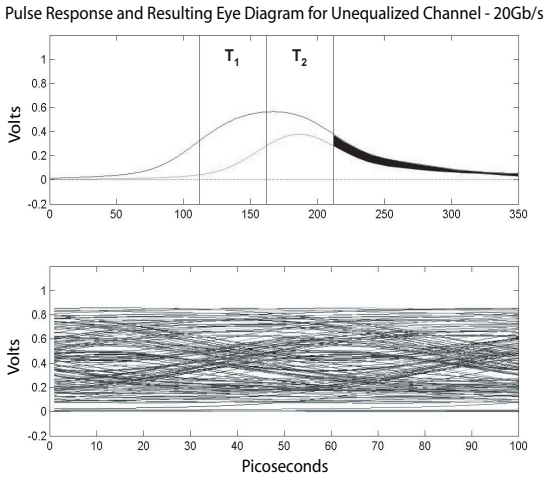


Figure 2.5: The upper window presents the 20 Gb/s single and double pulse responses of the six inch FR4 channel with no equalization. The lower window presents the resulting 20 Gb/s eye diagram. The shaded area in the upper window represents accumulating ISI.

what was then a seemingly unattainable goal of 2400 b/s across telephone lines[24], while today both electrical and optical signaling aim for data-rates in the tens of Gb/s [25, 26, 27, 28].

While the underlying cause of the degradation is different from that observed in high-speed electrical signaling, it may still be addressed and mitigated through similar techniques. In fact, much of the development of the decision feedback equalizer (DFE), to be discussed, has come from efforts to reduce ISI and pattern-dependent jitter (PDJ) in magnetic read channels [29, 30, 31, 32].

2.2.2 Timing Noise - Jitter

Jitter is often, though not exclusively, the result of voltage noise. Amplitudinal shifts in the common-mode level of a signal occurring near the transition causes the signal to pass through the transition threshold at an instant either preceding or delayed from the expected transition time. As long as the shifts in signal voltage level remain smaller in magnitude than the underlying signal swing, then the noise to

jitter translation occurs linearly and is computed by dividing the voltage variation by the slew-rate of the signal transition or the transition slope near the crossing point as illustrated in Fig. 2.6. As will be shown, this interdependence of noise and jitter may be exploited during the signal generation process when signals with explicit jitter are required for simulation.

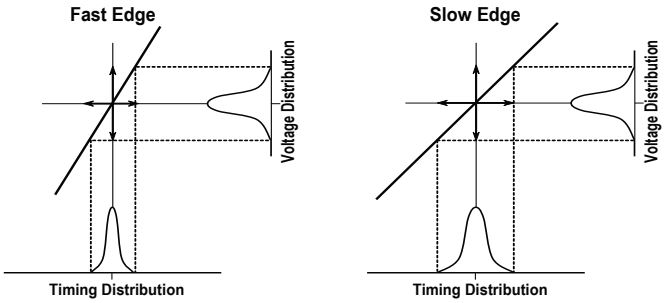


Figure 2.6: Illustration of the translation of random noise to random jitter through the slew-rate of the signal transition.

To understand how timing uncertainty plays a more dominant role in the band-limitation of multi-Gb/s communication links, consider what might be referred to as the "aspect ratio" of a 10-20 Gb/s data eye. While the vertical or voltage dimension of an open eye might be limited to a few hundred millivolts, the horizontal axis or time dimension cannot exceed 50-100 ps, assuming binary or two-level pulse amplitude modulation (2-PAM) signaling. This represents an aspect ratio of approximately 1,000,000:1. From a practical perspective, ensuring a receiver sensitivity and input offset better than tens of millivolts is much simpler than providing phase or timing control with picosecond resolution. A more scientific explanation for the growing concern over timing margin is presented in [9].

There are methods, however, which increase the available timing window. For example, four-level signaling (4-PAM) doubles the symbol period, but simultaneously reduces the SNR by a factor of three. In some circumstances, this trade-off

between timing and voltage margins may be warranted, though 2-PAM signaling remains the most widely accepted standard. And as 2-PAM is more commonly encountered, focus has turned to the eye closure along the time axis.

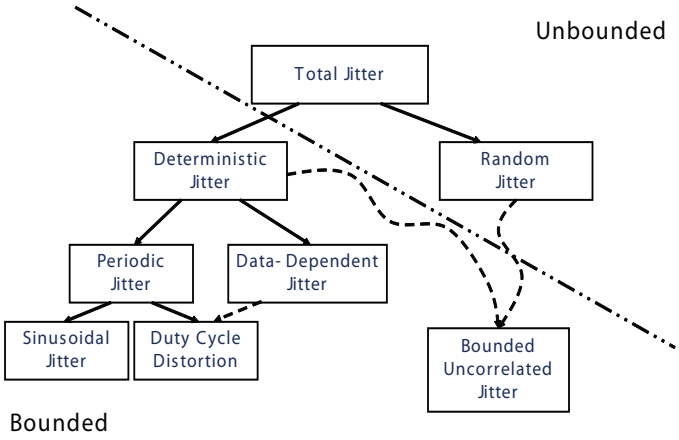


Figure 2.7: Decomposition of jitter.

In the attempt to obviate the eye-closing effects of jitter, it is important to identify all of the contributing factors, and recognize that a comprehensive solution must address the distinct characteristics of the many jitter components. As illustrated in Fig. 2.7, while jitter may be decomposed into several subcomponents, it is often useful to separate all jitter into two main categories: bounded or deterministic jitter (DJ) and unbounded or random jitter (RJ). Both classes of jitter represent a severe impediment to high-speed communication, but it is the less bounded nature of RJ that makes it the culprit in long term system failures [33, 34]. The RJ is in fact bounded in reality, but is unbounded in the stochastic model.

One important distinction between deterministic and random jitter is their probability distributions. DJ, being bounded in nature, may be quantified with a peak-to-peak value, while RJ, being unbounded, is typically approximated with a Gaussian probability distribution and its corresponding standard deviation (rms), in

accordance with the Central Limit Theorem of statistics. Thus DJ never exceeds a given limit while the potential magnitude of RJ is unlimited with the caveat that encountering larger and larger values becomes less and less likely. The total jitter (TJ), computed through the convolution of deterministic and random components, is dominated in the short term by DJ and over the long term by RJ. Because the TJ contains an unbounded random component, the TJ is also unbounded and hence is most appropriately quantified with respect to a given BER [33].

The remaining subcomponents of DJ, as presented in Fig. 2.7 are periodic jitter (PJ), which is commonly manifested as a sinusoidal modulation in signal phase, and data-dependent jitter (DDJ) which most often corresponds to ISI. Duty cycle distortion (DCD), which will be discussed in more detail, is sometimes treated as a subcomponent of DDJ. While it is true that DCD can further exacerbate DDJ, as will be shown, DCD is more accurately described as a periodic component.

As expected, some useful information may be gleaned from the pdfs of the various individual jitter components. Two well known sources discussing the specific characteristics of the various jitter components are [35] and [36]. In [36], the specific jitter pdfs are employed to decompose the total jitter into its individual components. By so doing, the root causes for any associated link failure become clear and addressable.

RJ, which as was mentioned is often assumed to exhibit a Gaussian probability distribution and is consequently quantified with an rms value, is typically associated with random perturbations in the signal amplitude. Such variations in amplitude occurring at or near signal transitions lead to a corresponding variation in the reference voltage crossing time of the signal, due to finite signal risetime and falltime. As illustrated previously in Fig. 2.6, this translation of voltage noise to jitter is inversely proportional to the signal slew-rate. As shown in the figure, a voltage noise distribution is translated into jitter through a fast edge, while the same process occurs on the right side through a slower edge.

This gives rise to some important trade-offs in the signaling design: higher slew-rates limit random noise-to-jitter translation, but lower slew-rates tend to minimize inductive effects, such as ringing in the signal as well as inductive and capacitive crosstalk, thereby reducing some components of the noise. In addition, because lower slew-rates are also associated with lower channel bandwidths, noise is filtered by the channel characteristics and attenuated at higher frequencies just as the signal is. This does not imply, however, that purposely limiting the channel bandwidth to reduce noise will simultaneously reduce jitter. In [9], this very circumstance was analyzed for the simple case of a single pole, lowpass channel. It was determined that while reducing channel bandwidth did reduce the overall noise magnitude, the consequential degradation in the slew-rate resulted in a more aggressive translation of noise to jitter. Specifically it was calculated that a 75% reduction in channel bandwidth could increase the signal jitter by a factor as large as ten. Therefore, a compromise is to increase channel bandwidth through whatever means possible, while providing explicit slew-rate control at the drivers to minimize inductive effects and crosstalk.

The increased RJ resulting from the effects of band-limitations on signal slew-rate is often referred to as jitter amplification, and increases with datarate for a given channel. Jitter amplification is not limited to RJ, but rather quantifies the magnification of all jitter that occurs as slew-rates degrade.

To determine the level of jitter amplification requires the system's jitter impulse response, often simulated by a single edge timing deviation within an otherwise ideal periodic signal. The number of trailing cycles required for the edge timing to re-settle to the ideal is a distinct characteristic of the system. The jitter impulse response is found by measuring the difference between the ideal edge timing and the timing due to the perturbation, which are represented by a train of delta functions whose individual magnitudes correspond to the jitter magnitudes of the sequential edges (see Fig. 2.8).

Once the jitter impulse response is acquired, the jitter amplification factor may be computed through the expression:

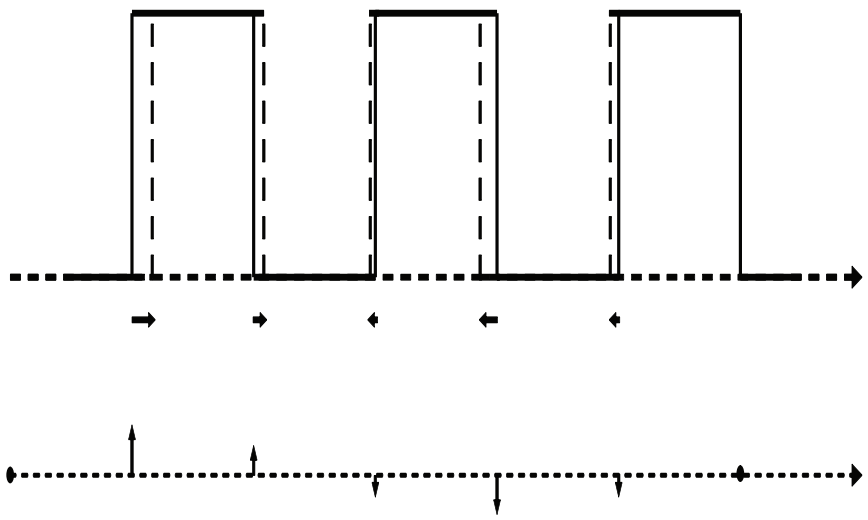


Figure 2.8: Definition of the jitter impulse response.

$$J_{Amp} = \sqrt{\sum_i JIR_i^2} \tag{2.8}$$

where JIR_i are the sampled values of the jitter impulse response between the initial occurrence of the perturbation and the final edge settling time. In the case of RJ, the jitter amplification factor may then be employed as a scaling term by which the known rms jitter level at the input of a system is multiplied to compute the expected output rms jitter level.

Based on this principle of noise-to-jitter translation, one potential method for minimizing RJ is to minimize the random noise. As will be discussed in detail, the most widely accepted method for addressing and reducing random noise components, or conversely increasing the SNR, is through matched filtering, in which the impulse response of the filter is the time reversed, delayed conjugate of the transmitted pulse. Mathematically it can be shown that the convolution of the transmitted symbol with the impulse response of the matched filter optimizes the SNR for the case of random noise, uncorrelated to the signal [10, 37].

While most RJ is associated with random noise near signal transitions, some RJ may have origins not as clearly linked to voltage noise. For example, the phase noise inherent in commonly used oscillators modulates the edges of the oscillator output with a nonlinear relationship to environmental factors.

As was mentioned DJ, sometimes referred to as systematic jitter, can be broken down into several sub-categories including DCD, DDJ due to ISI, and various uncorrelated jitter components injected into the signal through the power supply and ground paths. In the following pages, the characteristics of DCD and DDJ are discussed.

DCD is simply duty cycle error quantified in terms of absolute time. DCD exists when the ratio of the signal pulse-width to the period deviates from 1/2 due to DC offsets in the signal, rise/fall time discrepancies, device mismatch in the signal path or any combination of the three. Inequalities between the pulse and space-widths of clock signals are particularly troublesome in double-data rate (DDR) systems, where the data stream is sampled with both the rising and falling edges of the clock.

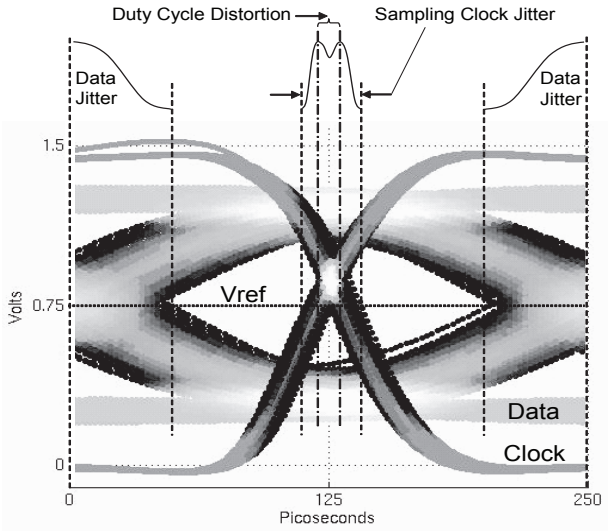


Figure 2.9: Eye diagram illustrating the effects of both clock and data jitter on timing margin. Duty cycle distortion produces the bi-modal sampling clock distribution.

Alone, the pdf of DCD consists of two Dirac delta functions with heights of 0.5 each, separated by the peak-to-peak DCD magnitude. When combined with random, Gaussian distributed jitter, DCD produces a bi-modal jitter distribution, as illustrated in Fig. 2.9. DCD is not the only jitter component that leads to the bi-modal jitter pdf. In fact, it is so common for TJ to take on the bi-modal form that methods for jitter decomposition have been developed based on the assumption that TJ may always be approximated as bi-modal [36].

With reference to the figure, while the ideal sampling instant (clock edge) should cross the vertical midpoint at the center of the data eye, the presence of DCD results in the concentration of clock threshold-crossings around a pair of timing instants, with the distance between the bi-modal peaks in the TJ pdf corresponding to the peak-to-peak DCD. Thus, the contribution of DCD to the spreading of the sampling distribution, and subsequent timing and voltage margin degradation, is significant.

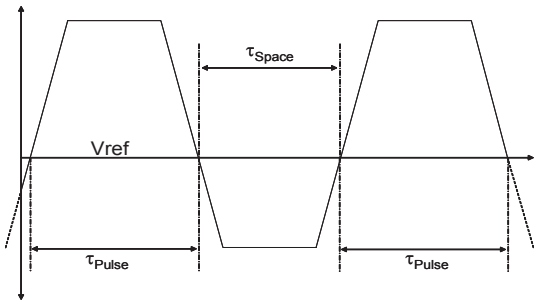


Figure 2.10: Illustration of how the addition of DC offset to a perfectly symmetric, finite rise/fall time, square wave generates duty cycle error.

The analysis of DCD requires consideration from both low and high frequency perspectives. Recall that one of the effects of the lowpass channel is to degrade the rising and falling signal transitions. The exaggerated rising and falling transitions shown in Fig. 2.10 help to demonstrate the dependence of duty cycle on DC offsets.

The signal shown is nothing more than a symmetric square wave that has been shifted in the positive vertical direction by a small amount. That small shift, in conjunction with the finite slopes of the transitions produces a shift in the reference voltage crossing times of the signal, and hence, duty cycle error. For the reference voltage shown, the duty cycle ($\tau_{Pulse}/(\tau_{Pulse} + \tau_{Space})$) is clearly greater than 50%. And while the presence of DC offset is not the only source of duty cycle error, an unwanted DC component tends to accumulate as a result of DCD, regardless of the source of the error, as illustrated in Fig. 2.11, which demonstrates the effect of lowpass channels on clock signals with duty cycle greater than 50%.

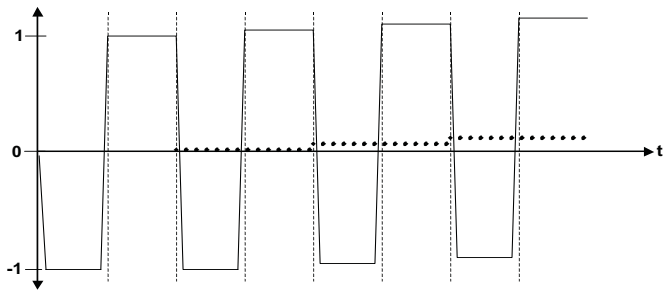


Figure 2.11: Illustration of how DCD in a signal accumulates across a lowpass channel.

With regard to the diagram, the mismatch between the positive and negative pulses results in a non-zero DC or average value due to the integrating nature of the channel (i.e. the area under the pulses do not cancel completely). Then, in accordance with the previous discussion surrounding Fig. 2.10, DCD will grow due to the increased offset. Thus, a cycle is born wherein DCD leads to increasing signal offset, and signal offset leads to increased DCD, which suggests that the suppression of low frequency signal components, or at least the DC component, should aid in the attenuation of DCD. This also implies that DCD amplification imposed by the lowpass channel will grow faster as the signaling frequency exceeds the bandwidth of the channel, and the rate of signal integration increases.

The high frequency nature of DCD can best be understood through Fourier analysis. A simple Fourier series, which models a clock with controllable levels of DCD, may be derived as follows:

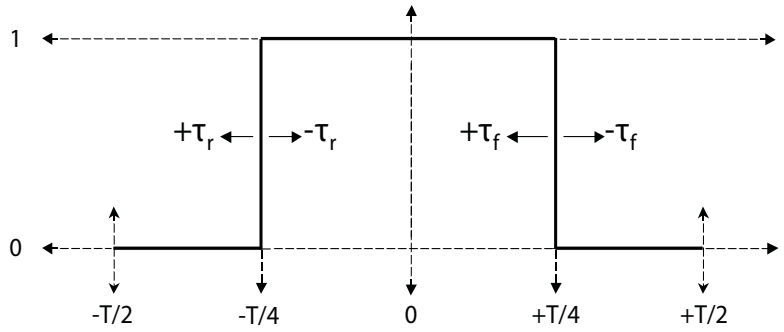


Figure 2.12: Waveform used in the derivation of the Fourier series representing a clock with DCD.

1. The waveform shown in Fig. 2.12 represents a clock signal which alternates between values of zero and one with period T. By including the variables τ_r and τ_f at the transitions it is possible to simulate the existence of duty cycle error through the manipulation of the rising and falling edges of the pulse as follows:

   Positive τ_r shifts the rising edge left (early),

   Negative τ_r shifts the rising edge right (delay),

   Positive τ_f shifts the falling edge left (early), and

   Negative τ_f shifts the falling edge right (delay).

2. The expression into which the Fourier coefficients will be inserted is:

$$C(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi}{T}t\right) + B_n \sin\left(\frac{2n\pi}{T}t\right)$$

where

$C(t)$ = the resulting clock signal,

t = the timing instant,

T = the signal period, and

n = the integer multiple frequency (harmonic).

3. The A_0 term, found by evaluating the integral:

$$A_0 = \frac{1}{T}\int_{-\frac{T}{4}\pm\tau_r}^{\frac{T}{4}\pm\tau_f} dx \tag{2.9}$$

represents the DC or average value of the waveform.

4. The A_n and B_n terms are similarly found by evaluating the following integrals:

$$A_n = \frac{2}{T}\int_{-\frac{T}{4}\pm\tau_r}^{\frac{T}{4}\pm\tau_f} \cos\left(\frac{2n\pi}{T}x\right)dx \tag{2.10}$$

and

$$B_n = \frac{2}{T}\int_{-\frac{T}{4}\pm\tau_r}^{\frac{T}{4}\pm\tau_f} \sin\left(\frac{2n\pi}{T}x\right)dx \tag{2.11}$$

and represent the harmonic content of the waveform.

5. The resulting coefficient values are:

$$A_0 = \frac{1}{2}\left(1 + \frac{2(\tau_r - \tau_f)}{T}\right), \tag{2.12}$$

$$A_n = \frac{1}{n\pi}\left[\sin\left(\frac{2n\pi}{T}\left(\frac{T}{4}+\tau_f\right)\right) - \sin\left(\frac{2n\pi}{T}\left(\tau_r - \frac{T}{4}\right)\right)\right], and \tag{2.13}$$

$$B_n = \frac{1}{n\pi}\left[\cos\left(\frac{2n\pi}{T}\left(\tau_r - \frac{T}{4}\right)\right) - \cos\left(\frac{2n\pi}{T}\left(\frac{T}{4}+\tau_f\right)\right)\right]. \tag{2.14}$$

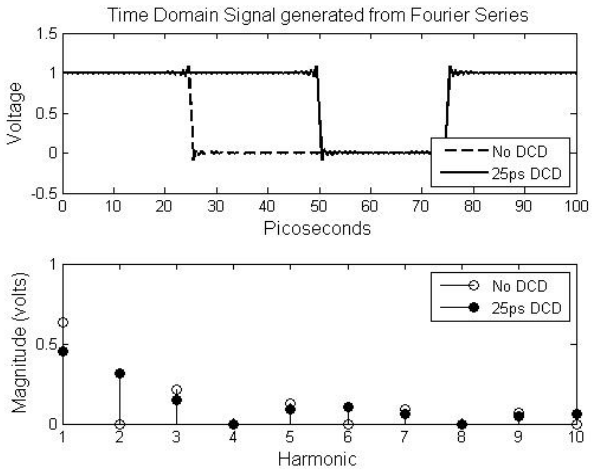


Figure 2.13: The upper window presents an ideal clock waveform compared with a clock exhibiting 25 ps of DCD as generated through the parameterized Fourier series just derived. The lower window presents the resulting variation in the 10 GHz fundamental and the first nine higher order harmonics, illustrating the high frequency nature of DCD.

Fig. 2.13 illustrates the effects of duty cycle error on the high frequency components of the clock signal. A 10 GHz clock signal, generated by the Fourier series just discussed, is shown in the upper window. The falling edge is delayed in one case by 25 ps to compare an ideal clock with one exhibiting DCD. The lower window shows the resulting shift in the magnitude of the first ten harmonic components. As these harmonics represent integer multiple frequencies of the fundamental, it can be

understood that DCD manifests itself at frequencies equal to and above the fundamental frequency of the signal. An additional point of interest is the fact that the even harmonics, which do not exist in the ideal signal, take on nonzero values as the duty cycle error increases, with the second harmonic appearing to be the dominant DCD component. This last fact is corroborated in [38].

Moving on to DDJ, it has become acceptable in casual conversation to use the terms ISI and DDJ interchangeably. This is a mistake, because though they are related phenomena, they are not equivalent. As was illustrated previously in Fig. 2.3, ISI refers to the vertical shifting in the signal amplitude that results from the additional positive or negative impact of neighboring bits in the data stream. DDJ is the deviation in edge timing that results from the same bit-to-bit interaction [39, 40, 41, 42, 43].

Interestingly, [41] goes on to show that DDJ may exist even when the bitrate is well contained within the bandwidth of the system implying that simple extension of the system or link bandwidth does not guarantee a reduction in DDJ, whereas channel bandwidth extension has been the long accepted method for reducing ISI.

The final jitter component, yet to be discussed, is the uncorrelated-bounded jitter shown in the lower right-hand corner of Fig. 2.7. This jitter is associated with supply noise, ground bounce, and other bounded environmental effects such as electromagnetic interference (EMI). As such, it is bounded, yet unpredictable and therefore not strictly deterministic.

2.3 Impact of Noise on Link Performance

Having laid a foundation through presenting the most common link architectures, as well as the dominant sources of signal degradation, it is now appropriate to discuss how those forms of degradation impact the performance of the meso-synchronous link, which is the signaling topology targeted in the remainder of this work. To facilitate the discussion, a more detailed diagram of the typical meso-synchronous signaling scheme is presented in Fig. 2.14.

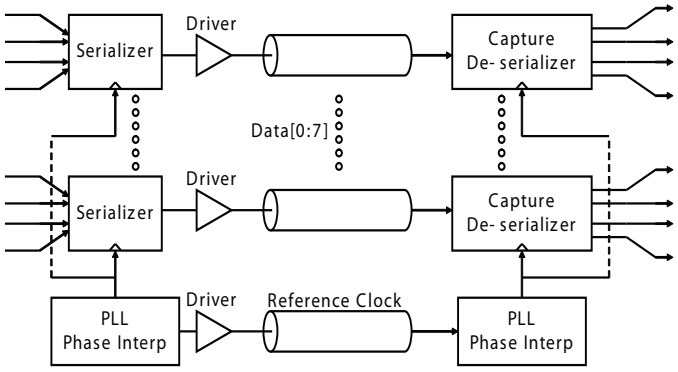


Figure 2.14: Detailed block diagram of the typical meso-synchronous link.

It is important to recognize that circuits contributing to the performance, or lack thereof, of the meso-synchronous signaling scheme begin a few layers before the driving circuits launch the clock and data onto the channel. As shown in the figure, lower frequency, parallel data from elsewhere on the chip is serialized before being fed to the drivers. The multiplexing operation used to serialize the data is triggered by clock edges typically generated from a PLL. While great care may be taken to reduce the signal jitter at the output of the PLL, some jitter is inevitable, and superimposed onto the data edges through the serialization process.

Following serialization, the driving circuits pull down on the power supply network generating the SSO noise previously discussed. Across the channel, both inductive and capacitive crosstalk occur due to the close proximity of the traces required to accommodate the number of routes. Signal reflections, due to discontinuities presented by connectors, vias, and possibly transitions to the distinct dielectric properties of additional circuit boards in the transmission path, combine with the transmitted signal either constructively or more often destructively. Frequency dependent losses in the PC board attenuate and smear the digital symbols causing ISI and the associated DDJ. At the receiver, the shared clock is distributed out to each of the data capturing circuits, a process through which the clock is vulnerable to additional supply and environmental noise, based on the sensitivity of the distribution

network. To compensate for the inherent clock-data routing mismatch of this scheme, a second PLL or possibly a DLL is used in conjunction with phase interpolation (PI) to realign the clock-to-data timing. The introduction of the PLL/PI into the clock path further reduces the correlation between noise and jitter that were once common to the clock and data signals.

Based on this discussion, it is not uncommon for the clock and data signals reaching the data capture mechanism to resemble the simulated 4 Gb/s signals shown in Fig. 2.15. To identify the synergistic relationship of clock and data jitter, the rising and falling edges of the corresponding sampling clock are overlaid. With the signal concentration accounted for by the shade of the waveform (higher concentration = lighter shading) it is possible to visualize, albeit crudely, the distribution of the signals in both the voltage and time dimensions. Ignoring for a moment the exact distribution of the noise and jitter, their general impact on the system performance can still be analyzed.

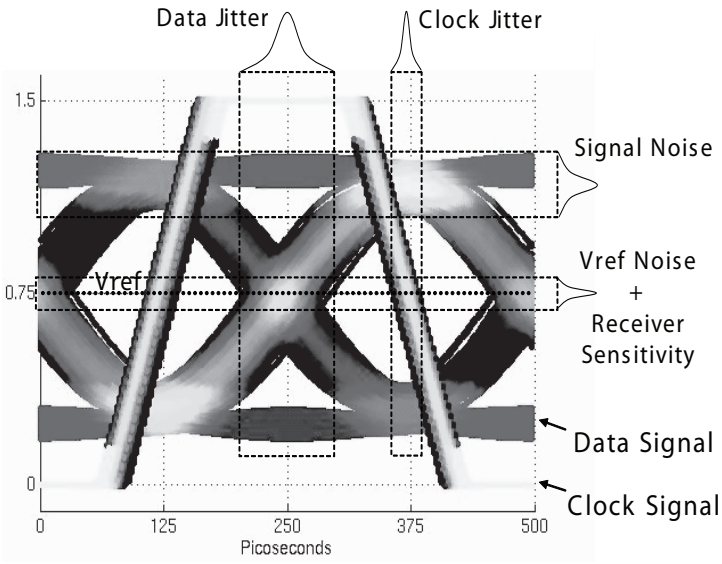


Figure 2.15: Diagram identifying various forms of signal degradation.

Even without the signal shading, it is clear that the timing uncertainty of the data signal is significantly greater than that of the clock. In this particular case, and in general, the data timing variation is dominated by DDJ stemming from ISI. In the figure, the closing of the data eye is the combined result of ISI, SSO noise, crosstalk, and other random noise components. It is the broadband nature of the data that makes it so sensitive to these "high-frequency" noise sources. While the clock passes over and is reshaped by a similar if not identical channel, its narrow-band periodicity is not affected by the high frequency channel losses in the same way. In fact, while the clock may experience SSO noise and crosstalk, depending on its proximity to the noisy data signals, it is immune to ISI, though it will experience both attenuation and jitter amplification.

Yet even though the clock signal integrity is typically superior at the receiving end of the link, it can no longer be taken for granted, for while clock jitter directly reduces the timing margin, it also indirectly reduces the voltage margin. To understand this, consider what happens as the sampling uncertainty or clock jitter increases. The result is that data sampling can occur further and further from the horizontal center of the eye. From the figure it is clear that when the rounded data eye is sampled near the transitions, the value sampled over that region in time will have less amplitude with respect to the reference voltage, and hence less voltage margin.

Noise on the reference voltage (V_{REF}), which serves as the detection threshold, also contributes to the perceived eye closure. V_{REF} uncertainty, which often consists not only of explicit noise but finite receiver sensitivity as well, may accumulate and directly decrease the voltage margin, while simultaneously reducing the timing margin based on the same argument as that used for the sample timing uncertainty.

Thus it is clear that predicting the probability of error in such systems is significantly more complicated than simply identifying the bounded and unbounded noise components of the data signal, and computing the SNR and BER using the equations presented earlier. Rather, verifying link functionality has become a design

problem all its own. And as will be discussed in the next chapter, innovative methods have been developed in an effort to account for the growing complexity of the verification problem.

Chapter 3

Current Modeling and Simulation Practices

As was indicated during the introduction, the design of high-speed chip-to-chip interconnects is not only impeded by the signal degrading effects of band-limited channels, but also by the difficulty in accurately verifying the interconnect performance prior to fabrication, which is critical as the enormous cost of integrated circuit fabrication prohibits an iterative approach to circuit design and product development. In fact, depending on the process technology node and the geometric complexity of the design, fabrication costs for the first prototype may exceed $1 million [44].[1]

As symbol periods fall into the hundreds of picoseconds range and timing uncertainty can no longer be ignored, the level of jitter introduced by the transmitter, channel, and receiver must be accurately predicted through methods accounting for the interaction of noise and jitter. Thus, the high-speed link verification problem raises two somewhat incongruent challenges: the need to accurately model signals exhibiting noise and jitter and the ability to efficiently simulate the interaction of the resulting noisy signals with the various system components. This chapter discusses the trade-offs between simulation precision and efficiency in standard modeling methodologies, and goes on to present known methods for generating signals with deterministic jitter.

3.1 Modeling Efficiency versus Precision

At multi-Gb/s data rates, the statistical nature of signal degradation, coupled with the already vanishing voltage and timing margins, has led to advances in

[1] This estimation was associated with the 90nm process node.

channel and circuit modeling. Alternative computational algorithms have been incorporated into existing simulators to complement traditional circuit analysis, while at the same time, high-level tools like Matlab and Simulink are finding greater use in the verification process. To efficiently capture the true impact of the entire communication link on signal integrity with the requisite level of precision requires an interleaving of simulation at both the transistor and system levels.

3.1.1 Transistor-level Analysis

Transistor-level analysis refers to the schematic entry of specific circuit blocks into Spice-like tools such as HSpice, PSpice, Cadence, and ADS for AC or transient analysis; complementary methods for determining signal integrity. AC analysis computes the frequency response of the channel or circuit and can help identify noise components and other degradation most visible in the frequency domain. Unfortunately, AC analysis is only carried out for a fixed circuit bias condition, while transient analysis provides a time domain simulation of the circuit behavior accounting for dynamic changes in the circuit biasing resulting from varying input levels and/or supply noise, thereby presenting the *real-time* impact of environmental conditions on passing signals.

During transient analysis, differential equations relating the voltage and current at each circuit node are solved at specified points in time. The time that elapses with each computation increases when diodes, transistors, and other components exhibiting nonlinear voltage-to-current relationships are included. To control the simulation run time, the level of precision in both time and amplitude are often adjustable. For example, the desired level of voltage or current resolution in Spice-based tools is designated through the AbsTol (absolute tolerance) parameter. Requiring tighter tolerance leads to a greater number of computational iterations in order to meet an associated error level while solving the differential nodal equations at each time step.

In a similar way, the timing resolution may be enhanced by decreasing the time span between each calculation. While simulators like HSpice, ADS, Spectre (Cadence), and HSim allow for the designation of a minimum transient step size, PSpice does not provide direct control over the minimum time step, but rather provides a maximum time step parameter which constrains the simulator to make at least one evaluation within the designated interval. Thus, for the purpose of jitter characterization, the timing precision of the industry-wide transient simulator is improved through a reduction in the simulated time step, the result of which is a simultaneous increase in both the simulation run time and the memory requirement.

In addition to the requirement of sub-picosecond timing resolution, the statistical nature of random noise and jitter demands that the signal-system interaction be computed over several clock cycles in order to provide the necessarily large number of samples required to properly build up probability distributions. Coupling the constraints of high resolution (small transient time step) with the need to observe the behavior over thousands or millions of cycles extends the transistor-level simulation run time and memory requirements even further.

An attempt to overcome the weaknesses of the general transient simulator has lead to the development of alternative time domain algorithms including harmonic balance, circuit envelope, and periodic steady state simulation. While these techniques have many distinct features, they all seek to avoid or minimize the time step dependency of transient simulation by operating as much as possible in the frequency domain.

As long as the circuit element passing the signal can be accurately modeled as a linear time-invariant (LTI) system, the time-consuming process of convolving the signal with the circuit impulse response in the time domain may be replaced by simple vector multiplication in the frequency domain due to the relationship:

$$A \otimes B = F^{-1}\left\{F\left\{A\right\} \times F\left\{B\right\}\right\}$$

where $\otimes$ denotes convolution and $F\{\}$ is the Fourier Transform and $F^{-1}\{\}$ is the Inverse Fourier Transform. The computational efficiency gained through this substitution is illustrated by considering the time domain convolution of two vectors A and B, which could represent a signal and the impulse response of the circuit through which it is passing. Recall first that the process of discrete-time convolution is carried out through the formula:

$$C(n) = A \otimes B = \sum_{k=-\infty}^{\infty} A(k)B(n-k). \tag{3.1}$$

Due to the finite length of the vectors under consideration, the sum need not be carried out to infinity. Thus, the number of computational steps to perform the convolution is found through:

$$1 + M + N + 2\sum_{k=0}^{N}\Big(M + N - k\Big)^2 - \alpha M^2 \tag{3.2}$$

where

$$\alpha = \begin{cases} 1 & \text{if } M+N \text{ is even} \\ 0 & \text{if } M+N \text{ is odd} \end{cases}$$

and where M and N are the number of elements in the longer and shorter of the two arrays, respectively. According to (3.2), the convolution of two vectors of 1000 elements each would require 4,670,669,001 mathematical steps. This may be contrasted with the number of steps needed to convert the two vectors to the frequency domain, perform an element to element multiplication, and return to the time domain, a process often referred to as Fast Convolution. When the Fourier Transform and Inverse Fourier Transform processes are carried out via the FFT and IFFT, the time-to-frequency and frequency-to-time domain translations require as little as $\frac{1}{2}N\log_2 N$

complex multiplications and $N \log_2 N$ complex addition steps each [45]. For the two equal length vectors under consideration, this leads to a total number of:

$$4.5N \log_2 N + N \tag{3.3}$$

or 45,846 computational steps to convert both vectors to the frequency domain, multiply them and return to the time domain. To be fair, increased accuracy and efficiency in the FFT algorithm is insured by padding each data set with zeros to the nearest power of two greater than the sum of the two data set lengths. Thus for M = N = 1000, the actual number of data points involved in the FFT process will equal 2048, causing the total number of steps in the overall calculation to increase to 103,424, still significantly shorter than direct convolution.

Unfortunately, this reduction in computational steps is only realized when the simulated circuits can be linearized. Thus harmonic balance and the other more sophisticated simulation algorithms tend to divide the simulated system down into those parts which can be appropriately modeled as LTI, and those parts which require nonlinear analysis (e.g., circuits containing diodes and transistors passing large signals) [46, 47, 48]. In circuit envelope simulation, further efficiency is gained by only performing frequency domain multiplication over the spectrum of the passing signal while avoiding unnecessary calculations at unrelated frequencies [48]. The ability of these more sophisticated algorithms to handle nonlinear circuit elements while exploiting the speed of frequency domain calculation is somewhat washed out, as they tend to target radio frequency (RF) circuit design, and in doing so incorporate functionality (complexity), such as signal mixing and intermodulation analysis, not typically considered or even applicable in this type of baseband link verification.

Even with highly-efficient simulators, the number of simulated cycles required to fill in the tails of statistically characterized noise and jitter prohibits a purely time-domain based link analysis. In [49], trade-offs between several possible modeling methodologies were considered with reference to a 20 Gb/s serial link. The proposed solution was to use Verilog rather than transistor-level models to speed up

simulation time. To regain some of the accuracy lost by moving away from transistor-based simulation, the behavioral Verilog models were modified to account for analog phenomena not typically considered. By so doing, modeling time was reduced from hours to minutes, without incurring significant error.

Another common way to minimize dependency on the transient time step is to analyze the resulting signal from within the frequency domain, never returning to the time domain. This is done through the phase noise spectral density. Using a variety of expressions, time domain jitter may be extracted directly by integrating the simulated phase noise over the bandwidth of interest [50, 51].

The discrete-time simulation methodology, inherent in Spice-based simulators, not only limits computational efficiency, but also imposes constraints on the variety of input signals and stimuli derivable from within the tools themselves. While there are a few exceptions, commercially available simulators typically construct signals in a piece-wise linear (PWL) fashion; voltage levels being designated for each step in time. HSpice, PSpice, HSim, Spectre, and ADS all provide for the instantiation of standardized periodic signals with control over the signal amplitude, period, delay, risetime, falltime, and pulsewidth. For complete control over the waveform, an arbitrary PWL voltage source is also available, wherein each time step and corresponding voltage may be specified directly. Using this approach it is possible, though terribly inefficient, to incorporate noise and jitter into the signal model, and for the majority of the simulators mentioned above, this method is the only commonly known means for adding pseudo-random noise in the time domain.[2]

The one exception is ADS, which in addition to the standard square-wave and PWL sources, also provides a pseudo-random data source and a clock waveform with an assignable rms jitter level. The jittery clock signal may be used to trigger the pseudo-random data waveform, thereby injecting Gaussian distributed jitter into the data signal. But even with the added sophistication, ADS does not provide complete

[2]Gaussian distributed noise may be crudely approximated by superimposing a carefully selected set of sinusoids with unrelated frequencies onto the fundamental signal. In accordance with the Central Limit Theorem, the accuracy of the approximation increases with each additional sinusoid, and with the length of the simulation [52].

control over the realized jitter distribution, as there is no utility for generating clock or data signals with periodic jitter components[4].

3.1.2 System-level Simulation

Because simulation time and memory requirements associated with transistor level Spice-based evaluation are prohibitive, much of high-speed link design is carried out at the system level with programs like Matlab and Simulink. These tools allow the designer to take a more statistical look at the link behavior.

The impact of various system blocks on signal integrity may even be computed by hand once a mathematical representation of the signal has been derived, assuming a closed-form expression for the response of the transmission channel or specific system block is known. One commonly adopted mathematical approach models a transmitted signal $x(t)$ carrying random data as:

$$x(t) = \sum_{n=-\infty}^{\infty} a_n p_{tx}(t - nT) \tag{3.4}$$

where a_n corresponds to the n^{th} data bit value, $p_{tx}(t)$ represents the pulse response of the transmitter, and T is the symbol period. Physically this equation states that the signal amplitude at any time t will equal the sum of the contributions of all previous and trailing symbols (bit value $\times$ transmitter pulse response), leading up to and including the current symbol, all sampled at time t plus or minus the relative position of the contributing symbol within the bit stream. This is somewhat uninteresting in the transmitted signal wherein the symbols have yet to be spread by the channel and therefore do not contribute much from UI to UI. There are circumstances, however, when the transmitted signal will exhibit ISI. Such is the case when pre or de-emphasis equalization (to be discussed) is applied to the signal in an effort to preemptively counter the degrading effects of the channel.

On the other hand, once the transmitted symbols are distorted by the response of the band-limited channel, it is not uncommon for the preceding and even trailing bits to overlap the bit of interest enough to contribute to the signal amplitude

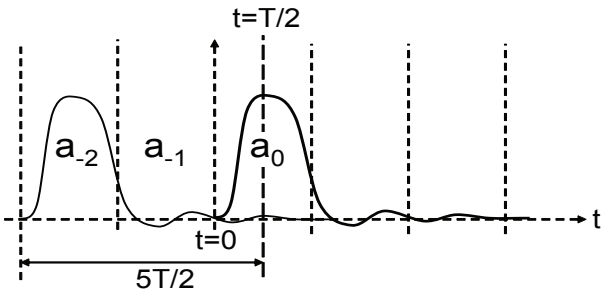


Figure 3.1: Received pulse train illustrating the contribution of symbols a_n to the signal amplitude at time t.

sampled at a specific instant. This ISI is illustrated in Fig. 3.1, which considers the data sequence $a_{-2}, a_{-1}, a_0 = 1,\ 0,\ 1$. In this case, the tail of bit a_{-2} lingers long enough to add to the energy of bit a_0 when the signal is sampled at time $t = T/2$.

Mathematically, what is shown in Fig. 3.1 is understood as follows. First the channel-affected received pulse response $p_{rx}(t)$ is computed through the convolution of the transmitted pulse $p_{tx}(t)$ with the channel impulse response $h(t)$:

$$p_{rx}(t) = p_{tx}(t) \otimes h(t). \tag{3.5}$$

By substituting the received pulse response found in (3.5) for the transmitted pulse response in (3.4), the received bit stream becomes:

$$y(t) = \sum_{n=-\infty}^{\infty} a_n p_{rx}(t - nT). \tag{3.6}$$

Now with reference to Fig. 3.1, and using (3.6), the received signal amplitude at time $t = \mathrm{T}/2$ is found to be:

$$y\left(\frac{T}{2}\right) = 1 \cdot p_{rx}\left(\frac{T}{2} - (-2)T\right) + 0 + 1 \cdot p_{rx}\left(\frac{T}{2} - (0)T\right) \tag{3.7}$$

which simplifies to:

$$y\left(\frac{T}{2}\right) = p_{rx}\left(\frac{5T}{2}\right) + p_{rx}\left(\frac{T}{2}\right). \tag{3.8}$$

The first term in (3.8) represents ISI or the contribution that symbol a_{-2} makes to the overall signal energy at the sampling instant, while the second term corresponds to the symbol of interest, a_0. This approach has been extended into a technique known as peak distortion analysis, through which the worst case eye diagram corresponding to a specific received pulse response may be constructed [11, 53].

Returning to equations (3.4)-(3.6), not only do these expressions provide for the quantification of voltage noise in terms of ISI, but they may also be used to predict the associated DDJ distribution. When the channel response and the transmitted pulse response are both expressible in closed-form, the value of the received signal $y(t)$ may be set equal to the detection threshold, and the threshold crossing instants of the transitioning signal may be found and compared with the ideal crossing times through the following process developed in [39, 40, 41, 42, 43]:

Beginning with (3.6), the difference between the ideal threshold crossing time of the n_{th} transition and the deviation that results due to ISI is found through:

$$\Delta t = -\frac{1}{\frac{dy(t)}{dt}|_{t=t_0}} \cdot \sum_{n\neq 0} a_n p_{rx}(t_0 - nT) \tag{3.9}$$

where t_0 is the ideal crossing instant, the denominator represents the slope or slew rate of the transition, and the summation accounts for the accumulated ISI due to all prior and trailing symbols. It may be noticed that this formula closely follows from the previous discussion on noise-to-jitter translation.

As it is often the case that one particular previous bit a_k will contribute more dominantly to the overall DDJ, it is possible to simplify the analysis further by considering only the kth edge (worst case). In this case, the peak-to-peak DDJ is predicted by:

$$DDJ \approx \left| \frac{p_{rx}(t_0 - kT)}{\frac{dy(t)}{dt}|_{t=t_0}} \right|. \tag{3.10}$$

While ISI and the associated DDJ may dominate the short-term signal degradation, random noise and jitter must also be accounted for. Voltage noise may be added to (3.6) as an independent random variable $\eta(t)$, resulting in an expression of the form:

$$y(t) = \eta(t) + \sum_{n=-\infty}^{\infty} a_n p_{rx}(t - nT). \tag{3.11}$$

As the voltage noise causes fluctuations in the signal at each point in time, the time at which the signal crosses the detection threshold will also vary resulting in a corresponding change in the observed jitter. In addition, this numerical analysis can be extended to include explicit jitter as well, but the correlation between noise and jitter is difficult to account for with these types of expressions. As a result, signals with noise and edges with jitter are often considered independently.

In fact, it is not uncommon for jitter passing through the system to be modeled as a signal itself [53, 54]. Then in accordance with the previously determined jitter transfer characteristics of the various system components, the jitter is filtered, shaped, and accumulated. Sometimes, the correlation between voltage and timing noise is approximated through voltage-to-timing translation parameters, by which the anticipated voltage noise may be scaled through simple multiplication to approximate an associated jitter level. This jitter component is then combined with the other anticipated jitter contributions to predict the total accumulated jitter at the output of the system. While such approximations do provide useful jitter predictions when designing to meet a required jitter budget, they fail to capture much of the non-linear noise-to-jitter translation that occurs in realized circuits, as most approaches

make assumptions regarding the biasing and general performance of the associated transmit and receive circuitry when deriving the corresponding jitter transfer models. Thus, where these techniques fall short is that they fail to account for the combined degradation imposed by simultaneous voltage and timing noise.

Even with the questionable efficiency of standard transistor-level simulation, many of the problems associated with the current modeling techniques could be overcome with the ability to generate input waveforms exhibiting both controllable voltage noise and jitter for transient simulation. While this would not resolve the need for acquiring millions of samples for statistical characterization, it would provide a more accurate understanding of the response of the system components to realistic signal degradation over the short term.

Some third party waveform generators provide a greater degree of flexibility in the signal generation process than what is included with currently popular simulators. Tools such as SynaptiCAD's WaveFormer Pro allow for graphical signal construction, which is basically a visual approach to building up PWL waveforms [52]. The user may begin with either an empty palette and construct arbitrary waveforms from scratch, or they may begin with one of several parameterized signals, and then manipulate the signal's timing and voltage levels to meet their specifications. WaveFormer Pro provides for the injection of jitter onto the edges of periodic clock waveforms, but provides no jitter for aperiodic signals.[3] Once the signals are complete, they may then be imported into simulators like Spice or Verilog in either analog or digital form. When imported into Spice-based tools, periodic clock and aperiodic data signals generated with WaveFormer Pro are mapped to the *VPULSE* and *VPWL* voltage sources respectively.

While the methods employed by WaveFormer Pro to add distortion to waveforms are not readily known, an accepted method for generating jittery signals is illustrated in Fig. 3.2. Derivatives of this method are presented in [55, 56]. Essentially the methodology exploits the noise-to-jitter translation spoken of repeatedly

[3]Synapticad makes no claims in their documentation regarding the model-able jitter characteristics, but only provides for the designation of a jitter "range".

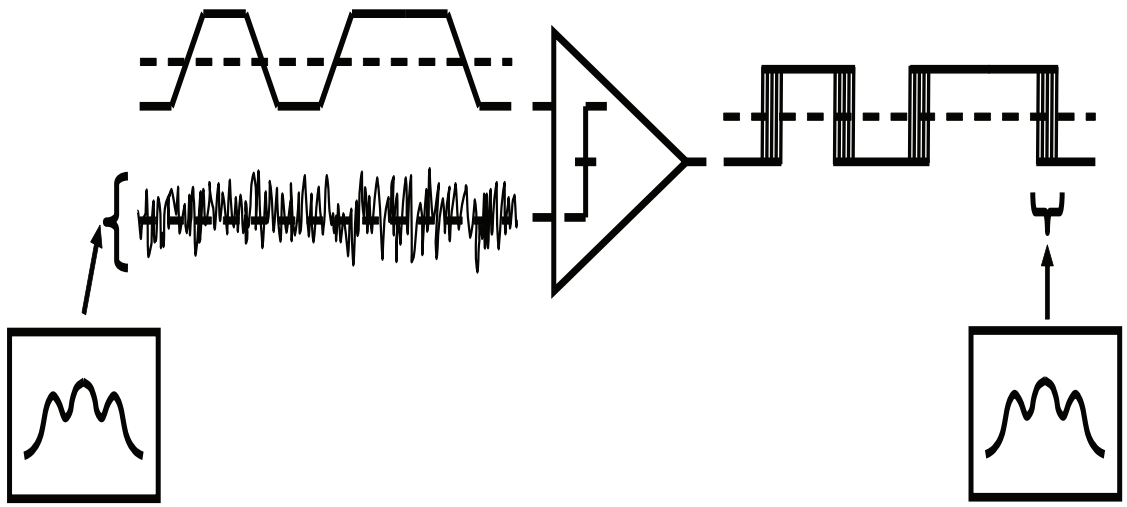


Figure 3.2: A known method for generating signals with jittery edges.

throughout this work. By comparing the required bit stream at the upper left of the figure with noise, the effect is to shift the triggering of the comparison operation in time. Because the comparator outputs are saturated, the signal variance is only evident in the transition timing of the output signal. By constraining the magnitude of the noise, the translation of noise to jitter remains linear and the desired jitter probability distribution can be achieved by imposing the same probability characteristics on the comparison noise. The magnitude of the jitter is scaled through the slewrate of the input bitstream, but herein lies one of the limitations with this approach. The risetime and/or falltime of the input signal may not exceed 1/2 of the bit period, if both a rising and falling edge are to occur within the allotted time, and thus the characteristics or dynamic range of the output jitter is restricted. Because the noise, to which the input bit stream is compared, cannot exhibit true statistical tails, neither can the resulting jitter, a second shortcoming. In fact, in order to approximate the tails of a Gaussian distribution, the rms jitter level at the model's output may be severely limited in magnitude. Another potential problem with this approach is the potential for triggering unwanted pulses. If an instantaneous noise event superimposed onto the signal is large enough to cause glitching, or the repeated crossing

of the comparator threshold during a single transition, then multiple pulses may be generated where only one pulse was desired.

An example of such jitter limitation is found in [57]. Here a bit-error-rate tester (BERT) is used to inject both periodic and random jitter into the test data waveform. At higher frequencies (10-80 MHz) the magnitude of the PJ is limited to 0.5× the symbol period, as expected. Interestingly, at frequencies below 10 MHz, that jitter magnitude is extended to 2.2× the symbol period. But the rms level of the RJ component is always limited to 0.04× the symbol period.

A second approach, less limited in terms of jitter magnitude is described in [58]. In this case, it is proposed that the jittery signal be developed by passing the bitstream through a voltage controlled delay line and introducing jitter by modulating the delay control voltage. This provides a signal free from artificial voltage noise and limited in magnitude only by the timing range of the delay cells.

There are two shortcomings with the approach however. The first is that the control voltage to delay must be linear across a large range in order to accurately reproduce the desired statistical jitter characteristics. And if the jitter injection system is to function at several datarates, then the linear performance must leave margin for both the jitter and the static timing difference associated with the various operating frequencies. Second, it requires the design of a delay line, which will be specific to a particular process node and not trivially ported from one design to the next. A better solution, for the simulation environment, would be independent of circuitry. In the next chapter, new signal models are presented which allow for periodic clock and random data waveforms to be generated with controllable noise and jitter characteristics, while overcoming many of the limitations that have been discussed. In addition, some circumstances exist in which one of the models may be used to calculate circuit-signal interactions with improved simulation time and memory efficiency.

Before moving on, however, two additional methods for analyzing link performance through eye diagram generation are presented, both of which avoid lengthy transient simulation time at the expense of limited flexibility. These two techniques

have come to be known as "Peak Distortion Analysis" [11, 59] and "Statistical Eye Analysis" [11, 60].

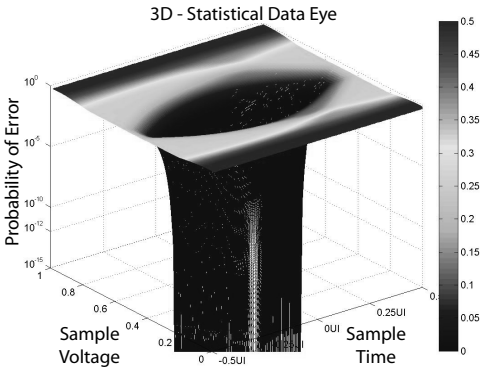
Early demonstrations of peak distortion analysis illustrated how it could be used to find the worst case eye opening and voltage margin from the system pulse response at the ideal sampling instant (pulse peak) [10]. In general, the principle states that when the pulse response is sampled at nUI intervals, then those samples which do not correspond to the sample at the pulse response peak constitute ISI and may be accumulated and subtracted from the value sampled at the peak to represent the potential difference between an ideal *one* and the worst case *one* at that instant. More formally, the process is expressed as:

$$RVD(t) = \left(\frac{p_{rx}(t)}{2} - \sum_{k\neq 0} \frac{|p_{rx}(t-kT)|}{2}\right) \tag{3.12}$$

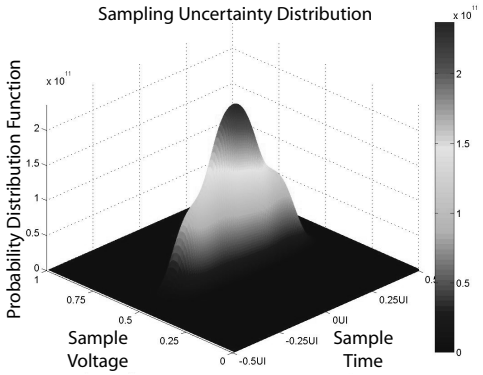
where RVD is the "received voltage difference", $T = 1$UI and k represents bit samples extending from -∞ to +∞. Additional channel noise terms, such as crosstalk and SSO noise may also be accounted for in the RVD calculation when their respective pulse responses are available. By repeating the process at regular time increments t over the pulse response, a set of sample time versus vertical eye opening values are generated, which can then be separated into the worst-case opening corresponding to either a transmitted *one* or *zero*. By superimposing the two resulting curves, the inner eye boundary is derived.

There are a few shortcomings associated with this approach. The first problem is that the generated inner eye boundary is always symmetric about the horizontal axis, which may not be accurate if asymmetries exist in the transmit circuitry. This could, of course, be resolved by using both the rising and falling step responses in the process, though this has not been demonstrated in the literature. A second issue regards the statistical nature of this approach. The probability of encountering the absolute worst case pattern required to close the eye to this degree corresponds to a BER near 10^{-20}, which is pessimistic when the link specification only calls for a BER of 10^{-12}. Still another limitation of this approach is that the magnitude of jitter

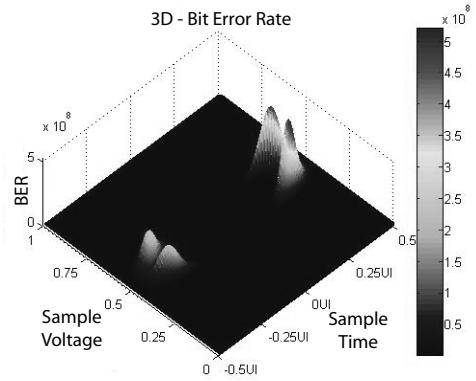
modeled in this way may not exceed 1UI, eliminating the modeling of periodic jitter extending over multiple cycles. Finally, the pulse response from which the worst case eye is derived only corresponds to a specific bias condition of the underlying circuitry. Dynamic changes in supply levels and other noise sources can only be built in through assumptions of how that noise would impact the pulse.



(a) Statistical Data Eye



(b) Sampling PDF



(c) Bit Error Rate

Figure 3.3: Illustration of the BER eye derivation. (a) Probabilistic data eye generated from ISI pdf at 1 ps intervals. (b) Sampling uncertainty distribution generated from the products of independent voltage and timing noise distributions. (c) BER derived from the product of the values from b and c.

To overcome the inherent pessimism of worst case eye generation, a second more flexible method for determining the probability of error based on sampling position within the eye was proposed in [11]. This technique involved the derivation of a statistical eye, from which the BER could be identified for any sample time - sample voltage level combination.

The process for deriving the more descriptive statistical eye diagram also begins with the received pulse response and again steps in time while calculating the voltage characteristics of the eye. But rather than simply calculate the maximum voltage attenuation resulting from interfering components at the same point in time, the method calculates the probability of incorrectly determining the transmitted symbol with respect to all possible reference voltage levels. This is done by calculating the vertical bathtub curve (cumulative distribution function) corresponding to the probability distribution of the symbol interference at each time step. Thus a three-dimensional structure is constructed, as shown in Fig. 3.3a, with the x, y, and z axes corresponding to the sample time, sample voltage level, and an associated probability of error, respectively.

With the probability of incorrect symbol detection calculated for each point in the time-voltage plane, a corresponding probability of each sample point occurrence must also be generated. By multiplying the anticipated sample timing uncertainty distribution with the anticipated reference voltage level uncertainty, a combined three-dimensional sampling uncertainty distribution is built up, as shown in Fig. 3.3b. The product of the probability of error found in Fig. 3.3a and the sampling probability distribution then provides the three-dimensional statistical structure shown in Fig. 3.3c. To calculate the BER of the system, this structure is then integrated along the x and y or time and voltage axes.

Unfortunately, while this method is an important break through in the effort to avoid lengthy time-domain simulations, it still suffers from some of the shortcomings of the worst case eye approach. For example, the method is not compatible with jitter magnitudes in excess of 1UI. It also fails to account for dynamic changes

in the system environment, being based, as was the worst case eye, on a single pulse response captured for a specific system configuration and bias condition.

Chapter 4

Realistic Signal Generation for System Verification

As the previous chapter discussed, the verification of high-speed board-based interconnects is not only constrained by simulation inefficiency, but also by an inability to generate realistic input stimuli for transient simulation. This chapter presents methods for constructing both clock and data waveforms to be used at any level of the simulation hierarchy. The techniques proposed allow for both periodic clock and random data signals to be formed with complete control over both the voltage noise and jitter distributions. In the sections that follow a new set of expressions for generating clock and data waveforms are derived, and several simulations are presented illustrating the precision and flexibility of the proposed techniques.

4.1 Fourier-Based Waveform Generation

The first methodology for generating jittery clock and data signals is an extension of the technique presented in [61] and is based on Fourier theory, which states that any periodic waveform may be represented as a simple DC value combined with an infinite sum of sine-waves and/or cosine-waves at specific harmonic frequencies, as expressed in (4.1). The periodic nature of clock signals makes them well suited to Fourier series representation, while the aperiodic nature of data signals does not lend itself to Fourier series representation directly. Yet as will be shown, this obstacle is overcome in the proposed data waveform generation process.

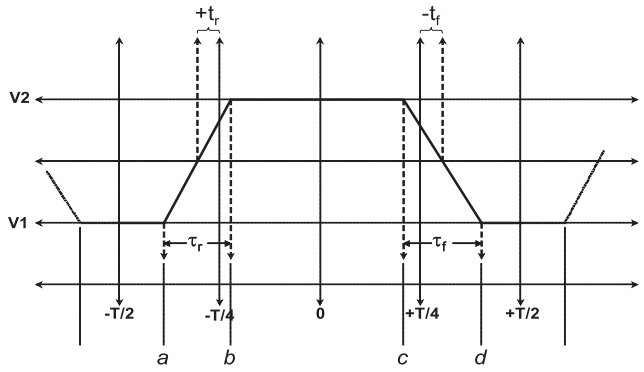


Figure 4.1: Signal model from which the coefficients of the generic Fourier series are derived.

4.1.1 Fourier-based Clock Signal Derivation

The first step in the derivation of the general clock signal Fourier series is to plot out one complete cycle of the periodic waveform to be modeled (see Fig. 4.1). To increase the flexibility of the model and match the degrees of freedom provided in Spice, the following parameters are included:

V_1 = the minimum voltage,

V_2 = the maximum voltage,

T = the period,

τ_r = the risetime,

τ_f = the falltime,

t_r = the rising edge jitter, and

t_f = the falling edge jitter.

To facilitate the Fourier series calculation, the waveform is separated into four segments where boundaries a-d, which will later serve as the limits of integration, are defined to be:

$a = -\frac{T}{4} - \frac{\tau_r}{2} - t_r,$

$b = -\frac{T}{4} + \frac{\tau_r}{2} - t_r,$

$c = +\frac{T}{4} - \frac{\tau_f}{2} - t_f,$ and

$d = +\frac{T}{4} + \frac{\tau_f}{2} - t_f.$

The standard Fourier expression into which the calculated coefficients will be inserted is as follows:

$$C(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi}{T}t\right) + B_n \sin\left(\frac{2n\pi}{T}t\right) \tag{4.1}$$

where

$C(t)$ = the time-domain clock waveform,

t = the timing instant,

T = the signal period, and

n = the integer multiple frequency (harmonic).

The coefficients A_0, A_n, and B_n are found by computing the following integrals:

$$A_0 = \frac{1}{T}\left[\frac{V_2 - V_1}{\tau_r}\int_a^b \left(x + \frac{T}{4} + \frac{\tau_r}{2} + t_r\right) dx \right. \tag{4.2}$$

$$\left. + V_2 \int_b^c dx + \frac{V_1 - V_2}{\tau_f}\int_c^d \left(x - \frac{T}{4} - \frac{\tau_f}{2} + t_f\right) dx\right],$$

$$A_n = \frac{2}{T}\left[\frac{V_2 - V_1}{\tau_r}\int_a^b \left(x + \frac{T}{4} + \frac{\tau_r}{2} + t_r\right)\cos\left(\frac{n\pi x}{T}\right)dx \right. \tag{4.3}$$

$$+V_2\int_b^c \cos\left(\frac{n\pi x}{T}\right)dx$$

$$\left. +\frac{V_1 - V_2}{\tau_f}\int_c^d \left(x - \frac{T}{4} - \frac{\tau_f}{2} + t_f\right)\cos\left(\frac{n\pi x}{T}\right)dx\right], and$$

$$B_n = \frac{2}{T}\left[\int_a^b \frac{V_2 - V_1}{\tau_r}\left(x + \frac{T}{4} + \frac{\tau_r}{2} + t_r\right)\sin\left(\frac{n\pi x}{T}\right)dx \right. \tag{4.4}$$

$$+\int_b^c V_2 \sin\left(\frac{n\pi x}{T}\right)dx$$

$$\left. +\int_c^d \frac{V_1 - V_2}{\tau_f}\left(x - \frac{T}{4} - \frac{\tau_f}{2} + t_f\right)\sin\left(\frac{n\pi x}{T}\right)dx\right].$$

The integrands in the above expressions are simply the set of functions which numerically describe the various segments of the waveform. By initially setting V_1 to zero, all computations corresponding to the regions outside the boundaries a and d are avoided. Any nonzero V_1 value is later added to the computed value of A_0 to account for DC offset.

By substituting A for the full signal swing $(V_2 - V_1)$, the resulting expressions for A_0, A_n, and B_n are:

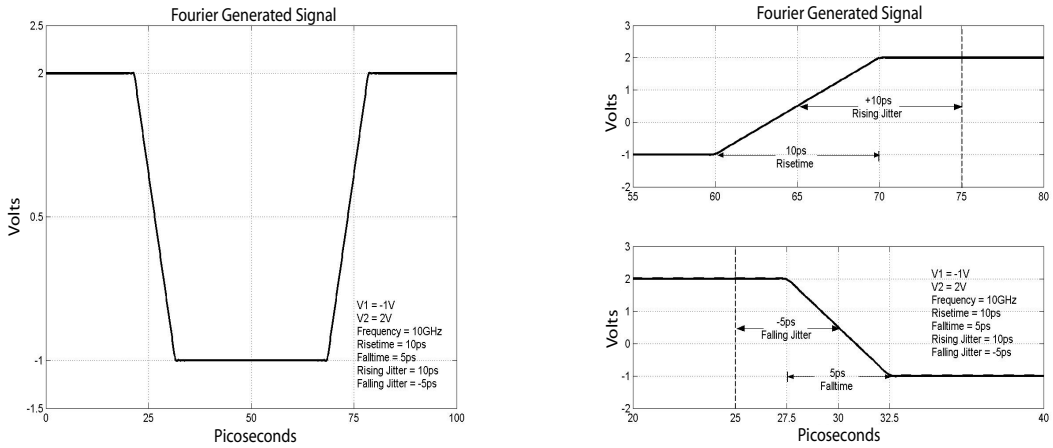
$$A_0 = \frac{A}{2}\left(1 + \frac{2(t_r - t_f)}{T}\right) + V_1, \tag{4.5}$$

$$A_n = \frac{AT}{n^2\pi^2\tau_r}\sin\left(\frac{n\pi\tau_r}{T}\right)\left[\cos\left(\frac{n\pi}{2}\right)\sin\left(\frac{2n\pi t_r}{T}\right) + \sin\left(\frac{n\pi}{2}\right)\cos\left(\frac{2n\pi t_r}{T}\right)\right] \tag{4.6}$$

$$-\frac{AT}{n^2\pi^2\tau_f}\sin\left(\frac{n\pi\tau_f}{T}\right)\left[\cos\left(\frac{n\pi}{2}\right)\sin\left(\frac{2n\pi t_f}{T}\right)-\sin\left(\frac{n\pi}{2}\right)\cos\left(\frac{2n\pi t_f}{T}\right)\right], and$$

$$B_n=\frac{AT}{n^2\pi^2\tau_r}\sin\left(\frac{n\pi\tau_r}{T}\right)\left[\cos\left(\frac{n\pi}{2}\right)\cos\left(\frac{2n\pi t_r}{T}\right)-\sin\left(\frac{n\pi}{2}\right)\sin\left(\frac{2n\pi t_r}{T}\right)\right] \quad (4.7)$$

$$-\frac{AT}{n^2\pi^2\tau_f}\sin\left(\frac{n\pi\tau_f}{T}\right)\left[\sin\left(\frac{n\pi}{2}\right)\sin\left(\frac{2n\pi t_f}{T}\right)+\cos\left(\frac{n\pi}{2}\right)\cos\left(\frac{2n\pi t_f}{T}\right)\right].$$



(a) (b)

Figure 4.2: (a) One cycle of the generated clock waveform. (b) Magnified rising and falling edges of the generated clock.

Once the Fourier coefficients have been computed, a time domain representation of the signal is constructed through the Inverse Fast Fourier Transform (IFFT).

Fig. 4.2a presents one complete cycle of a clock waveform generated with the first 100 harmonics of the Fourier series based on the following parameters:

V_1 = -1 V,

V_2 = 2 V,

f = 10 GHz,

τ_r = 10 ps (risetime),

τ_f = 5 ps (falltime),

t_r = 10 ps (early rising edge jitter), and

t_f = -5 ps (late falling edge jitter).

Fig. 4.2b zooms in on the rising and falling edges of the signal to verify the accuracy of the generated waveform. While the period, minimum and maximum voltages, risetime, and falltime are all easily observed to be correct, the jitter terms require some explanation. In this example, a duty cycle of 50% would result in falling and rising edge crossings at 25 ps and 75 ps respectively. The figure clearly shows the falling edge crossing to occur at 30 ps (5 ps late), corresponding to the desired jitter of -5 ps, while the rising edge crossing occurs at 65 ps (10 ps early), corresponding to the desired jitter of +10 ps.

It should be noted that while the model just derived represents the rising and falling transitions of the signal through ideal linear ramping, any function expressible in closed-form that would more accurately emulate the shape of true signal transitions could be incorporated into the model by replacing the ramps and integrating over the same limits along the time axis. Replacing the ramping edges with more rounded transitions may also lower the number of harmonics required for smooth waveform generation, and as a result, reduce the signal generation time.

Then, based on the current form of the parameterized Fourier series, this underlying signal generation methodology may be employed to either enhance the

efficiency of simulating signal-system interaction, or it may be used to construct signals with unconstrained voltage noise and timing jitter characteristics.

4.1.2 Enhanced Clock Simulation Efficiency

As was discussed in the previous chapter, computing the interaction of a signal with its environment is mathematically carried out either through convolution in the time domain or Fast Convolution in the frequency domain.

Because the Fourier series just derived provides the exact harmonic components of the clock signal, even more efficiency may be obtained during the simulation process. By limiting the number of sinusoidal components of the signal, the frequency representation reduces to a set of scaled delta functions located at the harmonic frequencies (i.e., the Fourier coefficients) as follows:

$$s(t) = A\sin(\alpha t) + B\sin(\beta t) + ... \Leftrightarrow S(\omega) = A\delta(\omega - \alpha) + B\delta(\omega - \beta) + ... \qquad (4.8)$$

where $s(t)$ and $S(\omega)$ are the time and one-sided frequency domain representations of the signal.

This is important because the signal energy at all other frequencies in the spectrum is zero. Many circuit simulators are not equipped to recognize the periodic nature of an incoming signal and often compute the Fourier transform by means of an FFT. Unfortunately, the finite number of points in the signal results in windowing effects during the FFT process, and rather than producing the true frequency response as a set of delta functions, the calculated response will exhibit a noise floor and spikes at the harmonic frequencies. Fig. 4.3 compares the representation of the first 10 harmonic components of a 10 GHz clock signal. In one case, an FFT was computed in PSpice, while in the other case, the component values were taken directly from the Fourier coefficients calculated in the proposed signal generation process. Two important distinctions are illustrated in the figure: first, the spreading at the base of the 10 GHz fundamental on the FFT curve, a manifestation of the finite data

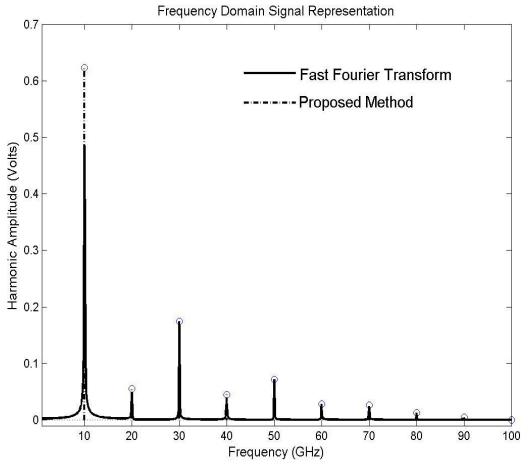


Figure 4.3: Comparison of the signal frequency response taken directly from the Fourier coefficients computed in the proposed signal generation process with those calculated in PSpice through the FFT.

windowing effect, is compensated for by a decrease in the corresponding peak value; second, the existence of nonzero values in between the harmonic frequencies requires a complete point-to-point multiplication of the signal and channel responses during the simulation to avoid a loss of information.

While the number of multiplications associated with direct convolution and the Fast Convolution method scale with the length of the signal and the length of the channel impulse response, the number of multiplications in the proposed method, up until the point of time domain signal reconstruction, is set by the number of harmonics in the Fourier representation of the signal and do not increase for longer signals or more complicated channel frequency responses. In fact, if the transfer function of the channel is known, the magnitudes and phase shifts of the resulting wave's sinusoidal components are found through evaluating the channel frequency response at the various harmonic frequencies and scaling the harmonic magnitudes and phases accordingly.

After the scaling process, the resulting magnitudes and phases may then be used to reconstruct the channel-modified signal. Thus with the exception of modeling

the incoming signal with a truncated Fourier series, the process of computing the signal-channel interaction is carried out through simple steady-state analysis.

If a measured frequency response (e.g., S-parameters) is used rather than a closed form transfer function expression, these formulas only change in the sense that the magnitude and phase multipliers of the channel response will be found by indexing the magnitude and phase data points corresponding to the appropriate frequencies.

Of course, the number of harmonics included in the simulation impacts the accuracy of the result. For the high-speed interconnect environment, the band-limitations of the lossy channel tend to suppress the higher order harmonics, and simulations incorporating 50 harmonic components generally produce excellent matching to the exact signal function.

In comparing the computational demands of this process with convolution-based approaches, as described previously, the new method requires $2k+1$ steps to calculate the initial Fourier coefficients (a single DC component, k sine components, and k cosine components), with k being the number of harmonics included in the Fourier series. This is followed by $2k+1$ steps to calculate the effect of the channel on the magnitudes of the Fourier components and an additional $2k+1$ steps to calculate the associated phase effects. Generating the final time domain representation of the signal from the scaled Fourier coefficients may be done in two ways: first, the various sinusoids may be generated and then summed together, a process which adds an additional $(2k+1)N$ computational steps; or the Fourier coefficients may be fed directly to an IFFT process, adding only $N\log_2 N$ steps. The later method is more efficient when the number of harmonics (k) is greater than $\frac{\log_2 N-1}{2}$, which is generally the case. Combining the number of computations for the entire process leads to:

No. Steps (Proposed Method) =

$$3(2k+1) + N\log_2 N \tag{4.9}$$

where k equals the number of harmonics in the finite Fourier series.

For two signals with 1000 data points each and a Fourier representation including 100 harmonics, the proposed method requires 7,511 steps, compared with 4,670,669,001 and 103,424 for direct and Fast Convolution, respectively.

To verify the positive effect on simulation time, a simulation was constructed in which the impact of a first order lowpass filter on 100, 200, and 300 cycles of a passing clock signal was computed through Fast Convolution and the proposed technique. With the simulation time step fixed at 50 fs, the simulations were completed with Matlab 6.5 running on a 900 MHz Pentium III desktop. The results reported in Table 1 reflect the computational requirements up to the calculation of the frequency domain representation of the filtered signal. The additional $N \log_2 N$ steps incurred by both methods during time domain signal reconstruction were intentionally excluded to better distinguish between the performance and efficiency of the two techniques.

Table 4.1: Simulation Time and Memory Requirements

	No. Cycles	Sim. Time (sec)	Memory (MB)
Proposed Method	100	7	3.2
Fast Convolution	100	7	21.6
Proposed Method	200	7	3.2
Fast Convolution	200	25	49.6
Proposed Method	300	7	3.2
Fast Convolution	300	27	54.4

While the allocated memory reported does not account for memory required by Matlab's internal functions, it does account for all variables and other memory usage accumulated during the simulation. As expected, the required memory and simulation time scaled with the number of signal cycles for the Fast Convolution approach, but no scaling occurred with the newly proposed technique. By way of comparison, the direct convolution method required 20 seconds to simulate only 10 cycles of the same signal.

In addition to providing superior efficiency over longer simulation periods, the benefits of the proposed method are also enhanced with each additional stage through which the signal must pass, assuming linear operation is maintained through each. This is understood by considering that the number of additional computational steps associated with the proposed method is equal to two times the number of stages times the number of signal harmonics (accounting for magnitude and phase), while the increase incurred in the Fast Convolution process is equal to the number of stages times the average number of points in the FFTs of the stages' impulse responses.

Unfortunately, many of the signal degrading components to be modeled are not periodic by nature, and therefore do not lend themselves to the proposed simulation methodology immediately. One such contributor to signal degradation is RJ. Many circuit designers, whose tool set is limited to Spice-like simulators, find that random noise, which in turn produces RJ, may be approximated with a set of sinusoidal signals at independent and unrelated frequencies. As long as the frequencies of the various noise sources do not factor into one another or into the true signal being modeled, then they tend to combine to produce a relatively random waveform. Of course, it is very difficult to control the distribution of the noise and therefore this technique can not be used to produce a perfectly Gaussian distributed noise source. Nevertheless, it may be employed to observe the response of the system under test to nearly Gaussian voltage and timing noise.

Based on the arguments of the previous chapter, even with such a crude approximation, designers must still wait for several cycles to simulate if they desire accuracy in characterizing RJ in the time domain. But if the true signal is being generated from its Fourier components, as proposed, then the many sinusoids constituting the noise source may be handled just as the harmonics of the fundamental signal, being represented as delta functions and consequently shaped by the channel in exactly the same way. At the output of the system, the resulting signal may be reconstructed, including these additional signal components, and all of this can be done while only increasing the number of multiplication steps in the computation to account for the increased number of sinusoids making up the signal.

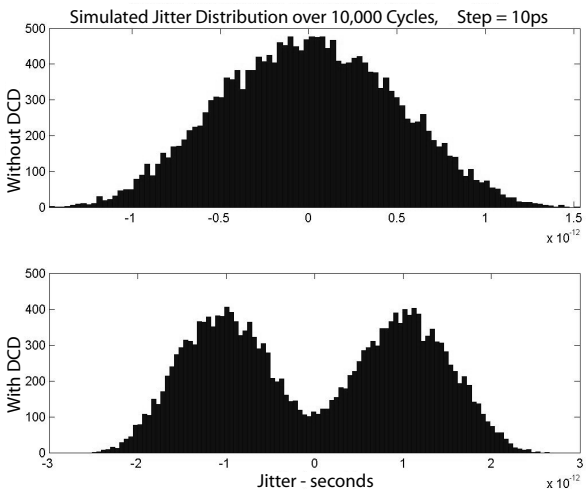


Figure 4.4: Simulation of random and deterministic jitter using the proposed method and a minimum time step of 10 ps. The upper window displays the jitter distribution generated with a set of unrelated sinusoidal noise sources. The lower window adds 0.75 ps of DCD to the total jitter distribution.

To verify this, a set of 6 sinusoids was chosen between the frequencies of 1 GHz and 18 GHz to model a high frequency noise source. Their amplitudes were chosen to provide for approximately 0.4 ps_{rms} of Gaussian-like jitter. Fig. 4.4 presents the simulation results for a pair 10 GHz signals, each with RJ (derived from the six sinusoids) and one with an additional peak-to-peak DCD of 0.75 ps passed through a lowpass filter. The upper window displays the results for the signal with only RJ, while the lower window presents the results for the signal with both RJ and DCD. What makes these results impressive is that they were computed with a minimum transient time step of 10 ps while still providing sub-picosecond jitter resolution. The large time step, while not affecting accuracy, allowed the simulation of 10,000 clock cycles to be computed in approximately 20 seconds, with the majority of that time attributed to the generation and plotting of the histograms.

Matlab's rand() and/or randn() functions may be used to generate sinusoidal noise sources with a controllable standard deviation in amplitude and randomness in frequency without reverting to deriving such a set of signals by hand. This

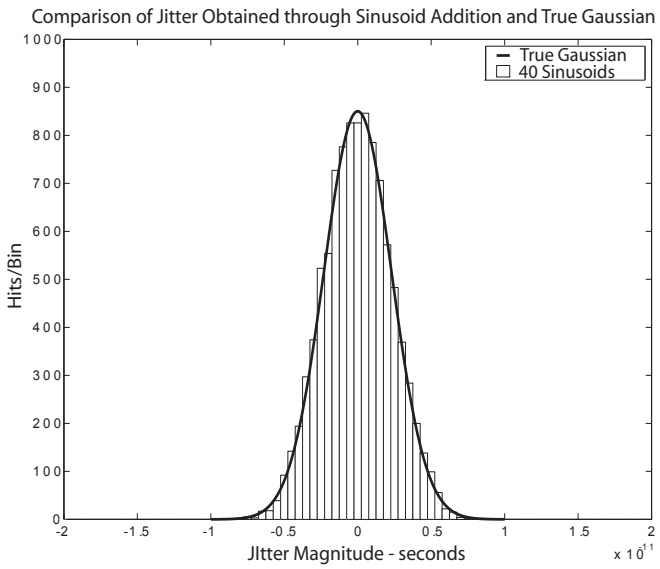


Figure 4.5: Simulated jitter (40 sinusoids) compared with a true Gaussian pdf.

would maintain the periodic nature of all frequency components in the process, allowing the noise to be treated as additional signal harmonics, while still leading to a better approximation of Gaussian white noise.

As an example, a set of 40 sinusoidal noise sources was generated in Matlab spanning the frequency spectrum from DC to nearly 25 GHz (frequencies expected to contribute at the receive end based on the known channel response). Initially the frequencies were determined by specifying that Matlab select the number of desired signals along a logarithmic scaling of the required spectrum. To avoid signal beating, which occurs when two or more signals have a common multiple, the selected frequencies were modified with a random frequency offset generated with the *rand()* function. To obtain nearly Gaussian distributed noise, the amplitudes of the 40 sinusoids were generated with the *randn()* function. Fig. 4.5 compares the resulting simulated jitter histogram and a true Gaussian curve with the same mean and standard deviation.

Even when the proposed simulation methodology of computing the signal-channel interaction through straight multiplication will not suffice, as may be the case when the impact of nonlinear circuit elements on signal degradation of a passing

signal must be accounted for, simulation efficiency of commercially available tools may still benefit when the internal signals provided by the simulator are replaced with signals generated through the proposed technique. This is because the summation of the harmonic components in the time domain leads to timing precision far below the fundamental time step of the waveform but still capturable by the simulator, allowing for faster simulation.

Simulations have been run in which static timing offsets on the order of 1×10^{-23} seconds were generated and visible even when the fundamental time step of the time-voltage vector representing the signal was 50×10^{-15} seconds. In fact, the precision of the waveform timing is only constrained by the numerical limits of the simulator. All of these details imply that a signal generated through this technique may be passed through the commercial simulator with a larger time step, thereby lowering the simulation time and simulator memory requirements.

Before discussing the second advantage made possible through adopting the proposed signal generation technique, the algorithm for efficient computation of signal-system interaction is summarized here for clarity's sake:

1. Based on the estimated channel frequency response, select the number of signal harmonics to carry through the computation.

2. Using equations (4.5) - (4.7), calculate the magnitudes and frequencies of the Fourier components of the desired waveform.

3. Periodic noise such as DCD may be added and modified through a variation in the underlying Fourier series.

4. Quasi-random noise may be added through an additional set of sinusoids at carefully chosen, unrelated frequencies.

5. Scale the sinusoids of the Fourier series and noise by the magnitude of the channel transfer function evaluated at the corresponding harmonic frequencies.

6. Shift the sinusoids of the Fourier series by the phase angle of the channel transfer function evaluated at the corresponding frequencies.

7. Reconstruct the signal from the resulting set of Fourier components.

4.1.3 Unconstrained Waveform Generation

The second application facilitated by the proposed signal generation technique is the derivation of periodic and aperiodic signals with unconstrained control over voltage and timing characteristics.

Because the IFFT returns only one time domain cycle for a given set of Fourier coefficients, rather than copy that one cycle over and over to produce a purely periodic waveform, allowing for the enhanced simulation efficiency just discussed, several different cycles may be generated and pieced together to provide a more realistic transmitted signal. For example, the transition terms t_r and t_f from equations (4.5) - (4.7) may be considered as random variables and a new set of Fourier coefficients may be calculated for each cycle, implying that both deterministic and random jitter may be completely controlled during the waveform generation. When implemented in Matlab, the formation of 100 - 1000 jittery clock cycles takes 3-4 seconds, though the generation time grows in proportion to the product of the number of cycles and the number of harmonics incorporated into each cycle.

4.1.4 Fourier-based Data Signal Generation

In the same way that several cycles with varied timing parameters at each edge may be pieced together to form a jittery clock signal, several distinct data symbols may be pieced together to produce a random data waveform, though the aperiodicity of random data adds complexity to the derivation. Rather than computing the Fourier series for a single periodic waveform, each transition and binary state of the desired signal requires a separate symbol to insure continuity at each cycle edge.

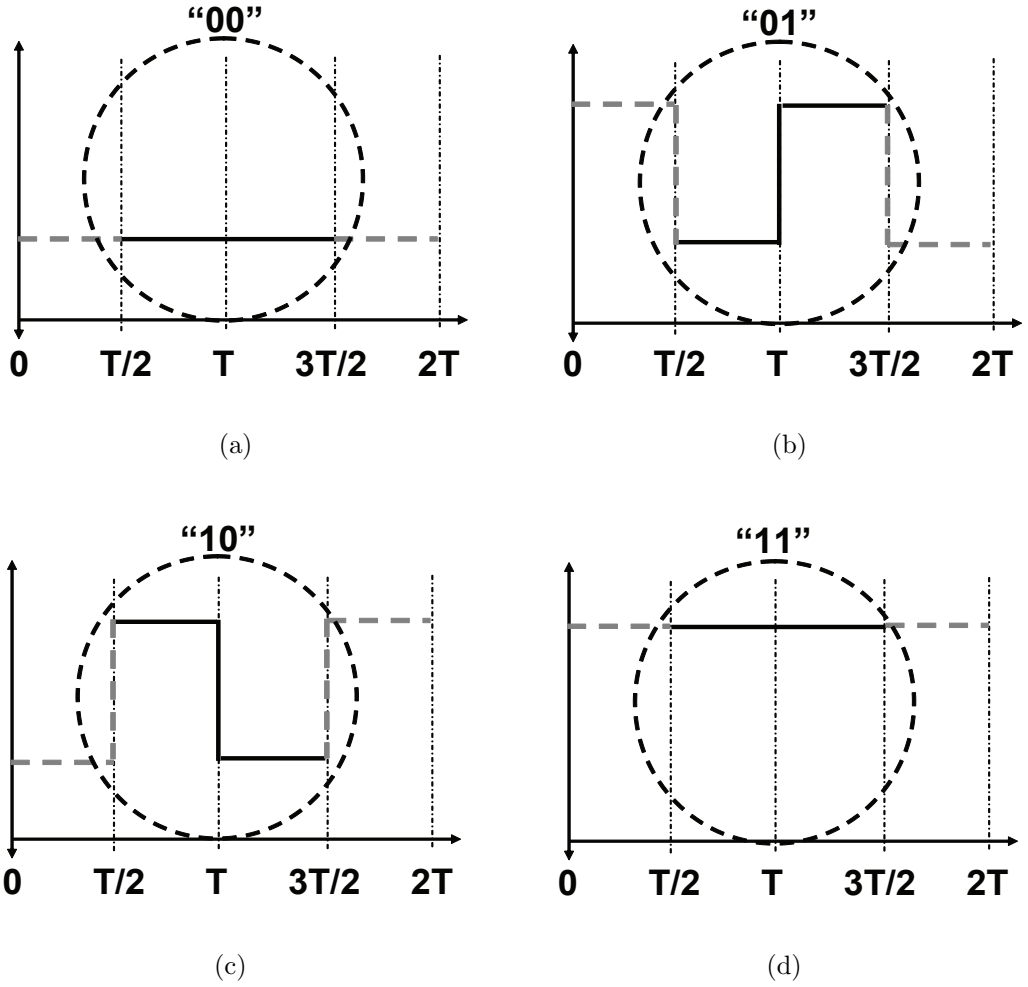


Figure 4.6: Four data symbols used to represent binary NRZ signaling.

Fig. 4.6 presents the four symbols needed to generate binary NRZ data:

- "00" implies two consecutive transmitted zeros;
- "01" implies a zero followed by a one;
- "10" implies a one followed by a zero; and
- "11" implies two consecutive transmitted ones.

The Fourier series derivation is then very similar to what was carried out for the clock signal. While the "00" and "11" symbols are trivial, expressions for the parameterized Fourier coefficients of the "01" and "10" symbols must be computed. To overcome the non-periodic nature of each data symbol, the Fourier series is calculated assuming periodicity. Because the IFFT process returns one cycle at a time, the data waveform may be pieced together not only with specific edge information defined for each cycle, as was the case with the clock, but the symbol itself may also change from cycle to cycle. Thus, once the desired bit stream has been encoded with these four symbols, the data signal is pieced together without discontinuity.

Those familiar with Fourier series may recognize a flaw in this approach. Because the IFFT returns only one cycle at a time, and because the Fourier series is designed to repeat after each cycle, the end points of the "01" and "10" symbols shown in Fig. 4.6 will tend to bend inward toward the vertical midpoint. To overcome this, the algorithm is modified slightly by cutting the user-defined frequency in half to spread out the symbol in time. When the time domain representation is returned by the IFFT, intentionally constructed with twice as many time points, the symbol is truncated symmetrically about the horizontal midpoint down to the desired period length, thereby eliminating the unwanted tail curving at the endpoints and guaranteeing continuity between successive symbols.

4.1.5 Signal Generation Summary

From the discussion presented in the preceding sections, a summary of the proposed method for generating realistic clock and data signals with controllable noise and jitter characteristics is as follows:

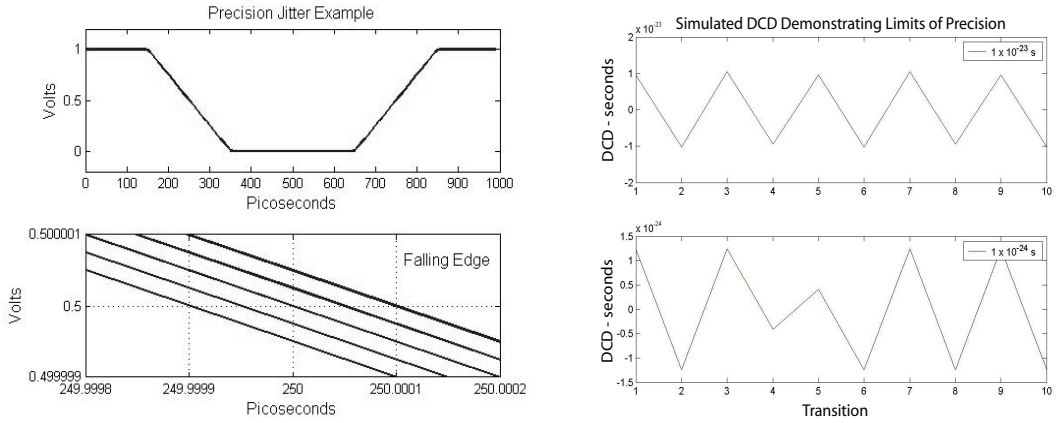
1. Based on the desired jitter distribution, generate a vector of timing values representing the jitter at each sequential edge.

2. Based on the estimated channel frequency response or system bandwidth, select the number of signal harmonics to carry through the computation.

3. If generating a clock, use equations (4.1) - (4.7) to calculate the magnitudes and frequencies of the Fourier components of each cycle, convert to the time domain through the IFFT, and piece the cycles together.

4. If generating data, encode the bit stream using the four symbols shown in Fig. 4.6 and then use a set of similar equations to calculate the magnitudes and frequencies of the Fourier components of each cycle, convert to the time domain through the IFFT, and piece the cycles together.

5. Based on the desired voltage noise distribution, generate a vector of voltage values representing the noise at each time step and add these noise values to either the clock or data signals at the corresponding step in time.



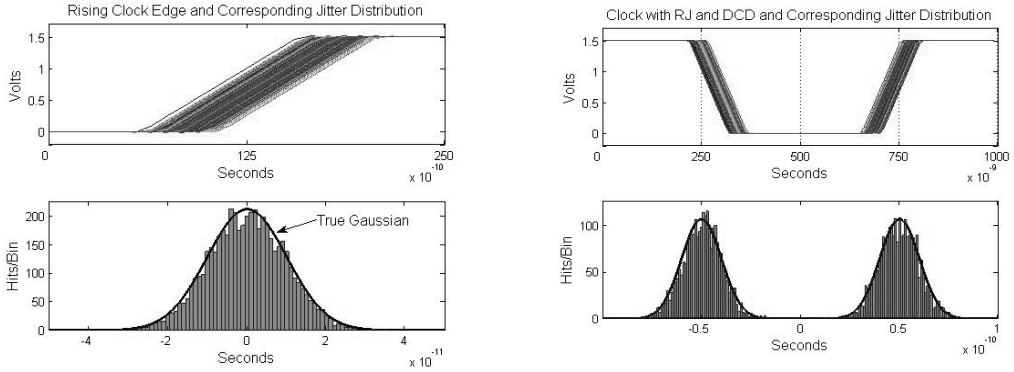
(a) Ramping Jitter

(b) DCD

Figure 4.7: Demonstration of the time-domain precision of the proposed waveform generation. (a) The upper window shows a 1 GHz clock waveform generated through the proposed method. The lower window presents an incremental jitter of 0.5 fs generated with a time step of 10 ps. (b) Demonstration of DCD successfully simulated down to 1×10^{-23} with a time step of 50 fs.

4.1.6 Verification

To verify the precision of the proposed method, in terms of jitter resolution and distribution approximation, several Matlab simulations were completed. In Fig. 4.7a, five 10 GHz clock cycles were simulated with specified jitter (each cycle constructed from 50 harmonic components). For edges 1-5, the designated jitter magnitudes were -1E-15, -0.5E-15, 0, 0.5E-15, and 1E-15 seconds. The upper window of the figure presents the superposition of the five clock cycles. By zooming in on the falling edge it is not only possible to distinguish the five edges, but it is observed that the edges cross the midway point with the designated timing. It is important to note that the underlying code plots these signals out with 100 time steps per cycle. In other words, the simulation demonstrates a resolution of better than 0.5E-15 seconds with a simulated time step of 1E-11 seconds. Similarly, Fig. 4.7b demonstrates the successful simulation of DCD down to 1×10^{-23} with time step of 5E-14 seconds, more than nine orders of magnitude larger.



(a) Random Jitter

(b) Random Jitter + DCD

Figure 4.8: Comparison of generated jitter and theoretical jitter distributions. (a) Rising clock edge exhibiting RJ. (b) Clock exhibiting both RJ and DCD.

Fig. 4.8a, Fig. 4.8b, and Fig. 4.9 demonstrate the ability of the proposed technique to approximate specific jitter distributions. In the first case (Fig. 4.8a), a signal was generated to exhibit only Gaussian distributed jitter. After producing the signal, the edge timing was extracted and binned in the histogram shown. A true Gaussian curve was then overlaid for comparison. In the second case (Fig. 4.8b), the RJ was combined with 50 ps of DCD and the true distribution was again superimposed. Fig. 4.9 displays the jitter distribution extracted from a signal generated with a single sinusoidal jitter component.

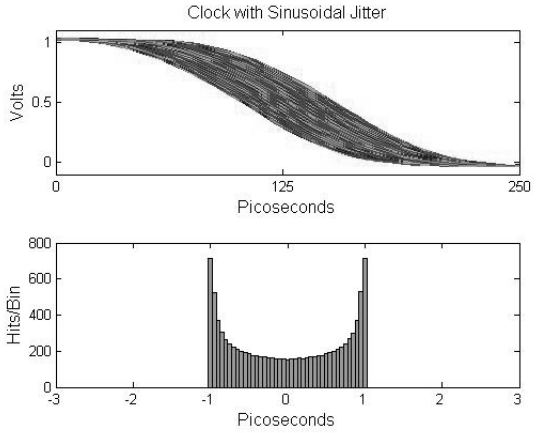


Figure 4.9: Clock jitter distribution indicating the presence of sinusoidal jitter.

4.2 Jitter Injection

To inject jitter into an existing signal with both time and voltage dimensions is complicated. The proposed method for executing this operation is to:

1. Measure the mean "0" and "1" values of the existing signal.

2. Measure the mean risetime and falltime of the existing signal.

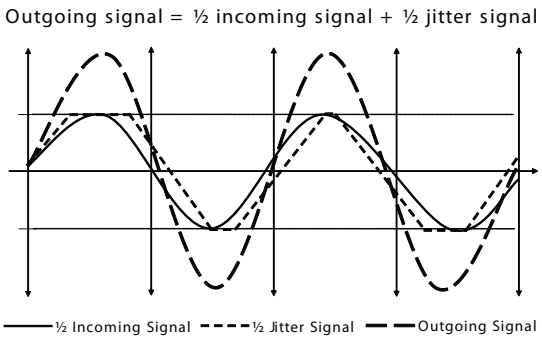


Figure 4.10: Method for injecting jitter into an existing signal.

3. Extract the transition timing of the existing signal. This might be done by interpolating when the signal crosses a designated threshold.

4. Derive a second signal whose voltage swing was determined in step one, whose risetime and falltime were found in step two, whose initial phase is in sync with the mean phase of the original jittery signal, and whose jitter characteristics represent the additive jitter.

5. Scale the two signals by a factor of 1/2 and then use vector addition to combine them. Steps one and two minimize the reshaping of the original signal during this averaging process.

This procedure is illustrated in Fig. 4.10. As can be seen, at some edges the injected jitter adds constructively to the total timing deviation of the transition, while at other edges it may reduce the final jitter value.

4.2.1 Additional Applications

While the target use of the proposed waveform generation techniques is to build up input stimulus with controllable jitter for transient simulation, the ability to alter the waveform's characteristics on a cycle to cycle basis also facilitates other forms of circuit and system characterization. A short list of additional possible applications is provided here:

1. Not only does the ability to simulate combinatorial jitter (e.g., RJ + DCD + Sinusoidal) provide an additional degree of freedom in signal modeling, but it also proves useful in the characterization of core communication circuits, such as PLLs and DLLs. Typical analysis of PLL and DLL control loops consists of stepping the frequency of the input signal and then observing the convergence of the control signal as the PLL/DLL locks to the new frequency through the phase comparison process. This technique provides information regarding the overshoot, settling time, and stability of the control loop.

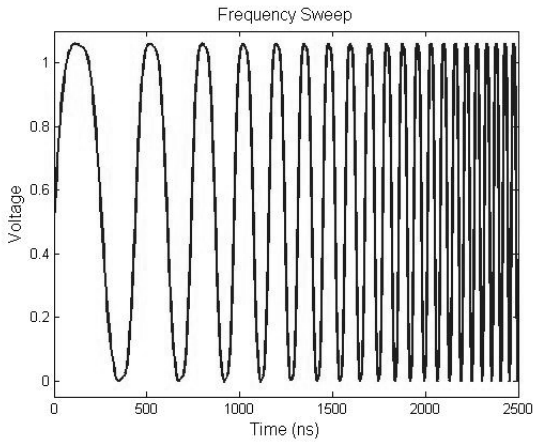


Figure 4.11: Signal derived from Fourier components while the frequency is modulated from 1 MHz to 20 MHz.

In a realized system, however, the phase offset between the input signal and the reference signal will vary continuously as both the oscillator and the delay line contribute additional jitter to the equation. In the case of the PLL's VCO, small drifts in the free-running oscillator frequency manifest themselves as jitter in the reference voltage, which tends to accumulate and pass to the output until the control signal offers compensation, resulting in a phenomena referred to as jitter peaking. By employing a voltage-controlled-delay-line (VCDL) rather than a VCO, the DLL avoids significant jitter accumulation, but still displays

a moderate translation of power supply noise to output jitter as a result of delay element sensitivity. Thus to truly model the phase-locking and tracking ability of PLLs and DLLs requires the application of signals with continuously varying phase, potentially including sinusoidal variation, at the circuit input. Only in this way can the true jitter transfer and jitter peaking of the circuits be characterized.

Along these same lines, to provide an even more accurate representation of PLL behavior requires an accounting for the oscillator frequency drift. Simple modulation of the reference signal's frequency is accomplished by specifying the period of each cycle during the initial signal generation process. Fig. 4.11 presents a frequency modulated signal derived from Fourier components, where the cycle-to-cycle frequency follows a linear ramp, but the change in frequency could be allowed to vary randomly or in accordance with understood oscillator phase noise behavior [50].

2. The jitter control may also be used to study the setup-and-hold time of latching circuits. This procedure often entails comparing two periodic waveforms, one of which is assumed to trigger the capture of the other. The timing of one of the waveforms is then slipped or delayed incrementally while observing the output of the comparator. When the transitions of the two waveforms occur close enough in time, the output of the comparator will behave erratically. The window over which this behavior occurs is the minimum setup-and-hold time required for proper comparator operation. Constructing such waveforms by hand is time consuming, and so the temptation is to sacrifice timing resolution in terms of the slipping increment, in order to speed up the simulation setup time. But through the techniques presented, the ramping of the edge timing only requires that the input jitter vector represent a linear ramp, and the resolution of the time-slip can be made arbitrarily small.

3. The ability to adjust the common-mode level of the waveforms can be exploited in input common-mode range simulations. One approach to characterizing input

common-mode range is to ramp the common-mode level of both the reference voltage and the signal to which it is being compared. At the extreme common mode levels, the comparison operation will fail. By adjusting the swing of the signal being compared, the sensitivity of the input common-mode range to signal swing may also be observed. Using the proposed techniques, it is trivial to generate a pair of waveforms that track each other while ramping in their respective common-mode levels, a task that would again be time consuming if done by hand.

4. Finally, the input sensitivity of comparator circuits can be studied by incrementally decreasing the swing of the input signal while observing the comparator's output for the point at which the output fails to resolve to the correct level. This is again a trivial operation as the high and low voltages of the signals generated through the proposed approach can be designated at the beginning of each new cycle.

4.2.2 Limitations

There are two main limitations associated with the proposed signal generation techniques. The first is related to the time needed for waveform generation. In order to avoid Gibbs phenomenon, or signal ringing in the presence of fast edges, the number of harmonics must be increased. As a new set of harmonics is computed for each cycle, the computation time increases at a rate proportional to $N(2k+1)$ where N is the number of cycles and k is the number of harmonics. So while 1000 cycles of the waveform may be generated in a matter of a few seconds, 10,000 cycles could require up to a minute, and so on. To a degree, the computation time is also hindered by the required memory allocation needed to store the harmonics associated with each cycle.

The second limitation, which applies only to the data waveform, is that to maintain continuity from symbol to symbol requires that the peak-to-peak jitter does not exceed 1/2 of the bit period.

4.3 Alternative Signal Generation Algorithms

To overcome these limitations, a pair of alternative algorithms, presented here, take a more straightforward approach to the signal generation process, while insuring a computation time approximately proportional to N.

The clock generation is carried out as follows:

1. Build a vector v, equal in length to the required number of cycles, where each indexed point is assigned a value of zero.

2. Build a second vector t of equal length, whose values range from zero to the value $\frac{cycles-1}{datarate}$ in $\frac{1}{datarate}$ increments. This second vector represents the locations of the ideal edges.

3. Add the desired jitter sequence directly to the vector defined in the previous step. The jitter sequence is in the form of a signal itself, and may take on any realizable distribution.

4. Upsample both vectors by 3 in order to insert two empty place holders between existing values.

5. Starting with index i=1 of vector v, every third index point is assigned the value midway between the low and high voltages of the waveform (V1 and V2).

6. Starting with index i=2 of vector v, every sixth index point is assigned the value of V1.

7. Starting with index i=3 of vector v, every sixth index point is assigned the value of V1.

8. Starting with index i=5 of vector v, every sixth index point is assigned the value of V2.

9. Starting with index i=6 of vector v, every sixth index point is assigned the value of V2.

10. Starting with index i=2 of vector t, every sixth index point is assigned the value of t(i-1) + falltime/2.

11. Starting with index i=3 of vector t, every sixth index point is assigned the value of t(i+1) - risetime/2.

12. Starting with index i=5 of vector t, every sixth index point is assigned the value of t(i-1) + risetime/2.

13. Starting with index i=6 of vector t, every sixth index point is assigned the value of t(i+1) - falltime/2.

14. The final signal voltage and timing vectors are found by resampling the vectors v and t at with the desired timestep, while computing the associated signal levels through a "nearest neighbor" interpolation algorithm.

The algorithm for generating a random data waveform is similarly as follows:

1. Build a vector v, equal in length to the required number of cycles, where each indexed point is assigned a value of zero.

2. Build a second vector t of equal length, whose values range from zero to the value $\frac{cycles-1}{datarate}$ in $\frac{1}{datarate}$ increments. This second vector represents the locations of the ideal edges.

3. Add the desired jitter sequence directly to the vector defined in the previous step. The jitter sequence is in the form of a signal itself, and may take on any realizable distribution.

4. Upsample both vectors by 3 in order to insert two empty place holders between existing values.

5. Build a third vector r, equal in length to the required number of cycles, where each indexed point is randomly assigned a binary *zero* or *one* value.

6. Starting with index i=1 of vector v, every third index point is assigned the value midway between the low and high voltages of the waveform (V1 and V2).

7. Starting with index i=2 of vector v, every third index point is assigned the values contained in vector r.

8. Starting with index i=3 of vector v, every third index point is assigned the values contained in vector r.

9. Starting with index i=4 of vector v, remove unwanted transitions at every third index point by comparing the values found at index points i-1 and i+1. If these values are equal, $v(i)$ is assigned value found in $v(i-1)$.

10. Starting with index i=2 of vector t, every sixth index point is assigned the value of t(i-1) + falltime/2.

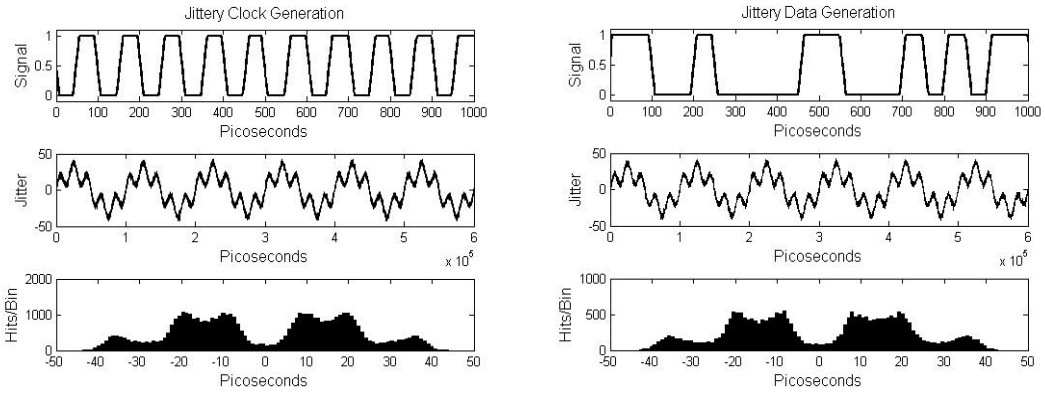
11. Starting with index i=3 of vector t, every sixth index point is assigned the value of t(i+1) - risetime/2.

12. Starting with index i=5 of vector t, every sixth index point is assigned the value of t(i-1) + risetime/2.

13. Starting with index i=6 of vector t, every sixth index point is assigned the value of t(i+1) - falltime/2.

14. The final signal voltage and timing vectors are found by resampling the vectors v and t at with the desired timestep, while computing the associated signal levels through a "nearest neighbor" interpolation algorithm.

The capability of the proposed algorithms are illustrated in Fig. 4.12, where Fig. 4.12a corresponds to a jittery clock signal and Fig. 4.12b corresponds to a random bit stream. Using the algorithms just presented, both clock and data signals are constructed to exhibit the following jitter characteristics: 2 ps_{rms} Gaussian distributed jitter plus two sinusoidal jitter components at frequencies of 10 MHz and 50 MHz with magnitudes of 50 ps and 25 ps, respectively. In each case, the upper windows present several cycles of the waveform in the time-domain, the middle windows present the jitter (in picoseconds) extracted at each signal transition, and the lower windows present histograms of the extracted jitter pdfs.



(a) Clock Signal

(b) Data Signal

Figure 4.12: Periodic clock and random data signals exhibiting both random jitter and sinusoidal jitter components as generated by the proposed algorithm with associated time-domain extracted jitter and associated histograms. (a) Jittery clock signal. (b) Jittery random data signal.

These alternative algorithms are significantly faster, and unlimited in terms of jitter magnitude in both clock and data waveforms, with the constraint that the signal edge timing must increase monotonically to maintain causality. In addition, because the mathematics of this second pair of algorithms do not require the FFT or IFFT, implementation of this form of signal generation in other, less computationally friendly, programming languages is more feasible. This does not negate the value of the preceding techniques, however, as a degree of flexibility is lost with the new methods. For example, while jitter is designate-able on a cycle to cycle basis, all other waveform parameters remain fixed, implying that with the exception of setup-and-hold simulation, the new algorithms are not compatible with the remaining circuit characterization processes previously listed. In addition, this alternative approach requires the entire signal to be built up at once, while the previous methodology allowed for the designation of signal characteristics on a cycle-to-cycle basis, implying that it can be placed into models where the signal may be controlled or manipulated over time (e.g., by the control loop of a PLL).

Chapter 5

Mitigating Noise and Distortion in the Channel

The preceding chapters, with the exception of Chapter 4, laid out the fundamental problem statement of this thesis, namely that high-speed chip-to-chip communication is restricted by the deterministic noise or distortion and jitter associated with the physics of the PC board channel, the random noise and jitter generated from within the I/O circuits themselves, all coupled with an inability to simulate system performance with the requisite level of realism. In Chapter 4, methods for enhancing link verification through realistic jittery signal generation were presented. This chapter explores methods for reducing degradation due to noise and distortion, and specifically considers the impact of matched filtering, transmit pulse shaping, and channel equalization on the performance of the link.

To discuss the evolution of signal conditioning techniques, it is helpful to separate all known methods into two main categories: attenuation of random noise through filtering and minimization of signal distortion through pulse shaping and/or equalization. Interestingly, a similar separation between techniques specifically targeting clock integrity and those aimed at data conditioning can also be made due to spectral distinctions between the two types of signals. The narrowband nature of clock signals makes them somewhat immune to the distortion associated with the channel (e.g. nonuniform group delay, frequency dependent attenuation, etc.). On the other hand, clocking signals are sensitive to SSO noise, crosstalk, and other forms of uncorrelated noise, particularly when they are not isolated from noisy data lines. And while clocking signals are not degraded by ISI, slewrate and amplitude degradation at high frequencies does tend to magnify the impact of uncorrelated noise.

Conversely, the broadband nature of random data makes these signals very susceptible to channel distortion, including ISI and DDJ. Data signals are also sensitive to uncorrelated noise, but as will be shown, the prohibitive cost of simultaneously addressing both noise and distortion in data signals typically results in the high-level design choice to address only the more dominant short term problem of distortion in practice. This does not mean that the study of reducing uncorrelated noise in data signals has been forsaken. On the contrary, methods to reduce SSO noise and crosstalk are regularly published [14, 15, 62], but power/area limited products, when choosing between minimizing random noise and pulse distortion, typically adopt ISI targeting channel equalization.

5.1 Filtering Noise

As mentioned repeatedly throughout this work, random noise not only closes the received data eye in the vertical direction, but also, as the result of noise-to-jitter translation through the signal slewrate, contributes to the horizontal eye closure as well. But before timing uncertainty was ever considered problematic, the need to suppress amplitudinal noise in communication systems had motivated signal processing research for decades.

In the late 1940s, the problem of estimating signals in the presence of random noise experienced a breakthrough as the result of Wiener's work and subsequent publication on what is now referred to as the *Optimal Wiener Filter* [63]. One interpretation of the Wiener filter operation is that it identifies a signal's frequency content and in turn only provides amplification at those frequencies, thus avoiding the simultaneous amplification of noise. Unfortunately, the mathematical techniques proposed by Wiener can be difficult to implement, and the Wiener-Hopf equations, which produce the impulse response of the "optimal" filter, are often unrealizable in hardware. As a result, it is not uncommon for only suboptimal approximations of the true Wiener filter to be feasible.

Follow-up work by Kalman overcame several of the difficulties inherent in the Wiener filter through the use of conditional distributions and expectations. By

redefining the problem in terms of states and state transitions, the Kalman filter employs feedback to approach the Wiener filter from a *Controls* point of view, and in doing so reaches a more readily implementable solution without the mathematical complexity [64].

5.1.1 Matched Filtering

In the digital communication systems considered here, however, it is not necessary to retrieve or rebuild the signal as it was originally transmitted. While the received signals appear very analog in character, the only requirement of the link is to identify the intended binary value of each received bit. Thus, the problem of signal conditioning across the high-speed interconnect is more a question of detection rather than estimation. As a result, a more appropriate solution for mitigating uncorrelated noise in digital transmission channels is through matched filtering [10, 37]. Interestingly, the output of the matched filter may not look at all like the transmitted signal (i.e., the output of a filter matched to a square pulse produces a triangle), but rather exaggerates the differences between the transmitted symbols, thereby making it easier to distinguish between the symbols themselves, and between the symbols as a group of deterministic waveforms from the surrounding noise.

By definition, the impulse response of the matched filter is the time-reversed, conjugate of the transmitted pulse, as illustrated in Fig. 5.1. To insure causality, the impulse response of the matched filter also includes some time delay. Mathematically, it can be shown that the convolution of a transmitted symbol with the impulse response of the associated matched filter maximizes the SNR [10, 37]. Intuitively, this is understood by considering the convolution operation that takes place as the signal passes through the filter. Due to the well defined relationship between the transmitted symbol and the matched filter, the convolution of the received symbol with the filter impulse response actually computes the cross-correlation of the noisy received symbol with the ideal symbol, a process which tends to average out randomness. In [65], the process was described in the following way.

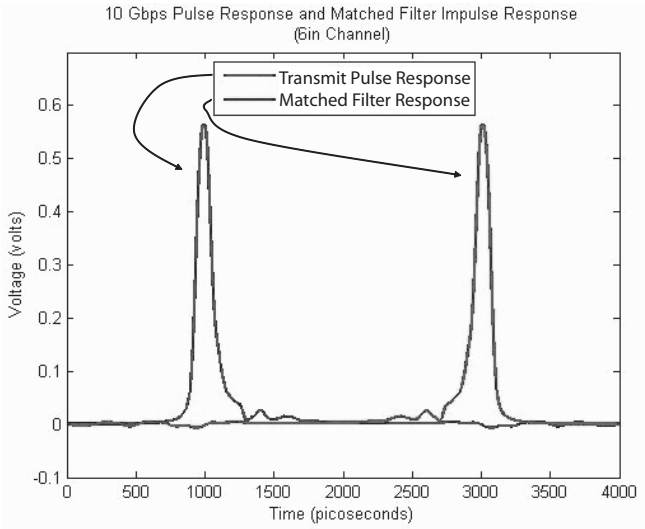


Figure 5.1: The six inch channel - 10 Gb/s pulse response and corresponding, artificially delayed, matched-filter impulse response.

When a transmitted pulse is represented by $h_p(t)$ for $0 \leq t \leq T$, the impulse response of the matched filter over the same range (assuming real signals) will take the form:

$$h_m(t) = h_p(T - t), \tag{5.1}$$

and for a symbol $s(t)$ passing through the matched filter, the output waveform $y(t)$ is computed through the convolution integral:

$$y(t) = \int_T^0 s(t)h_m(T - t)dt, \tag{5.2}$$

and by replacing $h_m(t)$ in (5.2) with expression (5.1), the convolution takes the form:

$$y(t) = \int_0^T s(t)h_p(T - (T - t))dt, \tag{5.3}$$

which may be reduced to:

$$y(t) = \int_0^T s(t)h_p(t)dt, \tag{5.4}$$

which is simply the cross-correlation of the received symbol with the ideal transmitted symbol. Consequently the output of the matched filter grows as the symbol enters the filter and peaks at the instant in time when the noisy symbol most closely "matches" or resembles the ideal symbol. It is this integration of the incoming symbol energy that increases the SNR. While at any one moment, the instantaneous noise magnitude may be greater than that of the signal, when averaged or integrated over time, zero-mean random noise tends to cancel while the symbol energy continues to accumulate.

For the NRZ data transmission, typical in high-speed wireline interconnects, this operation is often implemented through the "integrate and dump" process, in which the symbol energy accumulates through integration, and following the sampling, which ideally occurs when the cross-correlation between the transmitted pulse and the filter response is maximized, the accumulated energy is eliminated as quickly as possible in preparation for the next symbol. Inherent in this approach is a sensitivity to sampling uncertainty. If the sampling clock exhibits jitter, it becomes impossible to guarantee sampling at the optimal point, and as a result, SNR is no longer maximized by the process.

The claim that matched filters optimize SNR also assumes that the noise to be removed is Gaussian or at least uncorrelated with the data, and it is the orthogonality of Gaussian noise with most signals of interest that enables the matched filtering technique to be so effective [10, 37, 66]. On the other hand, the pattern and channel dependence of ISI implies that it is not orthogonal to the underlying signal and therefore the matched filter is not expected to suppress it effectively. In fact, depending on the severity of the ISI, matched filtering has the potential to initially magnify the degradation. Fig. 5.2 verifies this by comparing a few cycles from a simulated 10 Gb/s random bit sequence after traversing a severely band-limited channel and consequently passing through either an ideal matched filter or the second order continuous-time equalizer discussed in the next chapter. As expected, the equalizer,

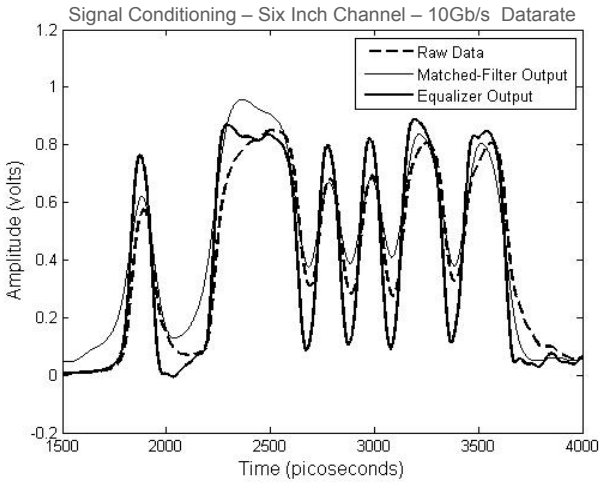


Figure 5.2: Comparison of raw, match-filtered , and equalized 10 Gb/s data at the receiving end of a six inch FR4 PC board channel.

which is designed to extend the bandwidth of the transmission channel, provides a clear improvement over the raw received data. But it is also interesting to compare the matched filter output with the raw data. Here it is observed that the matched filter output tends to be even slower to respond to data transitions, due to the integration process, and as a result, increases the ISI and the corresponding DDJ.

This comes as no surprise, as the frequency response of matched filters in this type of application must be lowpass to coincide with the band-limited transmit pulse response. One solution might be to combine matched filtering with alternative techniques to be discussed shortly, but in a cost sensitive design, area, power, and complexity constraints force a choice to be made between the competing conditioning circuits in practice. Thus, the decision to incorporate matched filtering hangs on which source of degradation most limits the link performance: uncorrelated noise or distortion in the form of ISI. As will also be discussed in the next chapter, the noise floor of the typical PC board channel may be as low as -100 dB, though additional uncorrelated noise may be coupled to the signal from other sources. At the same time, the ISI alone resulting from the dispersive effects and frequency dependent attenuation produced by even channels of modest length, is enough to close the received eye

completely during multi-Gb/s operation. Thus, focus is consistently placed on the reduction of signal distortion, and matched filters are rarely found in high-speed chip-to-chip interconnects.

5.2 Minimizing Distortion

The two most commonly employed methods for countering the pulse distortion imposed on the signal by the channel are transmit pulse shaping and channel equalization.

5.2.1 Transmit Pulse Shaping

While channel equalization is perhaps the more common solution, it is possible to minimize ISI without equalizing the channel, and this may be accomplished with or without matched filtering [67, 68, 69, 70, 71, 72]. In fact, it is well known that a set of pulse shapes exist, often referred to as the "generalized Nyquist pulses," that ensure the received symbols will not interfere in a degrading manner, even after crossing the unequalized channel [10, 37, 70, 71, 72]. At a high level, pulse shaping is understood to reduce the signal energy at frequencies most effected by the channel, and hence the pulse distortion incurred across the channel is minimized.

Unfortunately, to realize the maximum benefits of pulse shaping techniques, two specifications which are difficult to achieve in high-speed environments, must be met. First the claim of ISI free transmission assumes ideal mid-point sampling of the received symbols, and thus intolerance to sampling clock jitter, much like the matched filter. And as has been discussed, clock jitter is a growing problem in high-speed links. While methods, such as the bandpass filter presented in Chapter 7, exist for suppressing clock jitter, total jitter elimination is an impossibility.

The second requirement is that the circuits used to implement the pulse shaping are realizable. In [73, 74, 75, 76], methods for analog pulse shaping circuit realization are proposed. Unfortunately, cutting edge CMOS technology does not produce the transistors needed to implement elegant pulse shaping at the requisite frequencies. In fact, most transmitters struggle to simply drive the load of the channel

itself and rely on brute force to generate enough pulse energy to reach the far end of the channel. The only capability vaguely resembling pulse shaping in current systems is the inclusion of slewrate control, produced by sequentially turning on parallel output stages of the driver. Thus, as was the case with matched filtering, pulse shaping could be incorporated by reducing the target bandwidth of the link, but as bandwidth is the over-riding goal, this has yet to happen in the multi-Gb/s regime.

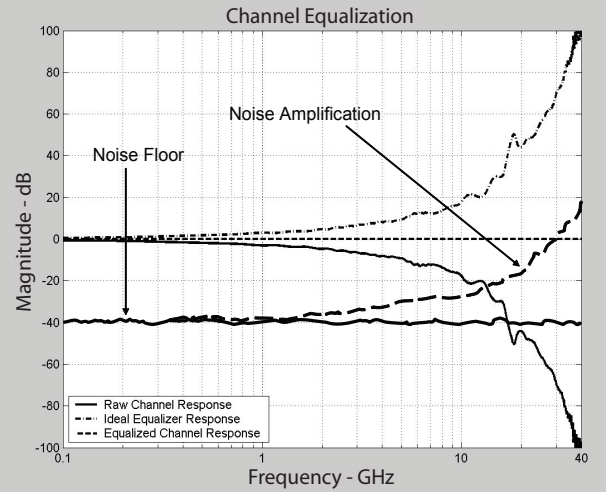


Figure 5.3: Illustration of the basic channel equalization concept.

5.2.2 Channel Equalization

Having concluded that the benefits of matched filtering will be minimal due to the low noise floor of the PB board channel, and that the maximal benefits provided by pulse shaping are unattainable without relaxing the link bandwidth requirement, the next alternative to consider is the method of channel equalization. The goal of channel equalization, as depicted in Fig. 5.3, is to compensate for high frequency signal loss incurred across the band-limited channel. This is typically accomplished by realizing or approximating the inverse of the channel frequency response with some form of variable gain amplification and/or filtering. Thus, by either preceding

or following the channel with the equalization circuitry, the signal degrading effects of the channel are canceled, leaving a "flat" channel-equalizer response. For the example shown, the insertion loss of a six inch copper channel in FR4 produces the lowpass transfer characteristic tracked by the solid downward curve, the dash-dot upward curve represents the equalizer transfer function needed to compensate for the channel loss, and the straight dashed line at 0 dB represents the ideal equalized channel response. The fictitious white noise floor and subsequent equalized noise spectrum are also shown to illustrate the potential problem of high frequency noise amplification. As will be discussed, the level of noise amplification is dependent on both the equalizer topology and the equalizer coefficient tuning algorithm.

Channel equalization may be implemented at either end of the channel, and the trade-offs between transmit and receive-side equalization are well known. The main advantages of transmit equalization are their ease of implementation and their relative effectiveness with respect to receive side counterparts of the same complexity. Transmit equalization may be incorporated into the pipeline prior to the serialization process, allowing it to be carried out at lower frequencies, which in turn increases achievable precision and possibly decreases power consumption. At the same time, mitigating signal degradation prior to transmission minimizes sensitivity to the noise exaggerating effects of the channel. For example, jitter injected into the signal prior to transmission, through noisy PLL-triggered serialization circuits, is of great concern in that it modulates the transmitted pulse width (and possibly height) and has the tendency to reduce the received data eye in both the voltage and timing dimensions, whereas jitter injected through the components of the receiver typically only impact the width of the data eye opening [54, 59]. In fact, the impact of transmit jitter may warrant additional resources in addition to channel equalization, which typically does not reduce random noise components. These additional efforts most often take the form of more carefully designed transmit clocking. For example, to achieve 20 Gb/s communication, three independent projects sacrificed on-chip area and clock tunability to incorporate LC-based low phase noise PLLs in the transmitter [27, 28, 77], thus reducing the injected clock jitter during the serialization process.

There are two main drawbacks associated with transmit equalization. First, because the signal prior to transmission is likely at or near CMOS levels, there is not much room for the boosting of any portion of the pulse. Rather, transmit equalization usually consists of de-emphasizing certain characteristics of the transmitted pulse. Second, in order to calibrate the response of the transmit equalizer to compensate for the channel requires some form of feedback to identify how changes in the equalizer affect the quality of the received signal. That feedback most often is sent back over a dedicated transmission channel, requiring at least one additional pin for each bus. To avoid the added pin and routing cost, one alternative is to time-multiplex the feedback information onto the link being adapted, but this limits link bandwidth when data must be held up to allow for the feedback transmission. A second approach, which only applies to differential links, is to feedback equalizer update information on a common-mode backchannel [49, 78]. It is the theoretical immunity of the differential forward link to common-mode noise that makes this possible. By generating feedback in the form of common-mode level shifting, information regarding the equalizer performance may be transmitted back across the same channel without interfering with the forward-going data transmission. In single-ended links, however, common-mode signal variation is considered eye-closing noise. And because the cost of an additional feedback pin and channel is too expensive, transmitter equalization, if implemented at all, usually takes the "fixed by design" approach, in that the design is based on the anticipated response of the channel and is fixed, thereby requiring no feedback.

On the other hand, because signals arriving at the receiver are attenuated by the channel, the allowable gain of receive-side equalizers is only limited by the capabilities of the underlying circuit components. At the same time, receive-side equalizers are more easily adaptable as they experience the signal degradation and require no feedback from the opposite end of the channel. Rather, they rely on the minimization of some error metric, generated from within the receiver, to tune the response of the equalizer. This error term may be the difference between measured received data and the known true data values previously stored in static memory for

the purpose of I/O circuit training, or it may simply be the difference between the input and output of the decision operation within the equalizer. What distinguishes one tuning algorithm from the next is the way in which it uses the error term to manipulate the equalizer transfer function.

One tuning method, often referred to as zero-forcing is designed to force all ISI terms to zero. While relatively simple to implement, this approach has the potential to degrade the system SNR. This is understood by referring back to Fig. 5.3, wherein the white noise floor was spectrally shaped by the highpass characteristics of the equalizer. The equalization shown in the figure is an example of the zero-forcing algorithm, which results in the flat channel-equalizer bandwidth over the frequency range of interest. Not only does zero-forcing have the potential to amplify high frequency white noise, but it will amplify any noise corresponding in frequency to dips or nulls in the channel spectrum. The channel shown in Fig. 5.3, though lossy at high frequencies, would be considered relatively well behaved over most of its frequency response, indicating that the number of discontinuities in the channel (e.g. connectors, vias, and other impedance variations) have been minimized. If one or two connectors were added to the transmission path, then a corresponding null in the channel frequency response would likely be observed. To compensate for the signal attenuation produced by the null, the zero-forcing algorithm would adjust the equalizer coefficients in such a way as to produce a peak in the equalizer frequency response over the corresponding null frequency, and this may turn problematic if the SNR is degraded through the resulting amplification of noise over the frequency band of the null.

The minimum mean squared error (MMSE) algorithm avoids these problems by only seeking to minimize the mean-squared error of the residual ISI terms. While the error metric may be derived identically for zero-forcing and MMSE implementations, the MMSE equalizer tends to produce a smoother composite spectrum, less affected by nulls. As will be shown, however, the low noise floor of the PC board channel may lead to significant similarities between the optimal zero-forcing and optimal MMSE equalizers.

In addition to the method of adaptation, the frequency of adaptation may also affect system performance. Specifically, the question of whether or not a single-pass calibration is sufficient for long periods of operation must be answered. In [79] a comprehensive study of the effects of design (channel routing, board materials, etc.), manufacturing (etching, etc.), and environmental (temperature, humidity, etc.) variance on channel performance was carried out leading to the following conclusions that:

1. Channel sensitivity to manufacturing and environmental variations increased with operating frequency, and hence channel equalizers must be adaptable at higher frequencies to compensate for higher levels of variation in the channel behavior.

2. Without continual equalizer adaptation, the BER of the link under consideration was observed to degrade from 10^{-12} to 10^{-4} depending on the temperature, a parameter likely to change over time.

3. Channel performance was degraded more severely by environmental variance than by manufacturing variance, and thus limiting adaptation to a single pass designed to tune out manufacturing variance, leads to suboptimal performance.

In a similar study, [80], it was determined that while at least one round of "set and forget" coefficient tuning significantly improves link performance over a "fixed by design" approach, on-going coefficient adaptation leads to even more performance enhancement. Thus, the recent trend has been to design these circuits in such a way as to dynamically adapt the equalizer response to counter or reverse the undesirable time-varying characteristics of the channel [81], but the challenges associated with such cycle to cycle calibration grow with the operating frequency, and less frequent retuning may soon be the best alternative.

While the need for channel equalization was noticed at least as far back as Morse during his work with the telegraph [82], efforts to mitigate bandwidth limitations in electrical communication can be traced back to the 1920s and 1930s, when

several patent applications were filed with the United States Patent and Trademark Office disclosing a variety of channel equalizer topologies [83, 84]. The earliest equalizer topologies were inherently continuous-time, commonly employing passive filtering techniques to compensate for the high frequency loss of the targeted channel. Discrete-time equalization, such as finite impulse response (FIR) filters [24, 25, 26, 85] and decision feedback equalization (DFE) [30, 31, 32, 86, 87, 88] came later, with first evidence being the transversal filter proposed in a 1953 paper from MIT [89]. As the architecture and underlying theory of discrete-time and continuous-time equalizers are distinct, they will be covered separately.

5.2.3 Discrete-Time Equalization

Perhaps the simplest discrete-time equalizer is represented by the block diagram in Fig. 5.4. In the z-domain, the corresponding transfer function may be expressed as:

$$H(z) = 1 - z^{-1} \tag{5.5}$$

which is translated to the s-domain through the substitution $z = e^{sT}$ resulting in:

$$H(s) = 1 - e^{-sT}. \tag{5.6}$$

To quantify the filtering behavior of this function, the Fourier transform of the s-domain expression is computed and found to be:

$$H(\omega) = 2 - 2\cos\left(\frac{\pi}{T}\right) \tag{5.7}$$

which is a highpass function over frequencies ranging from $0 \rightarrow \frac{\pi}{T}$ seconds.

Intuitively, when an ISI-degraded signal fails to breach a given detection threshold, as shown earlier, there may still be a measurable difference between the present and past samples. This is captured through the process of differentiation, as implemented by the "delay and subtract" architecture shown in the figure.

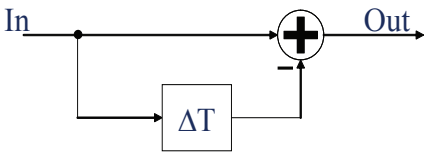


Figure 5.4: "Delay and Subtract" discrete-time channel equalizer, which differentiates the passing signal, identifying signal transitions.

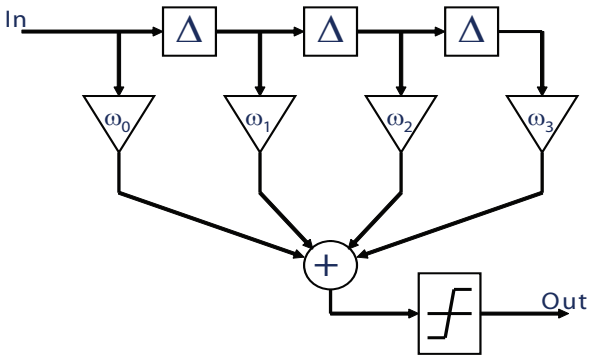
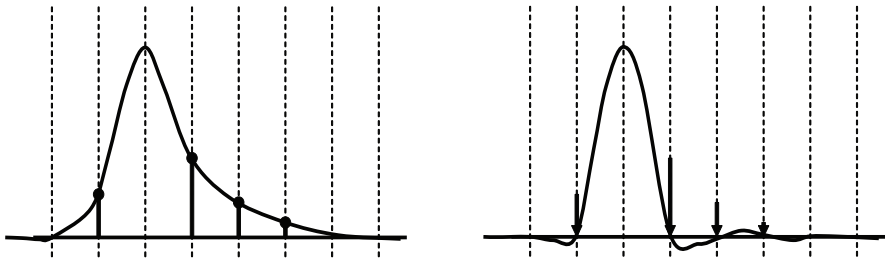


Figure 5.5: Block diagram of a 4-tap finite impulse response or transversal filter.

The common feature of discrete-time equalization topologies is that they employ regularly sampled values of the signal to be filtered, or the post-filter decision, to shape the signal prior to the decision point. In the case of the FIR filter, sometimes referred to as the feed-forward or transversal equalizer, shown in Fig. 5.5, the incoming signal is sampled or tapped at symbol-spaced intervals. Those sampled values are then weighted while passing through a set of independently controlled variable gain amplifiers. And finally the tapped values and original signal are recombined through the summing node at the input of the decision device (comparator). By weighting the taps appropriately, any residual pulse response (ISI) occurring at symbol-spaced intervals from the pulse peak or cursor, may be zeroed out, as illustrated in Fig. 5.6.

Adaptive tuning of transversal filter coefficients for discrete-time channel equalization was first proposed in 1965 [24]. This initial proposal implemented the



(a) (b)

Figure 5.6: Effect of discrete-time equalization on degraded pulse response. (a) Unequalized. (b) Equalized.

zero-forcing algorithm discussed previously, while a later proposal presented in [85], implemented MMSE adaptation.

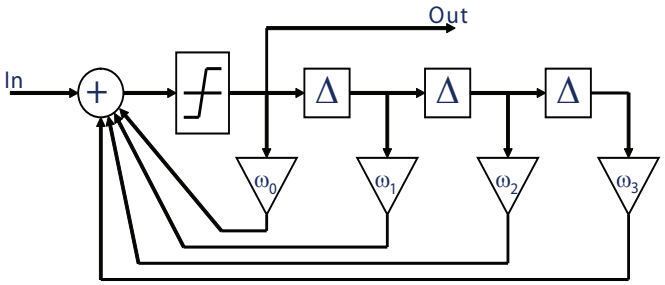


Figure 5.7: Block diagram of a 4-tap decision feedback equalizer.

Within the following year, a revolutionary equalizer topology was proposed in the form of the decision-feedback equalizer (DFE), as illustrated by the block diagram in Fig. 5.7 [86]. By using a linear combination of past decisions (noiseless values if the decisions were correct) to reshape the signal, the DFE compensates for the band-limited channel response while minimizing, and often eliminating, the amplification of noise inherent in corresponding linear equalization techniques, while

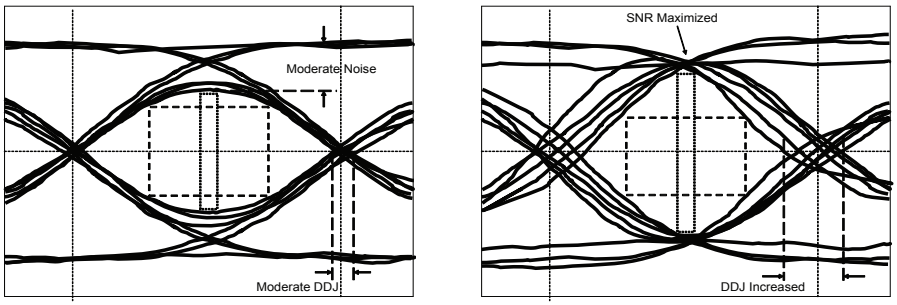
providing greater immunity to sampling phase noise [31]. Because the DFE inherently only addresses post-cursor ISI, it is often combined with feed-forward equalization for more comprehensive signal conditioning.

At high frequencies, FIR filters are more commonly employed, as their forward path topology makes them better suited for high-speed applications. DFEs, on the other hand, are more difficult to implement in the multi-GHz frequency range due to their reliance on feedback from past decisions, though techniques such as coefficient look-up tables [30] and loop-unrolling [32, 90, 91, 92] have proven to increase DFE throughput. It is the nonlinear functionality of the DFE, with its avoidance of high frequency noise amplification, that keeps it in competition with the inherently faster FIR-based topologies.

While the calibration or adaptation of discrete-time equalizers has enjoyed decades of refinement, there are still some drawbacks to employing such topologies when addressing both voltage and timing degradation simultaneously. To begin, discrete-time filters, whether they be FIR-based or DFE-based, are designed to reduce or remove ISI at a particular sampling instant enforcing no constraint on the signal condition at adjacent timing instants within the available sampling interval in a way similar to the matched filter. Thus a weakness of discrete-time equalizers is their inherent sensitivity to sampling variance or clock jitter.

Fig. 5.8a and Fig. 5.8b illustrate the sensitivity of discrete-time equalizers to sample timing uncertainty. Two sets of rectangles are placed within the unequalized (left) and equalized (right) eye openings. The narrow rectangles correspond to minimal sampling jitter, while the wider rectangles represent increased sample timing uncertainty. The fact that the narrow rectangle is taller in the equalized case is evidence that for small timing uncertainty, the SNR is improved through discrete-time equalization. On the other hand, the wider rectangle is taller in the unequalized eye, implying that the equalization may actually degrade receiver voltage margin, were the sampling jitter to increase.

To reduce the sensitivity of discrete-time equalizers to sampling jitter, fractionally spaced equalizers were introduced. As the name implies, fractionally spaced



(a) (b)

Figure 5.8: Eye diagrams used to illustrate the simultaneous impact of discrete-time equalization on SNR and jitter, and the sensitivity of discrete-time equalized signals to sampling uncertainty. (a) Unequalized (b) Equalized.

equalizers operate on the signal not once per symbol but at multiple points in time within the detection interval. The result is to smooth the equalized signal around the optimal sampling instant, thereby providing a greater level of tolerance to timing deviation in the sampling mechanism [93].

5.2.4 Continuous-Time Equalization

For decades, the fundamental FIR and DFE architectures dominated the area of channel equalization due to their tunability and relative simplicity, and the fact that digital signals, until recently, could be treated as purely digital. Yet parallel research led to the maturity of continuous-time equalization techniques, whether in the form of passive filtering, g_m-C filtering or more sophisticated methods [94, 95, 96]. The challenge of implementing channel equalization through continuous-time, analog circuits is in their limited tunability. In addition to the requirement of correlating with a specific channel response, analog-based equalizers must also be tuned simply to cancel out the high level of variability in the integrated passive and active components. The inherent lack of tunability in continuous-time analog equalizers often leads to designs which are fixed, in terms of their circuit parameters, with an associated hope

that careful design and layout will minimize process variations and the need for tuning. Even today, a commodity 6.4 Gb/s equalizer is available constructed completely from passive, non-adaptable components, with the exception of a CMOS level restoring limiting amplifier at the output [97].

Yet even though continuous-time equalizer adaptation is more challenging, there have been several successful designs. In [29] and [30], the continuous-time equalizer shown in Fig. 5.9 is proposed to work in conjunction with a DFE in a magnetic dispersive channel. The continuous-time equalizer replaces the more standard FIR filter in addressing pre-cursor ISI and is shown to perform better than a five-tap FIR filter over a certain range of channel dispersivity.

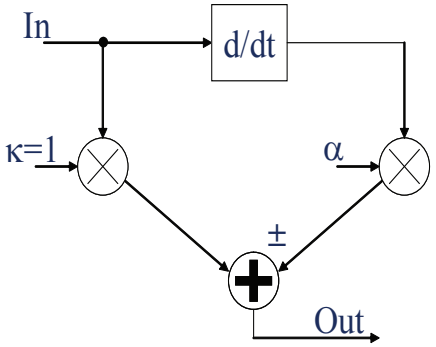


Figure 5.9: The continuous-time magnetic read channel equalizer.

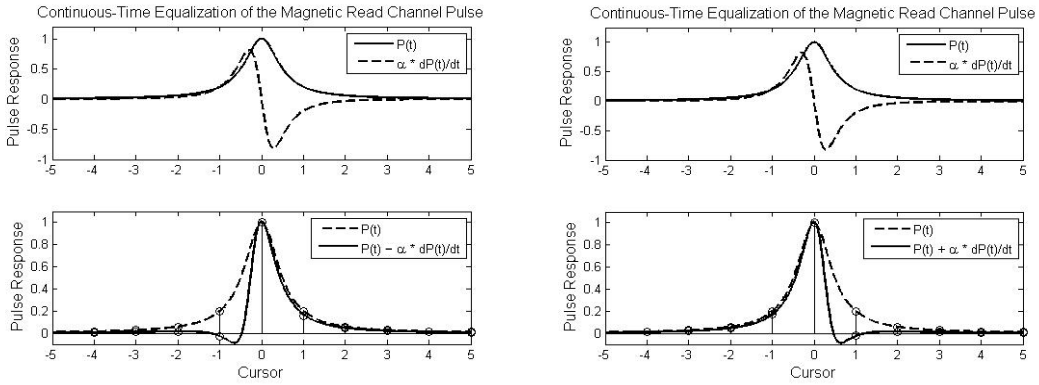
The differential equation describing this particular architecture is:

$$y(t) = x(t) \pm \alpha \frac{dx(t)}{dt} \tag{5.8}$$

where x(t) and y(t) represent the input and output of the circuit and α is a weighting factor by which the derivative of the input is scaled before being summed with the true input. Using s-domain analysis, the corresponding transfer function is found to

be:

$$H(s) = \frac{Y(s)}{X(s)} = 1 \pm \alpha s. \tag{5.9}$$



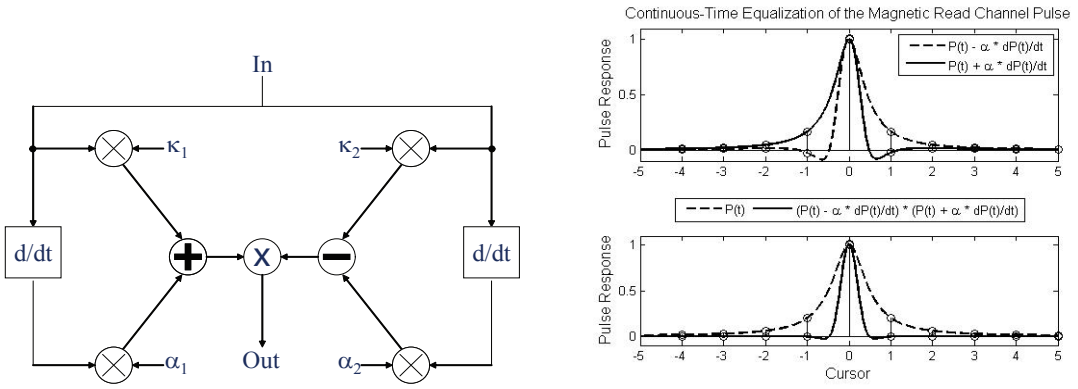
(a) $1 - \alpha \frac{d}{dt}$ (b) $1 + \alpha \frac{d}{dt}$

Figure 5.10: Application of the $1 \pm \alpha \frac{d}{dt}$ equalizer to the magnetic read channel pulse. (a) Pre-cursor Equalization. (b) Post-cursor Equalization.

From the frequency domain perspective, the transfer function represents a zero at the frequency $\frac{1}{\alpha}$ rad/s, or in other words a high pass filter. For the specific characteristics of the typical magnetic read channel, this simple transfer function may be very effective in reducing either the pre or post-cursor ISI, but never both. This is possible because the derivative of the magnetic read pulse response follows the true (Lorentzian) pulse response very closely over one half cycle and then inverts over the second half, as illustrated in the upper windows of Fig. 5.10a and 5.10b. The lower windows of the same figures show the pre-cursor (left) and the post-cursor (right) ISI completely eliminated through the equalization process.

As a side note, a seemingly obvious enhancement to this topology is to combine the achievable pre and post-cursor cancellation by passing the incoming signal through both formats in parallel and taking the product of the two outputs,

as demonstrated in Fig. 5.11a. Then as shown in Fig. 5.11b, the resulting pulse response is free from both pre and post-cursor ISI simultaneously without the need for DFE post-cursor cancellation. Unfortunately, it is the specific characteristics of the magnetic read pulse that allows for such comprehensive ISI cancellation, while direct application of this form of equalization to PC board channels was seen in simulation to be ineffective.



(a) Enhanced Read Channel Equalizer

(b) Equalized Pulse Response

Figure 5.11: (a) Enhanced magnetic read channel equalizer topology for canceling both pre and post-cursor ISI. (b) Application of the pre/post cursor equalizer to the magnetic read channel pulse.

A second example of successful continuous-time equalizer design is reported in [98], where Cherry-Hooper amplifiers implemented in a SiGe process are used to improve the uniformity of group delay in optical channels. The quality factor (Q) of the second order amplifiers was adjustable, and used to minimize the group delay variation, thereby reducing ISI significantly. In [99], an adaptive cable equalizer was used to enable 400Mb/s communication. The transfer function of the equalizer consisted of three frequency zeros, each providing +20 dB/Decade rise in the equalizer frequency response. The position of the zeros was controlled through an RC network

and adapted by comparing extracted high frequency signal content at the input and output of the equalizer, and subsequently tuning the filter to provide the requisite level of high frequency boost.

Another example is the continuous-time graphic equalizer proposed in [100]. In this design, several bandpass filters, with offset center frequencies, were placed in parallel, with their outputs summed together. The center frequencies and Qs of the respective filters were designed to span the frequency range of the passing data. By controlling the contribution or gain of each filter individually, the high frequency peaking needed to compensate for the channel loss was achieved without the need for frequency tuning of the filters.

Two of the most recently proposed continuous-time equalization methods use ISI monitoring circuitry to direct the coefficient adaptation process [101, 102]. In one case, the equalizer approximated a pair of independently tunable frequency zeros, which could be combined to compensate for as much as +20 dB of loss at 10 Gb/s [101]. The cross-correlation between past decisions and an error term generated the gradient of the adaptation. The second design included a 5-tap transmit equalizer as well as a tunable second order continuous-time receive-side equalizer providing a combined compensation of up to 35 dB at 6.4 Gb/s [102]. In this approach, the logic and other supporting circuitry required to perform the adaptation was large enough to require a second chip just for calculation purposes.

Continuous-time equalizers, though not as flexible do offer some advantages. To a degree, continuous-time equalization may be thought of as the limiting case of fractionally spaced discrete-time equalization, with the delay tap spacing reduced to zero. The result is a smoother shaping of the passing waveform and often less susceptibility to sampling clock jitter. Continuous-time filtering is also attractive in that, compared with their discrete-time counterparts, such circuits contribute very little in terms of noise, jitter, and potentially power dissipation to the system.

As chip area dedicated to integrating passive components decreases inversely with rising clock rates, passive continuous-time filter implementation becomes

more feasible. In fact, one recent paper reported a 30 Gb/s equalizer based on distributed LC delay taps [103]. Similarly, in [104], a passive RLC filter was used to enable 20 Gb/s data communication.

5.2.5 Disruptive Equalizer Technologies

While most of the advancement in data equalization is tied to incremental improvements, there have been some revolutionary designs which have stepped off the common path. This section identifies a few such technologies.

To overcome the inherent sensitivity of discrete-time equalizers to sampling uncertainty, a feed-forward equalizer was proposed that not only provides adjustable tap weights, but adjustable delay cells as well [105]. The intra-tap delay of 2-tap and 4-tap equalizers is controlled by an 8-bit digital-to-analog converter (DAC) and allows for tap delays to varying from 25-50 ps.

In [106], variable tap delay is also employed, but in this case the delays are not necessarily regularly spaced in time. As implemented in a DFE format, by allowing for irregular delay spacing, it is possible to make better use of the number of available taps. For example, with only a few taps, the equalizer may still address reflections and interference occurring several cursors out.

With the growing concern over jitter, equalizers specifically targeting deterministic phase degradation have been proposed [107]. This particular paper points out that it is impossible to target both ISI and DDJ by simply addressing amplitude distortion. The proposed solution is dedicated phase compensation to reduce jitter, in addition to standard equalization.

Based on similar concepts an equalizer which modulates the transmitted symbol pulse-width was presented in [108] and shown to exceed the performance of 2-tap FIR-based equalizers in many respects. Perhaps the greatest advantage obtained through this architecture is that it neither de-emphasizes nor boosts the signal amplitude to compensate for channel loss, implying that it may be applied directly to rail-to-rail signals prior to transmission.

Recognizing the need to account for both amplitude and timing degradation in the equalizer design, calibration schemes which consider the whole eye rather than just the vertical opening, as is done in the zero-forcing and MMSE cases, have been introduced. "Eye Opening" monitor circuits have been designed to work with the standard discrete-time equalizer topologies while imposing a smaller load on the signal path between taps [109].

5.2.6 Future Equalization

As has been shown, there are several flavors of channel equalization: transmit versus receive-side equalization, discrete-time versus continuous-time, fixed versus adaptable, etc. Clearly the ability to tune the equalizer response in realtime to compensate for environmental changes is critical. At the same time, the more tuning-compatible discrete-time equalizers are becoming more difficult to realize. At multi-GHz frequencies, the prospect of closing the adaptive feedback loop is limited by the delay through the weighting circuits, the summation node, and the decision circuit. As the challenge of tuning discrete-time equalizers grows comparable to that of continuous-time equalizers, the low power, low noise characteristics of continuous-time equalizers makes them more attractive, and efforts may need to be re-directed toward deriving reliable methods for continuous-time equalizer calibration.

In the chapter that follows, two new algorithms are proposed and shown to be effective in calibrating second-order continuous-time equalizers, using only one degree of freedom. While several potential compatible equalizer topologies are suggested, the focus of the contribution is the simple nature of the algorithms and the limited amount of supporting circuitry.

Chapter 6

Continuous-Time Equalizer Calibration

Over the course of several decades, channel equalization has attracted regular attention due to the fact that, regardless of the communication medium, the time inevitably arrives when physical bandwidth limitations must be overcome to exploit the capacity of the transmission channel. In the previous chapter, several forms of discrete-time and continuous-time equalization were presented for this purpose. While the flexibility and tunability of discrete-time equalization has lead to their dominance for decades, current technology does not permit their application in multi-Gb/s wireline environments has spurred interest in continuous-time equalization [98, 101, 102, 104], even though these circuits are often less flexible. But the argument was made that the "end of life" for cycle to cycle discrete-time equalizer adaptation may be approaching, and as a result, some level of effort should be directed at studying the problem of continuous-time equalizer tuning. The goal of this chapter is to advance a simple, yet effective, methodology for tuning continuous-time equalizers, as they are becoming more commonplace in high-speed environments.

While possible circuit implementations are proposed, emphasis is placed on the tuning theory with the assumption that physical realization of the theory will become easier as transistor speed increases. This statement does not imply that the proposals experience the same limitations associated with closing the adaptive feedback loop in discrete-time equalizers, because the methods presented here are meant to provide periodic, yet less frequent, recalibration and thus the delay through the feedback loop may be made arbitrarily long. The value of this work is in its simplicity and generality rather than in performance measurements associated with

a specific communication link. That said, as a point of reference, the performance of the proposed equalization is compared with that of the optimal zero-forcing equalizer, the optimal MMSE equalizer, and the best possible MMSE approximation to the two optimal responses achievable with the given second-order architecture, as applied to two target channel responses.

6.1 The Linear Equalizer

Before discussing the equalizer topology and associated calibration methods, it is useful to first identify the respective frequency responses of the two interconnects targeted in this experiment. Fig. 6.1 presents the insertion loss or transmission gain of the six inch and twenty inch FR4-based PC board channels to be addressed.

The equalizer used to verify the calibration techniques put forth in this chapter is made up of a single zero and a complex pole resulting in a second-order equalizer transfer function of the form:

$$F(s) = \frac{s+a}{s^2+bs+c}, \tag{6.1}$$

or in terms of more physical quantities:

$$F(s) = \frac{s+z}{s^2+\frac{\omega_0}{Q}s+\omega_0^2} \tag{6.2}$$

where z is the frequency of the zero, Q is the circuit's quality factor, and ω_0 is the filter's natural resonant, or peaking, frequency.

To reverse high frequency losses, frequency zeros are commonly built into the equalizer transfer function [99, 101], and in this case a single zero is employed to produce a +20 dB/Decade rise in the filter frequency response. Unfortunately the zero alone is not enough to completely reverse the high frequency losses of the two target channels. A second zero could be introduced to compensate for an additional +20 dB/Decade, but as the channel responses do not fall off at such a logarithmic rate, a second-order denominator was included to generate some exponential shaping

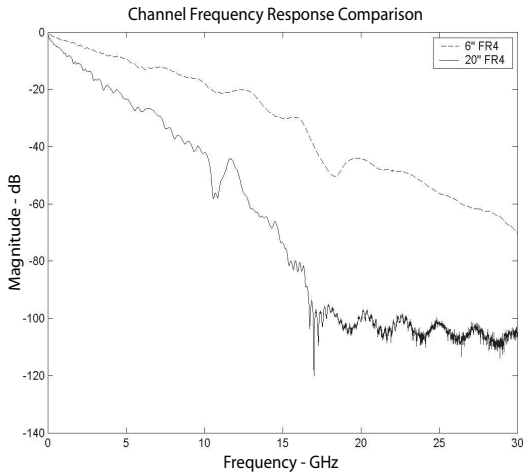
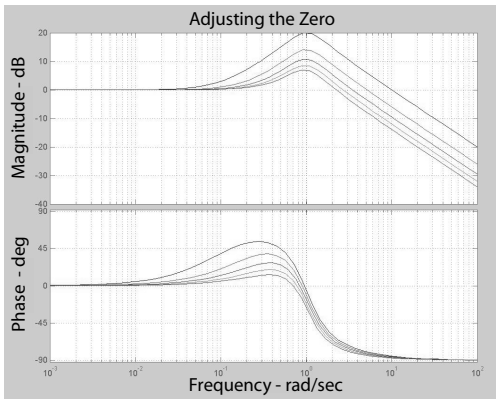


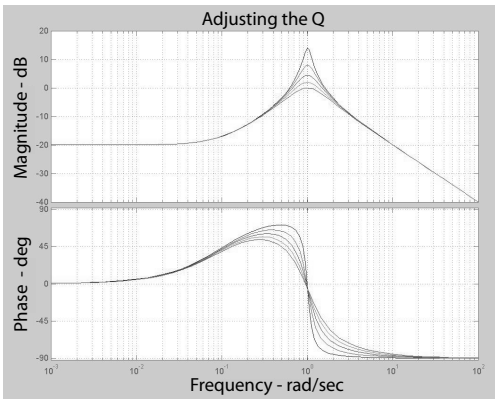
Figure 6.1: Channel frequency responses for the target six inch and twenty inch copper traces across an FR4 PC board.

of the filter response. The denominator contains two poles, which may be real and distinct (overdamped), real and equal (critically damped), or a complex conjugate pair (underdamped). Thus by changing the "b" coefficient in the denominator of the equalizer transfer function (6.1), or simply the circuit Q in (6.2), significantly different behavior may be achieved. Such tuning of the transfer function to produce high frequency peaking not only provides for more aggressive high frequency loss compensation, but also reduces high frequency noise amplification through the inherently sharp roll-off in the equalizer response above the resonant frequency.

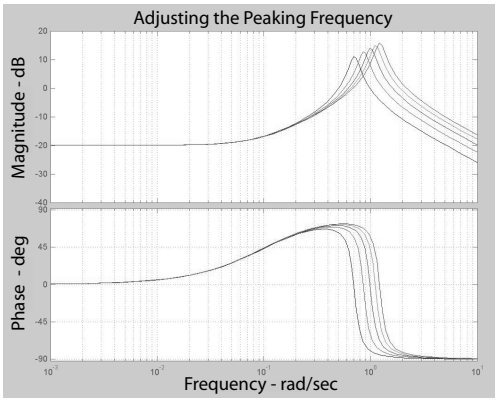
While the tuning approaches, to be described, were observed to work for the adjustment of any of the three parameters in the equalizer transfer function, emphasis within this explanation is placed on Q-tuning. But, if in practice the specific equalizer architecture favors the independent tuning of z or ω_0, then the algorithms presented here still apply after slight modification. Fig. 6.2 shows the variation in equalizer frequency response that can be achieved through adjusting either the zero, the Q, or the peak frequency ω_0. When considered in light of the target lowpass channels, it is clear that each form of tuning has the potential for providing at least coarse improvement in the combined channel-equalizer response.



(a) Zero-Tuning



(b) Q-Tuning



(c) ω_0-Tuning

Figure 6.2: Comparison of equalization through adjusting (a) the zero (b) the Q (c) the peak frequency (ω_0).

6.1.1 Equalizer Coefficient Placement

As has been mentioned several times, the greatest challenge associated with analog continuous-time equalizer implementation is the issue of tuning. While simultaneous tuning of multi-tap discrete-time equalizers is well understood, and their performance is easily predicted when the basic channel response and number of equalizer taps are known, methods for finding the optimal coefficient values in continuous-time equalizers are channel and equalizer specific. The methods proposed

in this chapter explore the effectiveness of a simple calibration method which consists of fixing two of the three coefficient values in the second-order transfer function while tuning the third.

To illustrate how the fixed coefficients may be chosen, consider the response of the six inch channel found in Fig. 6.1. If the circuit Q is designed to be the variable parameter, then both the z and ω_0 terms must be intelligently selected. To find a reasonable frequency location for the zero, a horizontal line may be drawn across the channel response curve at the -20 dB level. Knowing that the zero will produce a +20 dB/Decade boost in the response, it may be assumed that placing the zero a decade below the intersection of the channel response with the -20 dB line should lead to reasonable compensation up to that crossing frequency.

Choosing the appropriate location for ω_0 is a bit more challenging. When placed too low, over-equalization may occur if the effects of the zero and the peaking of the complex denominator overlap significantly. If placed too high, then the peaking may not contribute to the equalizer response over the frequencies of interest. However, this second possibility will likely not be a problem, as the parasitic loading associated with physical circuit implementation will certainly limit the maximum value of this term. For the six inch channel response shown in Fig. 6.1, and the goal of 20 Gb/s, f_0, or $\omega_0/2\pi$, was placed at two times the data bandwidth or 20 GHz (assuming half-rate clocking), to insure that the circuit, when tuned correctly, would flatten the overall response over the bandwidth of the data. With these parameters in place, the Q is then tuned to adjust the frequency response of the equalizer between the zero and the resonant frequency.

6.1.2 Equalizer Coefficient Tuning

With the fixed coefficients selected focus is shifted to the variable term. As discussed in the previous chapter, equalizer coefficient adaptation is directed by the minimization of a predetermined error metric. In theory, reducing the error coincides with approaching the optimal equalizer frequency response, thus the error term in general takes the form:

$$e(n) = s(n) - y(n) \tag{6.3}$$

where $s(n)$ represents the desired signal and $y(n)$ corresponds to the actual signal at the equalizer output. One of the most common methods for minimizing the error in practice is through the "steepest descent" or "gradient descent" algorithm, which in theory drives the coefficient update along the "steepest" path to the minimum error solution. To understand how this takes place, it is necessary to identify what the minimum error solution is. For the general case of the optimal transversal *Wiener* filter, the minimum error is associated with the mean-squared error criterion, often symbolized as:

$$\xi = E\left[|e(n)|^2\right], \tag{6.4}$$

where E[·] denotes statistical estimation.

The following derivation of the general adaptive coefficient update is taken from the presentation found in [110]. When following the well known least mean squared (LMS) adaptation algorithm, the coefficient update is expressed as:

$$w(n+1) = w(n) - \mu\nabla\xi, \tag{6.5}$$

where $w(n)$ and $w(n+1)$ are the present and future coefficient weights, μ is a scaling factor used to balance the trade-off between the rate of convergence and the residual error, and $\nabla\xi$ represents the gradient of the mean-squared error, or the derivative of the mean-squared error with respect to the coefficient weighting:

$$\nabla\xi = \frac{d(\xi(n))}{dw}. \tag{6.6}$$

In practical implementations, the statistical error estimate is often replaced with the instantaneous error estimate:

$$\hat{\xi}(n) = e^2(n), \tag{6.7}$$

and following this substitution, the error gradient may be calculated through:

$$\nabla\hat{\xi}(n) = \nabla e^2(n) = \frac{d(e^2(n))}{dw} = 2e(n)\frac{d(e(n))}{dw}, \tag{6.8}$$

with further substitution of the error value from equation (6.3) leading to:

$$\nabla e^2(n) = 2e(n)\frac{d(s(n) - y(n))}{dw}. \tag{6.9}$$

Considering that the desired signal is independent of the coefficient weighting, the expression further simplifies to:

$$\nabla e^2(n) = -2e(n)\frac{d(y(n))}{dw}, \tag{6.10}$$

and because for the general case of the single-tap transversal filter:

$$y(n) = w(n)x(n) \tag{6.11}$$

with $x(n)$ corresponding to the signal at the input of the equalizer, the final estimate of the error gradient takes the form of:

$$\nabla e^2(n) = -2e(n)x(n), \tag{6.12}$$

and hence the final LMS update follows:

$$w(n+1) = w(n) + 2\mu e(n)x(n). \tag{6.13}$$

Of course, because this derivation is based on the tuning of an FIR-based discrete-time filter, it may not apply directly to the adaptive equalizer under consideration. For that assumption to be valid, a similar relationship between the error term and the update must hold true. Fortunately, as will be shown, the LMS adaptation does translate well to the second-oder transfer function proposed.

With the update established, the method for estimating $e(n)$ in practice must be identified. For the equalizer in question, where the low and high signal levels

are zero volts and one volt respectively, one choice for the error is to compute the difference between the ideal high voltage and the peak of the pulse response for a lone *one* (single pulse preceded and followed by long strings of *zeroes*.):

$$e(n) = 1 - S_{SP}(n), \tag{6.14}$$

where $S_{SP}(n)$ is the sampled single pulse peak value at each iteration. The drawback to this approach is that it focuses on the single pulse response, while ISI is a multi-pulse problem. Thus a more appropriate error term would account for the relationship between multiple pulses. Fig. 6.3 presents two new error terms used to calibrate the variable equalizer coefficient, both of which account for multi-pulse interaction. The upper window illustrates what might be called the *symmetric pulse* error, while the lower window might be referred to as the *reduced tail* approach.

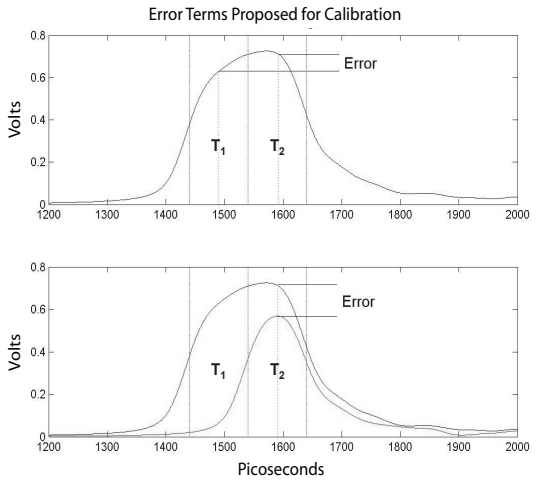


Figure 6.3: New error terms proposed for filter coefficient calibration.

"Symmetric Pulse" Equalization

The assumption of the *symmetric pulse* equalization method is that if a double pulse (two *ones* in a row) is sent, and the sampled cursor values from the two corresponding time cells are equal in magnitude, then the two pulses must be contributing equally to the overall pulse shape. Based on this assumption, it was thought that if two pulses contribute equally to the overall response, then even when ISI is not eliminated, at least it is made more consistent from bit to bit. It is this very fact that leads to the notion that there is no ISI in clocking signals, which is of course not technically true. Rather, the ISI is constant because the alternating nature of the clock results in a very consistent pattern, unlike the unpredictable patterns inherent in random data signals which result in the accumulation of ISI.

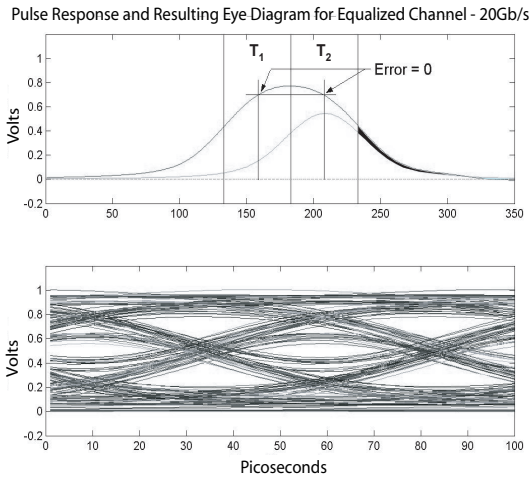


Figure 6.4: The upper window presents the 20 Gb/s single and double pulse responses of the six inch FR4 channel after applying the *symmetric pulse* tuning algorithm. The lower window presents the resulting 20 Gb/s eye diagram.

Unfortunately, this theory is not completely founded as there is often a difference between the pre and post-cursor tails of the individual pulse response. Thus the contribution of two consecutive pulses to the double pulse may not be distributed

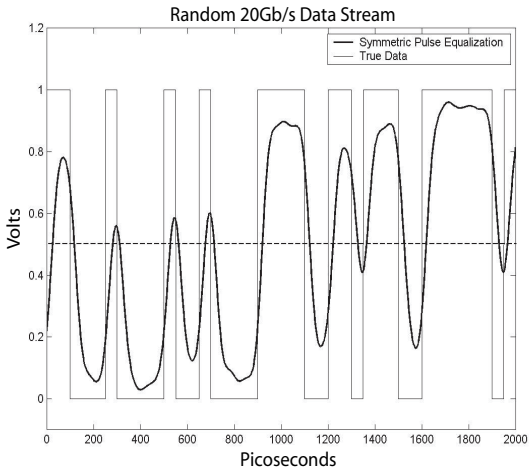


Figure 6.5: Comparison of the transmitted data and the received data after *symmetric pulse* equalization.

equally even when the resulting waveform appears symmetric. Hence, there may still be an accumulation of ISI even after *symmetric pulse* equalization. This can be observed in Fig. 6.4 where even with the error forced to zero in the double pulse response, there is still ISI build up as the post-cursor of the double pulse is larger than that of the single pulse (the difference is identified by the shaded area between the tails). Still, the resulting data eye is open as shown in the bottom window of the same figure. By comparison, the unequalized pulse response of this same channel produced the completely closed eye shown back in Fig. 2.5 used to illustrate the impact of ISI. Additionally, the simulated equalized signal found in Fig. 6.5, corresponding to the same unequalized data set shown in Fig. 2.4, identifies some improvement in that every bit transition breaches the detection threshold by at least 50 mV, whereas previously a large number of bits failed to even reach the threshold. While this does not represent significant voltage margin, the technique does produce favorable results, as evidenced by the open eye in Fig. 6.4, and can certainly be combined with transmit equalization for even more aggressive signal conditioning. Thus the associated tuning algorithm is worth presenting.

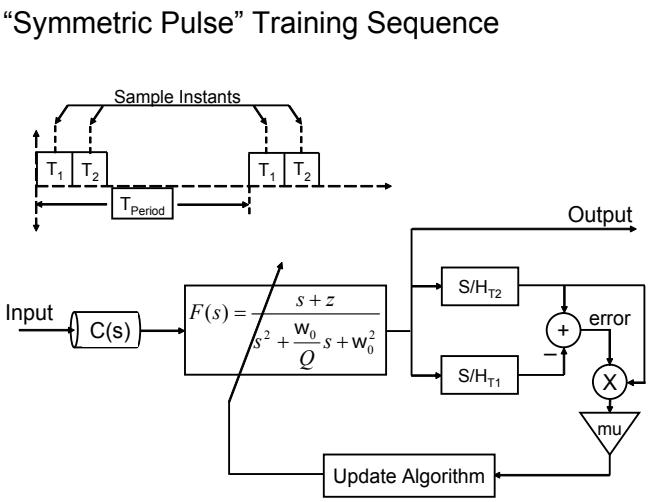


Figure 6.6: Block diagram of the *symmetric pulse* tuning algorithm. S_{T1} and S_{T2} are sample and holds taken during the T_1 and T_2 intervals respectively.

After fixing all but one of the filter coefficients, according to the method discussed, the remaining coefficient is tuned in such a way as to minimize the difference between the cursor samples from intervals T_1 and T_2, as shown in the upper window of Fig. 6.3. To achieve this, the variable coefficient is updated in accordance with the LMS algorithm, carried out as follows:

1. Initialize the $e(n)$ and $Q(n)$ terms.

2. For iteration n = 0,1,2,...

   Two consecutive pulses are sent and samples S_{T_1} and S_{T_2} are taken at the center of the corresponding time cells, as illustrated in Fig. 6.6. The new error term is calculated as:

$$e(n) = S_{T_2}(n) - S_{T_1}(n) \tag{6.15}$$

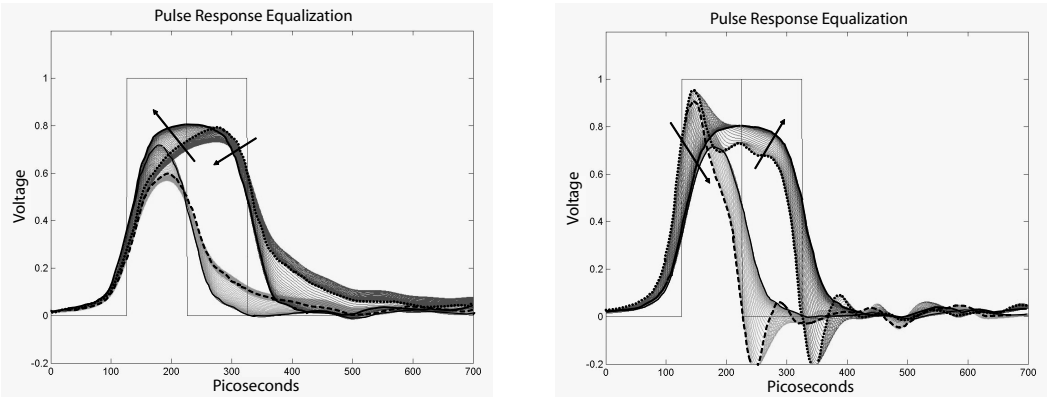
   where S_{T_1} and S_{T_2} are the samples of the double pulse taken at the center of intervals T_1 and T_2 respectively. One of the attractive attributes of this method of calibration is that the time between the double pulses may be made arbitrarily

long (limited only by the ability of the sample and hold circuitry to store an accurate measurement), allowing the tuning circuitry to operate much slower. By lowering the bandwidth requirements of the calibration circuits it may be possible to bias the active devices in the weak inversion region where more linear multiplication and other advantages (low noise, low power dissipation, etc.) are attainable [111, 112, 113, 114, 115]. The longer period between measurements also allows this technique to extend to higher datarates without the issue of adaptive loop stability, as experience by the established methods of discrete-time equalizer adaptation.

3. The new coefficient value is calculated as:

$$Q(n+1) = Q(n) + \mu e(n) S_{T_2}(n) \tag{6.16}$$

where μ is the scaling factor discussed previously and $S_{T_2}(n)$ represents the LMS approximation to the gradient of the squared-error or the system input.



(a) Initially Overdamped

(b) Initially Underdamped

Figure 6.7: Effect of *symmetric pulse* calibration on the single and double pulse responses. (a) Starting from an overdamped condition. (b) Starting from an underdamped condition.

The case $e(n) > 0$ occurs when the overall channel-equalizer response is somewhat overdamped with $S_{T_2} > S_{T_1}$. The update then increases the Q term in the denominator, thereby leading to a more underdamped filter response. This behavior of adjusting from an initially overdamped condition is illustrated in Fig. 6.7a, which presents the single and double pulse responses at several intermediate points along the *symmetric pulse* calibration process. If, on the other hand, $e(n) < 0$, some overshoot would be observed in the double pulse with $S_{T_1} > S_{T_2}$, and the algorithm responds by decreasing the Q, creating a more overdamped filter response, thereby leveling out the pulse, as shown in the process illustrated in Fig. 6.7b. It is observed that the process leads to the same solution regardless of the direction of the initial offset.

From this discussion, it might be questioned why the error estimate is not taken in the standard way, as the difference between the ideal signal level and the sampled equalized signal level. The response to this concern has two parts. First it must be argued that the goal of realtime adaptation (on a cycle to cycle basis) at multi-Gb/s datarates is unrealistic, and hence, periodic retraining through a simple data pattern as presented here is a better solution. As was just mentioned, this method avoids adaptive loop instability, because the bit period associated with the datarate may be several orders of magnitude shorter than the training period. Once it is agreed that the training sequence presented is a more reasonable, and perhaps, superior approach to the adaptation process, then the standard error of the ideal signal level minus the equalized level of the single pulse, in this case $1 - S_{T_1}(n)$, may be proven problematic.

Based on this error metric, and considering the nature of the training pattern, the sign of the error can not change, implying that the coefficient update will continue indefinitely until shut off by some other mechanism. On the other hand, when the error is taken as the difference between two non-ideal levels as proposed, the sign of the error may change, identifying the minimum error and fixing the tuned coefficient value.

In addition, it was claimed that the proposed methods allow for the tuning of any of the three coefficients, thus making the technique compatible with a greater

number of equalizer topologies. If the zero is chosen to be the tuned parameter rather than the Q, then step three in the algorithm may be changed to:

$$z(n+1) = z(n)\Big(1 - \mu e(n) S_{T_2}(n)\Big), \tag{6.17}$$

and similarly if ω_0 is to be the tuned parameter, then the update would be:

$$\omega_0(n+1) = \omega_0(n)\Big(1 + \mu e(n) S_{T_2}(n)\Big). \tag{6.18}$$

It should be pointed out here that the sign of the update changes when the frequency zero is tuned. Intuitively this is understood by looking again at the effect that tuning this coefficient has on the second-order response (see Fig. 6.2a). While raising the Q and/or peaking frequency leads to a more underdamped response, raising the zero actually increases the damping by shifting the high frequency compensation out to frequencies where it no longer matters.

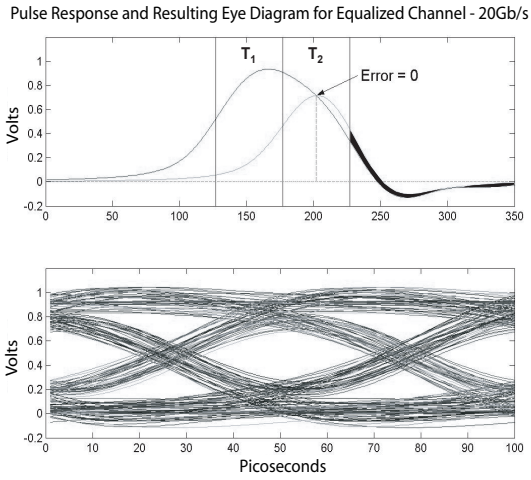


Figure 6.8: The upper window presents the 20 Gb/s single and double pulse responses of the six inch FR4 channel after applying the *reduced tail* tuning algorithm. The lower window presents the resulting 20 Gb/s eye diagram.

"Reduced Tail" Equalization

In the *reduced tail* equalization approach, the post-cursor of the double pulse is reduced in such a way as to produce relatively constant ISI over a series of successive pulses. In this case, the error term is generated from the difference between the sampled values of the second cursor of the double pulse and the cursor of the single pulse, as shown in the lower window of Fig. 6.3. The result is to force down the T_2 cursor value and consequently the tail of the double pulse thereby lowering the contribution of ISI to the following bit (see Fig. 6.8). Then when the pre-cursor of a third pulse is combined with the tail of the double pulse, the accumulation of ISI or the increase with each successive pulse is minimized. An alternative interpretation of that the *reduced tail* error zeros out the instantaneous ISI of the double pulse.

The resulting data waveform is presented in Fig. 6.9. The *reduced tail* response produces overshoot during the first of a train of pulses, but the level then remains relatively constant for the duration of the pulse train. The fast rise time that produces the initial overshoot is critical in cases where only a single *one* follows a long string of *zeros*. The simulation results presented in Fig. 6.9, based on the previous data set, show the equalized signal crossing the threshold by at least 150 mV with every bit transition.

Of course the double pulse tail could be attenuated even further for theoretically increased ISI suppression, but this would actually come at the expense of voltage margin. By forcing the second cursor of the double pulse to equal the cursor of the single pulse, a constraint is placed on the filter that allows for some overshoot while still guaranteeing a minimum steady-state level as demonstrated by the equalized waveform in Fig. 6.9; thus maintaining sufficient voltage margin. Another option would be to compare the cursor sample at the end of several consecutive pulses with the peak of the single pulse, but in simulation this led to varying levels of improvement. Were the number of transmitted pulses to span the duration of the unequalized channel impulse response, then the resulting error would account for all ISI. But the practical limitations of the sample and hold circuitry prohibits such an implementation. Thus, the original two pulse method provides the simplest, yet still effective,

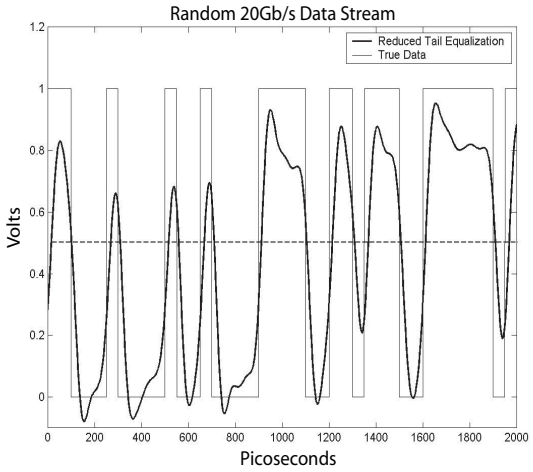


Figure 6.9: Comparison of the transmitted data and the received data after *reduced tail* equalization.

error metric. The *reduced tail* update algorithm, incorporating only two transmitted pulses, is as follows:

1. Initialize the $e(n)$ and $Q(n)$ terms.

2. For iteration n = 0,1,2,...

   Two consecutive pulses are sent as with the previous approach. Once the signal has re-settled to zero, a single pulse is sent. The new error term is calculated as:

$$e(n) = S_{T_2DP}(n) - S_{T_2SP}(n), \tag{6.19}$$

   where S_{T_2DP} and S_{T_2SP} are the sample of the double pulse taken at the center of the interval T_2 and the center sample of the single pulse as illustrated in Fig. 6.10.

3. The new coefficient value is calculated as:

$$Q(n+1) = Q(n) + \mu e(n) S_{T_2DP}(n), \tag{6.20}$$

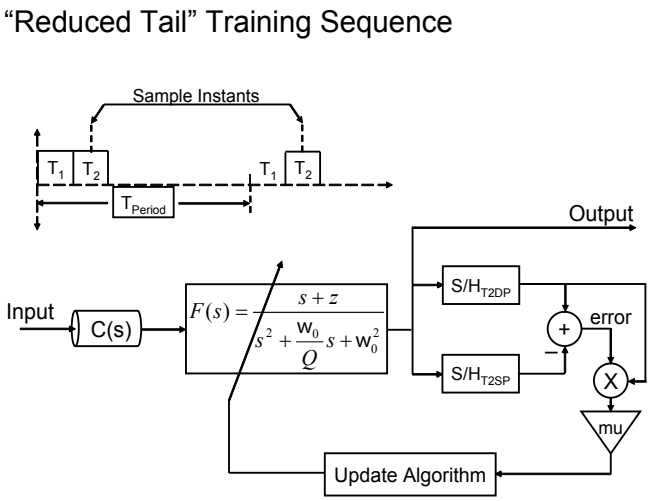


Figure 6.10: Block diagram of the *reduced tail* tuning algorithm. S_{T2SP} and S_{T2DP} are sample and holds taken during the T_2 interval of the single pulse and double pulse respectively.

where μ is again included to balance the speed/error trade-off and $S_{T_2DP}(n)$ represents the LMS approximation to the error gradient.

Similar adjustments may be made, as discussed previously, if either of the other two equalizer parameters is chosen as the variable. By way of comparison, Fig. 6.11 illustrates that error minimization is indeed achievable through the variation of any of the three parameters, with the assumption that the values of the remaining coefficients allow it. For the three simulations shown, the fixed coefficients were chosen to insure that calibration would lead to the same solution.

To understand how the second-order equalizer counters the effects of the channel, it is helpful to compare it with the optimal zero-forcing equalizer and the optimal MMSE equalizer. The discussion that follows compares three equalizer responses for the target six inch and twenty inch channels: the optimal zero-forcing response ($EQ_{OPT,ZF}$), the best approximation of the second-order equalizer ($EQ_{BEST,ZF}$) to the optimal zero-forcing response in the MMSE sense, and the response of the second-order equalizer calibrated through the *reduced tail* methodology, simply referred to now as the "adaptive" equalizer. Following that, the process

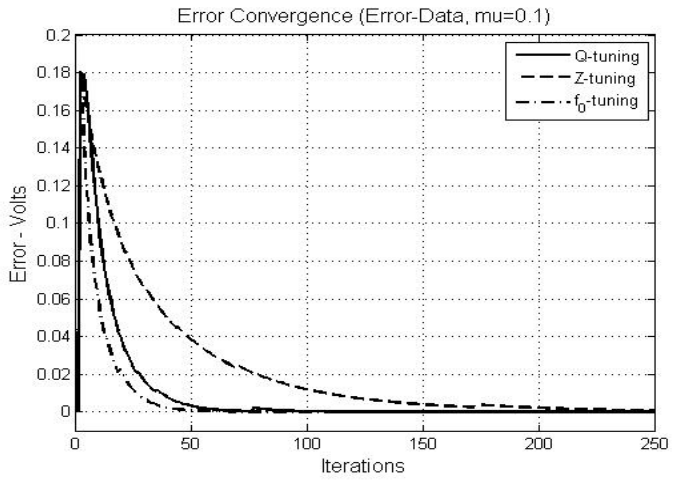


Figure 6.11: Error minimization achieved through the variation of each of the three equalizer parameters.

is repeated to compare the adaptive equalizer with the optimal MMSE topology ($EQ_{OPT,MMSE}$).

To compare the relative performance of the three equalizers, it is first necessary to explain how the optimal zero-forcing equalizer and the corresponding MMSE approximation to the optimal equalizer are derived. To begin with, the frequency response of the optimal zero-forcing equalizer is the inverse of the channel, or:

$$EQ_{OPT,ZF}(\omega) = \frac{\exp^{-j\omega\tau_0}}{H(\omega)}, \tag{6.21}$$

where $H(\omega)$ is the known channel response and $\mathrm{e}^{-j\omega\tau_0}$ is included to ensure linear phase at the equalizer output, with the constant τ_0 accounting for the time delay imposed by the equalizer circuit.

The best fit to the optimal zero-forcing equalizer is found by tuning the coefficients of the second-order adaptive equalizer to minimize the mean-squared error in the difference between the equalizer transfer function and $EQ_{OPT,ZF}(\omega)$ using the expression:

$$\omega_0, Q, z, \tau_0 = \arg\min \left\{ \int_{-\infty}^{\infty} \left| EQ_{OPT,ZF}(\omega) - EQ_{BEST,ZF}(\omega) \right|^2 d\omega \right\}, \tag{6.22}$$

or in terms of the coefficients to be computed:

$$\omega_0, Q, z, \tau_0 = \arg \min_{\omega_0, Q, z, \tau_0} \left\{ \int_{-\infty}^{\infty} \left| H(\omega) \frac{z + j\omega}{\omega^2 - \omega_0^2 + j\omega \frac{\omega_0}{Q}} - e^{-j\omega\tau_0} \right|^2 d\omega \right\}, \tag{6.23}$$

where the variables ω_0, Q, and z represent the frequency dependent coefficients of the second-order equalizer and τ_0 again represents the delay through the equalizer. What this expression seeks to do is tune the equalizer coefficients such that the product of the channel response $H(\omega)$ and the equalizer response $EQ_{BEST,ZF}(\omega)$ approaches one over the limits of integration, while at the same time, the combined phase response of the channel-equalizer product is also forced toward the linear phase behavior described by $e^{-j\omega\tau_0}$.

Due to the limited order of the proposed equalizer, the MMSE approximation to the optimal zero-forcing response must be bound to a limited frequency range in order to optimize the fit. If the equalizer were forced to match the ideal response in its entirety, the error between the two would be distributed over the whole range, and significant error would occur over the bandwidth of the data. By limiting the range, however, the equalizer is allowed to adapt more closely over the bandwidth of the data, while pushing the error out to frequencies where the spectral energy of the data is minimal. Thus the limits of integration shown in equation (6.23) are reduced to range from DC to 20 GHz for the six inch equalizer and DC to 8 GHz for the twenty inch equalizer, with the higher end chosen through trial and error. In addition, due to the nature of the measured channel responses, for which no closed form expression is available, the integration required in (6.23) is actually carried out through the summation:

$$\omega_0, Q, z, \tau_0 = \arg \min_{\omega_0, Q, z, \tau_0} \left\{ \sum_{i \in B} \left| H(\omega_i) \frac{z + j\omega_i}{\omega_i^2 - \omega_0^2 + j\omega_i \frac{\omega_0}{Q}} - e^{-j\omega_i \tau_0} \right|^2 \right\}, \tag{6.24}$$

where:

$$B = \{i | \omega_i < 2\pi \times 20 \text{ rad/sec}\} \tag{6.25}$$

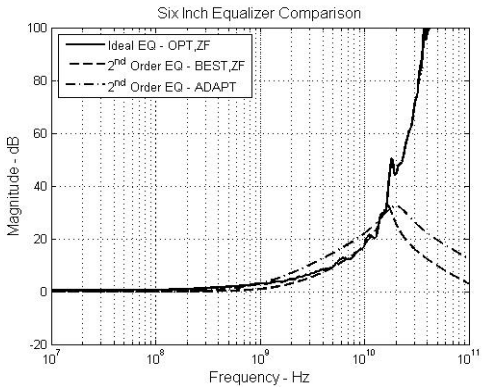
for the six inch channel and:

$$B = \{i | \omega_i < 2\pi \times 8 \text{ rad/sec}\} \tag{6.26}$$

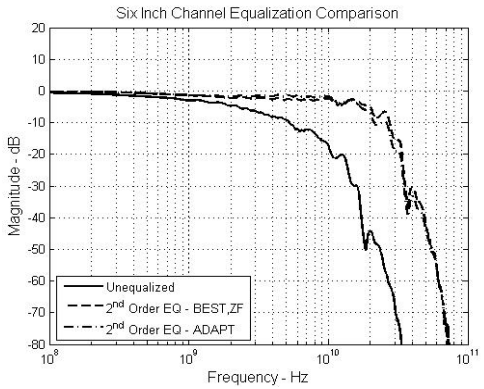
for the twenty inch channel, corresponding to 20 GHz and 8 GHz respectively.

Fig. 6.12 and Fig. 6.13, summarize the comparison between the optimal zero-forcing equalizer, the best fit approximation and the adaptive equalizer, with fixed coefficients chosen through the method suggested in Section 6.1.1, for the six inch and twenty inch channels.

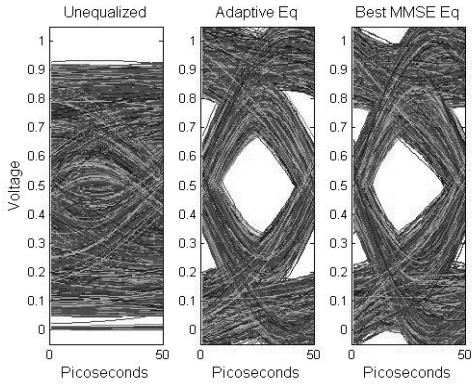
In Fig. 6.12a and Fig. 6.13a, the three distinct equalizer frequency responses are superimposed. While both figures show the best fit equalizer closely following the optimal zero-forcing curve, the adaptive equalizer gain appears too high at times, implying that the zero was initial chosen too low. This is understandable considering that the frequency of the zero was initially chosen without accounting for the potential compensation overlap provided by the resonant peaking. In a similar way, the peaking frequency of the adaptive equalizer is also observed to be too high. As a result, the best fit equalizer tends to provide a flatter response out to a higher frequency. This is more clearly observed in the six inch case presented in Fig. 6.12b, and as a result, the equalized eye, shown in Fig. 6.12c, exhibits less ISI at the sampling instant. Interestingly, however, the over-equalization provided by the adaptive equalizer still opens the eye a comparable amount, at least over the 10,000 symbols captured in the diagram.



(a) Equalizer Response



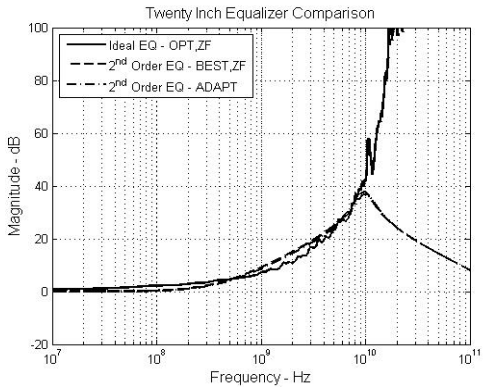
(b) Channel Response



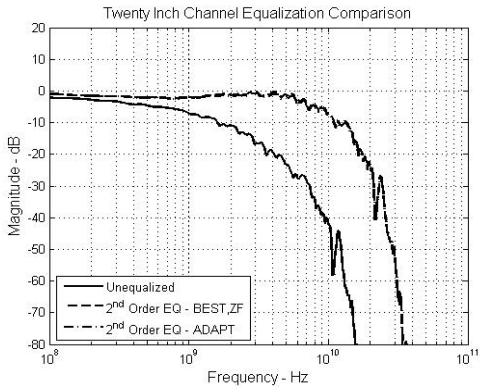
(c) Data Eyes

Figure 6.12: Zero-forcing equalization comparison for the six inch - 20 Gb/s interconnect.

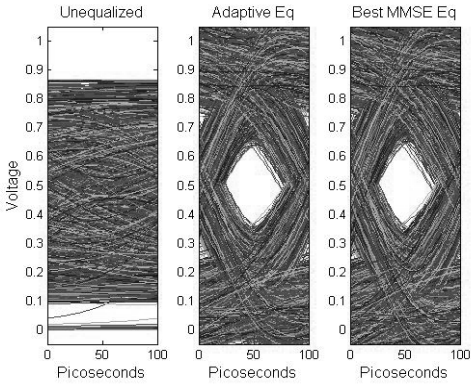
Further comparison of the relative performance of the three equalizers, was carried out by replacing the optimal zero-forcing equalizer with the optimal MMSE equalizer. As was discussed in the previous chapter, MMSE equalizers tend to provide better performance both in the presence of random noise and when the channel response is poorly behaved. The main difference between the computation of the MMSE equalizer and the optimal zero-forcing equalizer, is the inclusion of the noise floor N_0. As a result, the optimal MMSE equalizer is found through:



(a) Equalizer Response



(b) Channel Response



(c) Data Eyes

Figure 6.13: Zero-forcing equalization comparison for the twenty inch - 10 Gb/s interconnect.

$$EQ_{OPT,MMSE}(\omega) = \frac{\exp^{-j\omega\tau_0}}{H(\omega) + N_0}, \tag{6.27}$$

where $e^{-j\omega\tau_0}$ is again included to ensure linear phase at the equalizer output. The value of N_0 was chosen based on careful analysis of Fig. 6.1. In the figure, the noise floor was observed to reside between -105 dB and -110 dB, based on where the twenty inch channel measurement leveled off. Thus, this value was taken to represent

N_0. With that value established, the second-order approximation is then computed through the expression:

$$\omega_0, Q, z, \tau_0 = \arg \min_{\omega_0, Q, z, \tau_0} \left\{ \sum_{i \in B} \left| (H(\omega_i) + N_0) \frac{z + j\omega_i}{\omega_i^2 - \omega_0^2 + j\omega_i \frac{\omega_0}{Q}} - e^{-j\omega_i \tau_0} \right|^2 \right\}, \quad (6.28)$$

where ω_0, Q, z, and τ_0 again represent the peaking frequency, the filter Q, the frequency zero, and the delay through the equalizer, and where the summation is again taken over:

$$B = \{i | \omega_i < 2\pi \times 20 \text{ rad/sec}\} \quad (6.29)$$

for the six inch channel and:

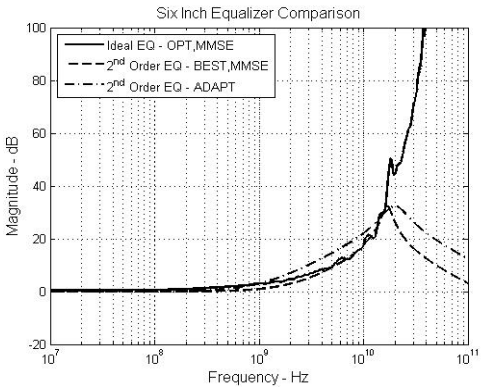
$$B = \{i | \omega_i < 2\pi \times 8 \text{ rad/sec}\} \quad (6.30)$$

for the twenty inch channel.

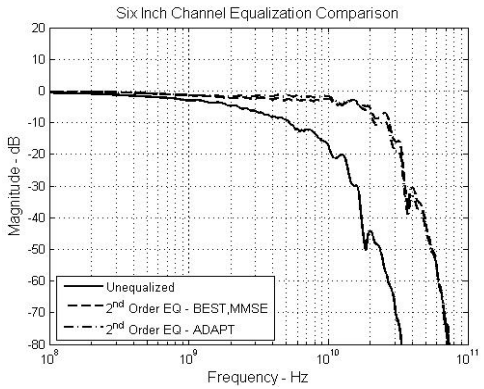
Table 6.1: Equalizer Coefficient Values - EQ_{ADAPT} / $EQ_{BEST,ZF}$ / $EQ_{BEST,MMSE}$

Rate	Length	Fixed Zero (GHz)	Tuned "Q"	Fixed $f_0(GHz)$
10 Gb/s	6 inches	1.00 / 2.04 / 2.07	0.19 / 0.83 / 0.80	20.00 / 16.61 / 16.88
10 Gb/s	20 inches	0.40 / 0.37 / 0.37	0.45 / 0.47 / 0.47	10.00 / 9.66 / 9.66
20 Gb/s	6 inches	1.00 / 2.04 / 2.07	0.33 / 0.83 / 0.80	20.00 / 16.61 / 16.88

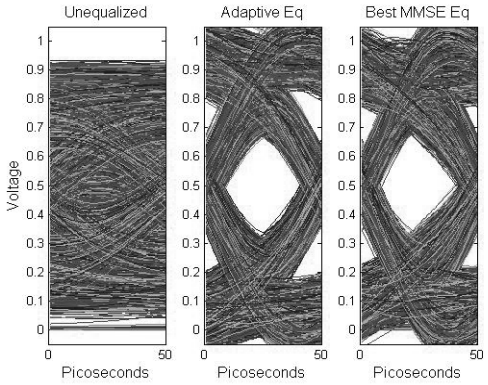
Table 6.1 compares the zero, Q, and peaking frequency of the adaptive equalizer calibrated through the *reduced tail* technique with the best fit coefficient values computed for the zero-forcing and MMSE cases. As expected, the relatively low noise floor led to nearly identical solutions for the optimal zero-forcing and optimal MMSE equalizers. The analysis was carried further to study the adaptive equalizer



(a) Equalizer Response



(b) Channel Response

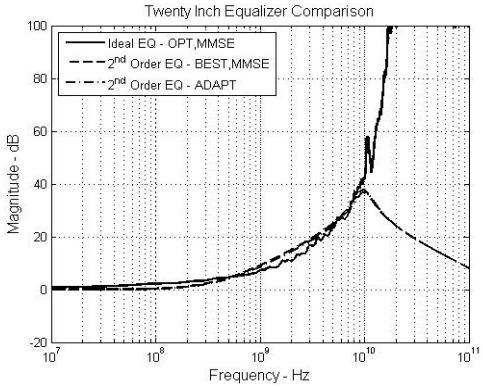


(c) Data Eyes

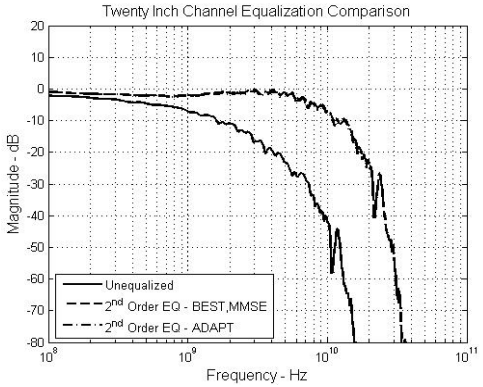
Figure 6.14: MMSE equalization comparison for the six inch - 20 Gb/s interconnect.

performance when starting from the best fit coefficient values. It was observed that the adaptive equalizer tended to settle closer to the MMSE approximation than to the zero-forcing approximation, yet a conclusion on which of the two equalizers it is approximating was difficult to determine.

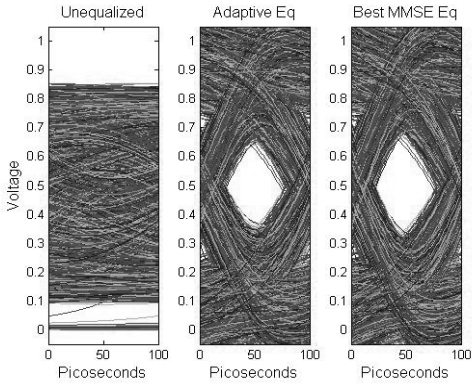
Once confident that the equalizer coefficients will converge through the proposed technique, it becomes prudent to look for additional points of simplification. In general, the LMS adaptation algorithm may be simplified by relying upon the sign of the error and/or the sign of the sampled pulse value rather than on the actual



(a) Equalizer Response



(b) Channel Response



(c) Data Eyes

Figure 6.15: MMSE equalization comparison for the twenty inch - 10 Gb/s interconnect.

analog values themselves. Thus the Q coefficient update can take on any one of the following four forms:

1. The error-data or standard LMS update:

$$Q(n+1) = Q(n) + \mu e(n) S_{T_2DP}(n); \tag{6.31}$$

2. The sign(error)-data or simply "sign" algorithm:

$$Q(n+1) = Q(n) + \mu \text{sign(e(n))} \text{S}_{\text{T}_2\text{DP}}\text{(n)}; \tag{6.32}$$

3. The error-sign(data) or "signed-regressor" algorithm:

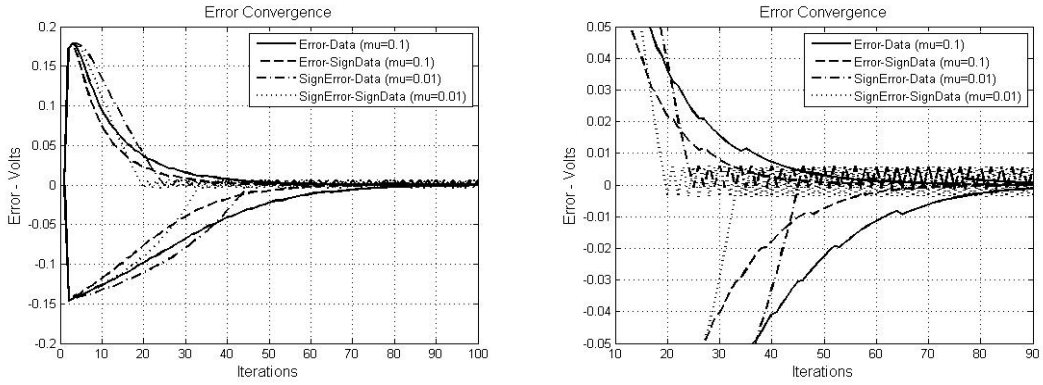
$$Q(n+1) = Q(n) + \mu e(n) \text{sign}\left(\text{S}_{\text{T}_2\text{DP}}\text{(n)}\right); \tag{6.33}$$

4. The sign(error)-sign(data) or "sign-sign" algorithm:

$$Q(n+1) = Q(n) + \mu \text{sign(e(n))} \text{sign}\left(\text{S}_{\text{T}_2\text{DP}}\text{(n)}\right). \tag{6.34}$$

In terms of circuitry, this implies that the error may be computed through the straight comparison of the two sampled pulse values rather than through a true analog subtraction circuit. In addition, subsequent scaling of the error term by the second sampled pulse value may be avoided altogether, as the sign of the sampled data always equals +1, based on the proposed training sequence, thus avoiding the need for true high-speed analog multiplication.

Sacrifices must be made, however, to enjoy this reduced complexity. The first trade-off is a higher average number of iterations needed to complete the calibration process. The second is an increased residual noise in the final tuned coefficient value that occurs as the computed error dithers around zero. These two points are illustrated in Fig. 6.16, which compares the performance of the calibration loop when implemented with the LMS, sign, signed-regressor, and sign-sign updates. To observe a comparable level of residual error, the μ factor of the two updates incorporating the sign(error) term was initially set two orders of magnitude below the μ associated with the methods using the true analog error values. But to insure complete calibration within one hundred iterations, a final value of 0.01 was chosen for the sign(error) μ, while the μ of the remaining two methods was left at 0.1. As a result, all four



(a) Error Convergence

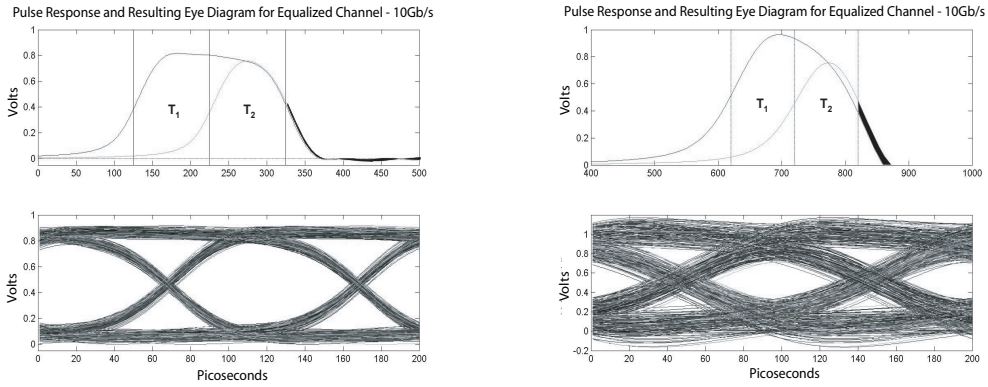
(b) Error Convergence Zoom

Figure 6.16: Simulations tracking the coefficient adaptation from both overdamped and underdamped initial conditions, when driven by the LMS, sign, signed-regressor, and sign-sign algorithms. (a) Zoomed out to show relative convergence time. (b) Zoomed in to show relative residual error.

algorithms are shown to converge in a comparable number of cycles, but as illustrated in Fig. 6.16b, the residual error of the sign(error) updates is measurably larger.

6.1.3 Additional Simulation Results

To further verify the effectiveness and generality of the calibration techniques, equalizers were designed not only for 20 Gb/s data across the six inch channel and 10 Gb/s across the twenty inch channel, but also for 10 Gb/s across both the six inch channel. Fig. 6.17a shows the 10 Gb/s data eye wide open after traversing the six inch channel, while Fig. 6.17b shows a clear eye opening after applying 10 Gb/s data to the twenty inch channel, which when unequalized resulted in a completely closed eye. In both cases, the equalizers were calibrated according to the *reduced tail* algorithm.



(a) 10 Gb/s Six Inches

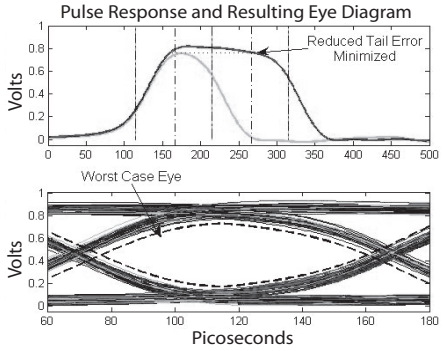
(b) 10 Gb/s Twenty Inches

Figure 6.17: (a) Pulse response and resulting eye diagram for a 10 Gb/s data stream (a) transmitted across the six inch channel (b) transmitted across the twenty inch channel.

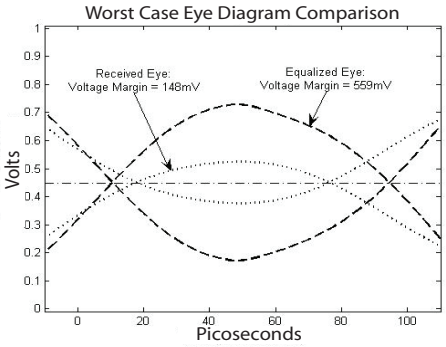
6.2 Performance Summary

In addition to comparison with the optimal equalizer responses, the second-order adaptive equalizer performance may be assessed in terms of the pre and post-equalizer eye opening and the achievable BER as derived through the methods presented in Chapter 3.

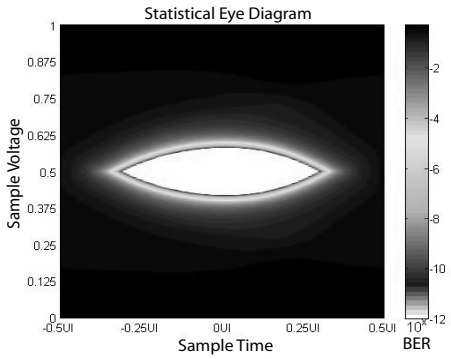
Fig. 6.18 provides a summary of the impact of the equalization on the six inch channel at a datarate of 10 Gb/s. In Fig. 6.18a, the single and double pulse responses (equalized through the *reduced tail* algorithm) and the corresponding simulated and worst-case eye opening are shown again for comparison. The eye captured through simulation approaches the worst-case boundary as expected, and with enough cycles passed through the system, the inner eye boundary would converge on the worst-case prediction. Fig. 6.18b contrasts the worst-case inner eye boundaries, for the same link condition, with and without equalization. As shown, equalization extends the received eye height and width from 148 mV_{pp} to 559 mV_{pp} and 58 ps to 85 ps, respectively. While an increased horizontal eye opening of 27 ps may not



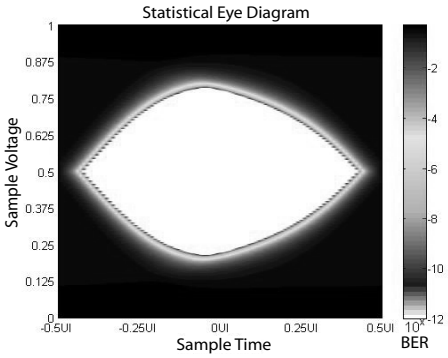
(a) Equalized Eye



(b) Worst Case Eye



(c) Unequalized Data Eye

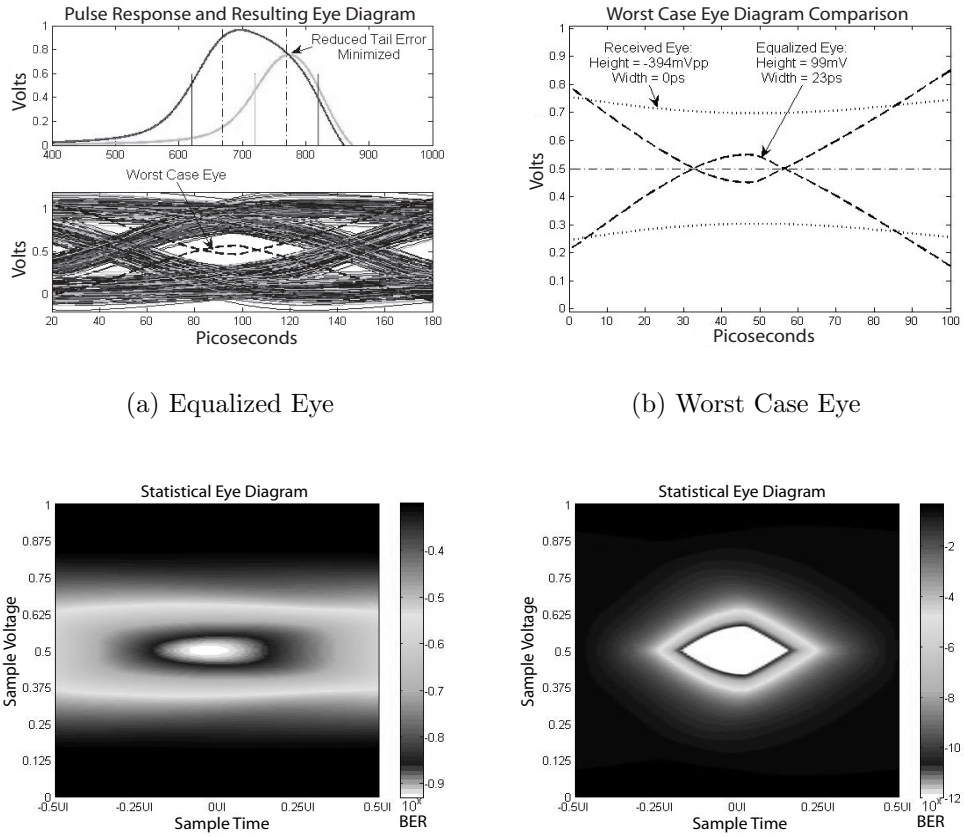


(d) Equalized Data Eye

Figure 6.18: Various illustrations of the impact of the *reduced tail* calibrated equalizer on the six inch channel at 10 Gb/s. (a) Single and double pulse responses and resulting eye diagram. (b) Worst case unequalized and equalized inner eye boundaries. (c) Unequalized statistical data eye. (d) Equalized statistical data eye.

seem significant, it is when considered in light of the total available bit time. The time enhancement reported here implies a jitter reduction from 0.42UI to 0.15UI. Fig. 6.18c and Fig. 6.18d provide a more descriptive view of the received data eyes (unequalized and equalized respectively) by shading the eye contour according the probability of error for any given sampling coordinate.

Fig. 6.19 provides a similar summary of the impact of the equalization on the twenty inch channel, again at 10 Gb/s. Fig. 6.19a again presents the single



(a) Equalized Eye

(b) Worst Case Eye

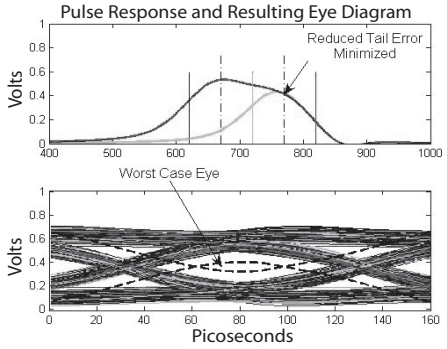
(c) Unequalized Data Eye

(d) Equalized Data Eye

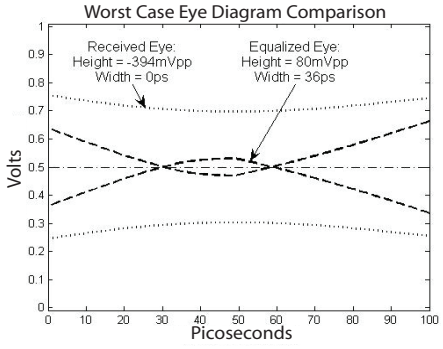
Figure 6.19: Various illustrations of the impact of the *reduced tail* calibrated equalizer on the twenty inch channel at 10 Gb/s. (a) Single and double pulse responses and resulting eye diagram. (b) Worst case unequalized and equalized inner eye boundaries. (c) Unequalized statistical data eye. (d) Equalized statistical data eye.

and double pulse responses (equalized through the *reduced tail* methodology) and the corresponding simulated and worst-case eye openings are shown again for comparison. In this case, the fact that the worst-case unequalized eye boundaries fail to intersect, as shown in Fig. 6.19b, indicates that the eye is initially closed. This same figure then shows equalization extending the received eye height and width from 0 mV_{pp} to 99 mV_{pp} and 0 ps to 23 ps respectively. Similar conclusions regarding the impact of equalization may be gleaned from the perspective of achievable BER. According

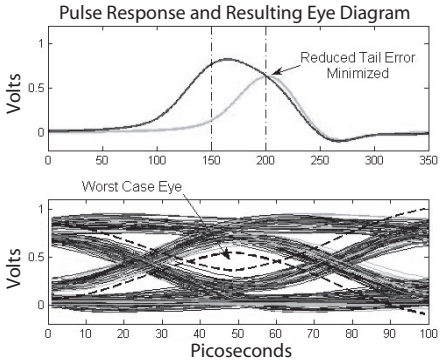
to the statistical eye presented in Fig. 6.19c, sampling directly in the center of the unequalized eye still corresponds to a probability of error near $10^{-0.9}$ or approximately 12.5%, while sampling in the center of the equalized eye shown in Fig. 6.19d, results in a BER of less than 10^{-12}.



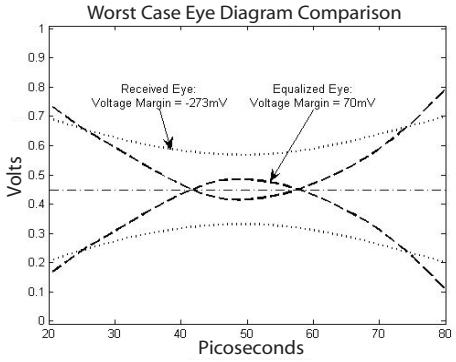
(a) Equalized Eye



(b) Worst Case Eye



(c) Equalized Eye



(d) Worst Case Eye

Figure 6.20: (a)-(b) Impact of the *reduced tail* calibrated equalizer on the twenty inch channel at 10 Gb/s. In this case, the frequency zero in the equalizer transfer function is initial set 3x higher than in Fig 6.19. (c)-(d) impact of the *reduced tail* calibrated equalizer on the six inch channel at 20 Gb/s.

Finally, Fig. 6.20 presents the eye opening achieved across the twenty inch channel at 10 Gb/s, as well as across the six inch channel at 20 Gb/s. The take-away

from these last figures is that only minor modification, if any, of the fixed equalizer coefficients was required to reach these levels of compensation over a variety of link configurations, implying a high level of generality in the calibration methods.

To calculate the full link BER, it is first necessary to assume a specific distribution of uncertainty in both the sample timing and the reference voltage, as discussed in Chapter 3. For the purpose of this presentation the sample timing uncertainty was assumed to follow a bimodal distribution consisting of a 5 ps_{rms} Gaussian component and a 20 ps_{pp} DCD component combined through convolution with a 20 ps_{pp} uniformly distributed jitter component. Similarly the voltage uncertainty was assumed to follow a distribution comprising a 5 mV_{rms} Gaussian component and a 60 mV_{pp} uniformly distributed component.

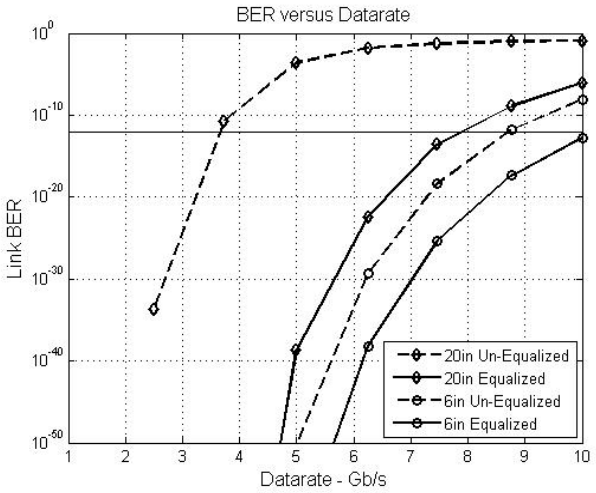


Figure 6.21: BER versus datarate for the six inch and twenty inch channels before and after equalization.

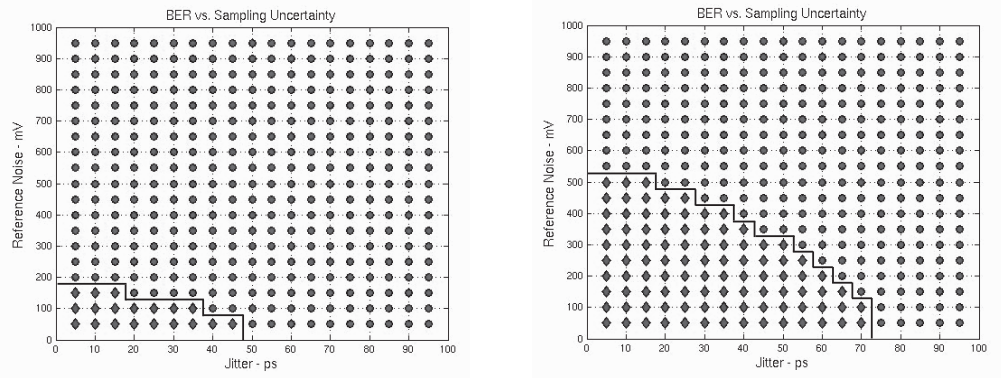
With these values selected, the BER of each link was calculated with and without equalization at several datarates. The results, presented in Fig. 6.21, show equalization consistently increasing the achievable BER at datarates ranging from 3 Gb/s to 10 Gb/s by two or three orders of magnitude. For a target BER of 10^{-12} at

10 Gb/s, equalization proves to enable the six inch link, while the achieved datarate without equalization is greater than 10^{-10}. While the figure also shows a drastic improvement in the BER achieved across the twenty inch link, it is clear that either additional transmit equalization or reduced sampling uncertainty are required to reach 10^{-12} functionality at 10 Gb/s over this length. A second way to interpret Fig. 6.21 is to consider that for a specified sampling uncertainty, equalization improves the achievable datarate on the six inch channel from less than 8 Gb/s to greater than 10 Gb/s. An even more impressive claim can be made for the twenty inch channel, whose datarate is increased from approximately 3.7 Gb/s to 8.7 Gb/s through the equalization process.

To avoid the possibility of incorrect sampling assumptions, another way to compare the impact of equalization on system performance is by looking at the uncertainty tolerance of the link by manipulating the sampling uncertainty distribution while monitoring the BER. Fig. 6.22 shows the results of such a simulation on the six inch channel at 10 Gb/s. The peak-to-peak reference noise and sampling jitter were incremented in 50 mV and 5 ps steps, respectively, while the achieved BER was recorded. In the figures, diamonds were used to indicate when the link BER remained below 10^{-12}.

Based on the resulting data, shown in Fig. 6.22a, it is observed that the reference noise and jitter levels may never exceed 150 mV_{pp} and 45 ps_{pp}, respectively across the unequalized link. But more importantly, when the reference noise reaches 150 mV_{pp}, the sampling jitter may not exceed 15 ps_{pp}. And when the sampling jitter reaches the 45 ps_{pp} level, the reference noise may not exceed 50 mV_{pp}. Conversely, for the equalized case shown in Fig. 6.22b, the tolerable reference noise and sampling jitter combinations are extended to 500 mV_{pp} - 30 ps_{pp} and 100 mV_{pp} - 70 ps_{pp}.

In addition to the clear improvements evident in the simulated eye diagrams and BER simulations, a final approach to qualifying the equalizer's effect on ISI is to observe the autocorrelation of the equalized versus unequalized data sets. The expected autocorrelation of a truly random data set (white noise) would be a delta function, with zero values for all positive and negative time lags. Correlated noise,



(a) Unequalized (b) Equalized

Figure 6.22: Tolerable sampling uncertainty levels in terms of sampling jitter and reference voltage noise. (a) Unequalized. (b) Equalized.

on the other hand, would produce more spreading of the nonzero correlation values. Fig. 6.23 presents the autocorrelation of the transmitted data (pseudo-random), the unequalized received data, and the equalized received data. These particular data sets correspond to the transmission of 10 Gb/s data across the twenty inch channel. The minimum values shown are nonzero due to the windowing effect of calculating the autocorrelation from data sets of finite length. The grid lines represent single UI time lag increments. As expected, the transmitted data generated with Matlab's *rand* function shows a clear spike at the zero lag, with theoretically zero values at all other points, supporting the claim of uncorrelated data. The unequalized data is widely spread, implying a large amount of correlation from bit to bit. The equalized data is spread, but not as severely as the unequalized set, implying that the equalizer tends to decorrelate the data, or in other words, remove the interaction or ISI between neighboring data bits.

6.2.1 Possible Circuit Implementation

All of the discussion leading to this point has been based on an ideal transfer function. This section presents several circuit implementations which are

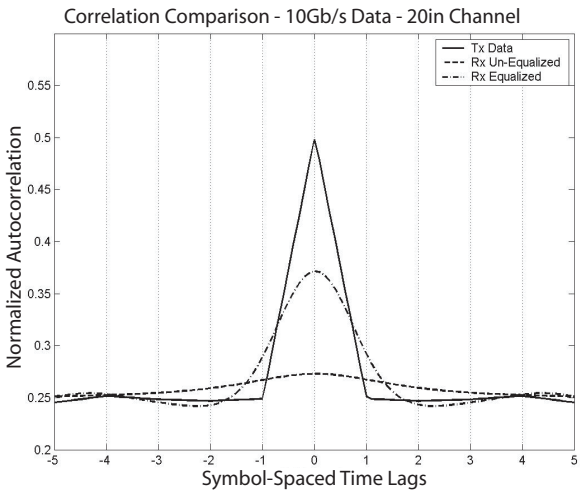


Figure 6.23: Comparison of the calculated autocorrelations of the transmitted, received, and equalized data sets.

capable of realizing the required equalizer response, with the understanding that obtaining the necessary amplification bandwidth in standard CMOS processes may still be challenging:

1. Sallen-Key amplifiers which provide for tunable Q factors are one possibility, though these circuits are highly sensitive to process variations [116].

2. Phase-shift filters, wherein the Q is set by the ratio of a pair of resistors [117] could also provide adaptable Qs were the resistors implemented with MOS devices or as a selectable resistive network.

3. The Cherry-Hooper amplifier designed in [118] for the purpose of reducing group delay variation across band-limited channels. The circuit is shown to be effective in restoring signal integrity in degraded signals through another Q-tuning scheme based on resistor relationships. While originally the tuning of the circuit was manual and static, flexibility could be designed in.

4. The Q-enhanced active lowpass filter proposed in [119]. This particular architecture allows for the independent tuning of Q with a single gate bias voltage.

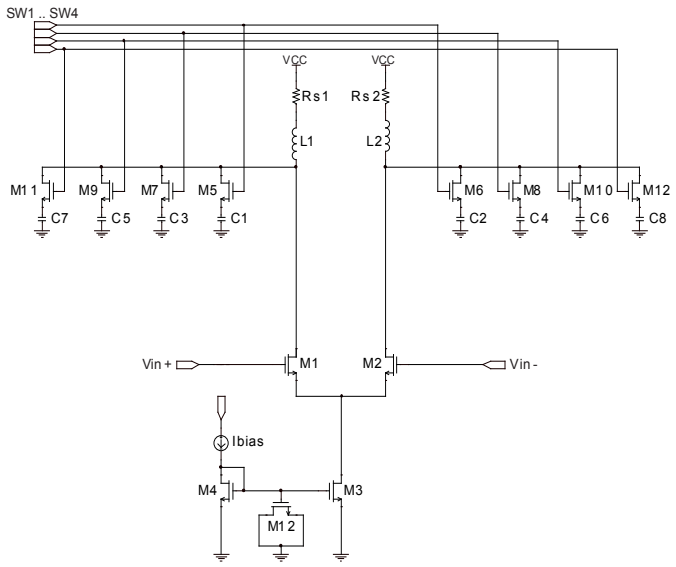


Figure 6.24: Equalizer with tunable inductive peaking.

5. One last possibility is the LC-based differential equalizer circuit discussed in [120] and shown in Fig. 6.24. This architecture provides tunable inductive peaking through the variation of the equivalent capacitive load produced by a pair of binary weighted capacitor arrays.

   The corresponding filter transfer function is:

$$F(s) = \frac{\frac{C_{gd}}{C_{gd}+C_L}\left(s + \frac{R_s}{L}\right)\left(s - \frac{g_m}{C_{gd}}\right)}{s^2 + \left(\frac{R_s}{L} + \frac{1}{r_{ds}(C_{gd}+C_L)}\right)s + \frac{1}{L(C_{gd}+C_L)}} \tag{6.35}$$

   where C_{gd}, g_m and r_{ds} are the gate-to-drain parasitic capacitance, transconductance and drain-to-source resistance of the input devices respectively. C_L is the combined capacitive loading selected through the array switches, R_s is the series resistance of the inductor, and L is the inductance. When the right-half plane parasitic zero is ignored, a reasonable approximation as it occurs far beyond the bandwidth of interest, the filter transfer function exactly matches the form assumed throughout this work.

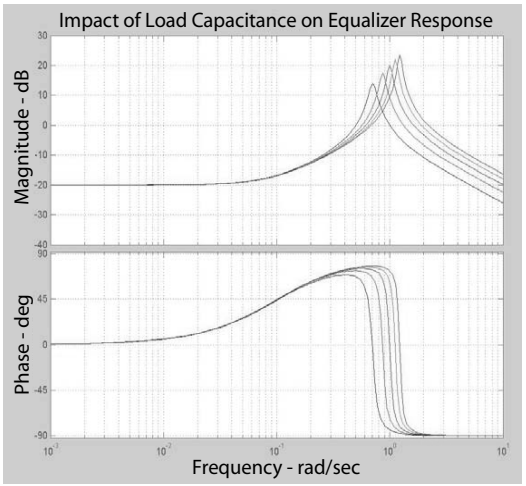


Figure 6.25: Frequency response of the suggested equalizer for various levels of tuned load capacitance.

While the transfer function implies some interdependence between the Q and ω_0 terms, the normalized circuit frequency response shown in Fig. 6.25 matches the desired equalizer response well and should provide a similar level of equalization in practice. This is because the third term in the denominator changes more quickly with variations in C_L, while the Q, which is dependent on both the second and third terms, remains relatively constant, approximating the independent tuning of the circuit's natural resonant frequency. As was mentioned, this may be accomplished by a simple modification of step three in the coefficient update algorithm. Two potential weaknesses of the implementation are the tuning resolution, which is determined by the unit capacitance in the load and limited by the parasitic capacitance of the switches, and the tuning range, which is limited by the area required to layout both on-chip spiral inductors and the required capacitor arrays.

This chapter concludes by presenting Table 6.2, which compares the maximum datarate enabled by the suggested second-order equalizer topology, when tuned with the *reduced tail* calibration algorithm, with previously reported achievements.

Table 6.2: Comparison of Equalizer Performance with Previously Published Work

Reference	**DataRate**	**Channel**	**Equalizer**	**BER**	**Gain**
[99]	270 Mb/s	200m Cable	CTLE(3)	n/a	40 dB
[121]	3.125 Gb/s	Optical	2-T FIR	4.5×10^{-15}	n/a
[108]	5 Gb/s	25m Cable	PW Modulation	10^{-12}	33 dB
[88]	6.25 Gb/s	30in FR4	4-T DFE	10^{-15}	20 dB
[102]	6.4 Gb/s	18cm FR4	5-T FIR + CTLE(2)	10^{-13}	20 dB
[25]	10 Gb/s	Optical	7-T FIR	10^{-12}	21 dB
[27]	20 Gb/s	7in FR4	4-T FIR + CTLE(1)	10^{-12}	16 dB
This Work	8.75 Gb/s	20in FR4	CTLE(2)	10^{-12}	22 dB
This Work	10 Gb/s	6in FR4	CTLE(2)	10^{-12}	16 dB

In the table, the term CTLE(n) refers to a continuous-time linear equalizer of order n. While several additional high-speed systems could have been chosen for comparison, the list presented is limited to systems implemented in a standard CMOS technology, thereby insuring a comparable level of difficulty in circuit realization. While the BER of the proposed system is highly dependent on the sampling uncertainty, if anything, the values chosen exceed the noise measured in the comparison systems. For example, the 20 Gb/s performance claimed in [27] corresponded to a total link timing uncertainty of 820 fs_{rms}. In addition, that system incorporated differential signaling to eliminate the problem of reference voltage uncertainty.

Based on the table, it is observed that the proposed algorithms perform to a similar standard, but do so with minimal complexity. For example, two of the links reported required both transmit and receive-side equalization to reach a similar datarate [27, 102]. At the same time, most of the discrete-time equalization implementations required several taps to reach a comparable level of performance [25, 27, 88, 102]. In terms of topologies, the third-order continuous-time equalizer presented in [99], most closely resembles the second-order response presented here, and still only achieves 270 Mb/s communication.

Chapter 7

High-Speed Clock Filter

Over the years, data channel signal integrity has enjoyed a disproportionately greater degree of attention, as the data signal's broadband nature makes it inherently more susceptible to degradation associated with limited channel bandwidth. Clock signal integrity, on the other hand, has received relatively little attention, as the clock's periodic nature side-steps pattern dependent degradation, and as a result, clock quality or lack thereof has contributed relatively little to I/O performance limitation in the past. At multi-Gb/s datarates, however, new phenomena including jitter amplification, in conjunction with stricter timing budgets to cope with vanishing margins, have raised interest in clock signal integrity.

As the high-speed clock finds use at more and more nodes within the system, as is the case with the source-synchronous and meso-synchronous topologies, the impact of clock signal integrity on link performance becomes more serious, as uncertainty in the timing of the clock, or clock jitter, rapidly degrades the maximum achievable datarate. Referring back to the source-synchronous link at the top of Fig. 2.1, and assuming multi-Gb/s communication, it is reasonable to expect the total clock jitter observed at the point of data capture to contain some, if not all, of the following components: jitter generated by the PLL used during the transmit-side serialization process; jitter generated by the transmit drivers, the majority of which stems from SSO noise; jitter amplification imposed by the band-limited, frequency-dependent characteristics of the transmission channel; jitter generation within the receive-side clock buffer, including DCD resulting from non-ideal DC signal levels

at the input buffer and rise/fall time asymmetries; jitter resulting from clock multiplication or phase interpolation circuits used to realign the phase of the associated clock and data signals; residual periodic jitter that results even after the static timing offset between the clock and data paths is minimized (This jitter component is amplified in the meso-synchronous link as intentional mismatch between on-chip clock and data routing lengths insures greater discrepancy between the phase of periodic clock and data jitter even after static timing offset has been eliminated.); jitter induced by power and ground noise at either end of the link; and finally jitter amplification incurred through the clock distribution network.

Interestingly, the CDR topology shown at the bottom of the same figure does not avoid much of the degradation just described simply by leaving out the forwarded-clock: jitter from the transmit-side system clock is still injected into the data during serialization; the resulting data jitter is exacerbated by the transmit driver circuitry; and jitter amplification across the band-limited channel still occurs. There are some differences at the receiving end of the link however. Mismatch between clock and data paths is not an issue as the clock path does not exist. But where some performance is regained through the avoidance of clock-data mismatch, it is quickly lost again due to the jitter generating characteristics of the clock recovery circuit. Finally, unless the clock is extracted on a per lane basis, it must still be distributed throughout the receiving port to capture the incoming parallel data, a process through which jitter again accumulates.

While it may not be a completely fair comparison, the performance achieved by the systems presented in [27] and [28], built from many identical circuit components, claimed achieved datarates of 20 Gb/s and 18.85 Gb/s over a similar channel for source-synchronous and CDR topologies, respectively.

This chapter describes a fully differential tunable bandpass filter fabricated in 90 nanometer CMOS technology intended for jitter reduction in the forwarded-clock signal employed in [27]. As will be shown, the filter serves to enhanced high frequency clock signal integrity through suppressing jitter-producing voltage noise and by attenuating several specific components of jitter directly.

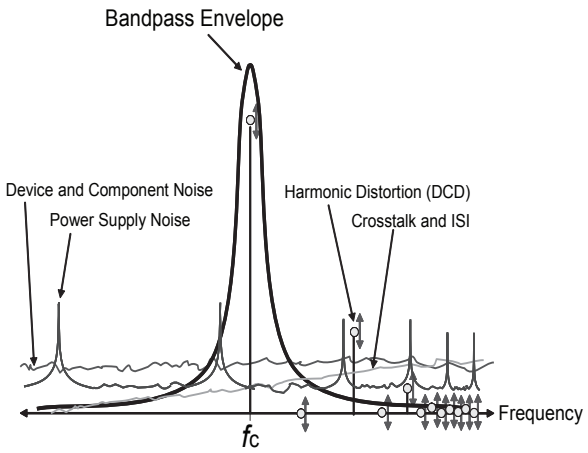


Figure 7.1: High-level frequency domain illustration of the impact that a bandpass filter should have on the spectral components of clock degrading noise.

Before proceeding, however, an intuitive argument for the use of bandpass filters in high-speed clock signal conditioning is provided. Fig. 7.1 presents the spectral components of an ideal clock waveform, with fundamental frequency f_c, along with the corresponding spectral characteristics of several forms of signal degradation previously discussed. The frequency response of a bandpass filter is superimposed for the sake of the discussion. The first thing to notice are the arrows pointed up and down at each harmonic component of the clock, representing harmonic distortion, which often includes DCD. Thermal noise, power supply noise, crosstalk, and ISI are also overlaid, though admittedly the noise levels are not to scale. Regardless, by identifying the spectral characteristics of the various noise sources with respect to the bandpass envelope, it becomes clear that sifting the dominant component of the signal through a bandpass filter will suppress noise occurring beyond the filter's bandwidth.

It may be argued that the filtering of the higher-order clock harmonics will further degrade rather than improve the condition of the clock signal. While it is true, that the slow transitions resulting from lost harmonics can result in greater peak-to-peak jitter, as was discussed in a previous chapter, those harmonics will

already be attenuated in the received signal when it arrives at the filter input, due to the band-limited nature of the interconnect. Thus no further slewrate degradation is imposed by the filter, only noise and jitter reduction. To further reduce the peak-to-peak jitter, the clock edges may be enhanced by following the filter with a carefully designed limiting amplifier.

7.1 Review of Clock Jitter

To appreciate the favorable impact that clock filtering provides to high-speed link performance, it is helpful to review the components of clock degradation.

7.1.1 Suppression of Random Jitter

As was discussed previously, RJ is the result of random noise or signal amplitude shifts translated into timing error at each signal transition. This, often linear, translation is inversely proportional to the slewrate, with faster edges reducing the jitter. This is one reason for the apparent jitter amplification that occurs incurred across the band-limited channel. As fast transmit edges are degraded by signal loss and irregular group delay, the slow edges observed at the receiving end of the line exhibit a measurably larger amount of jitter.

Earlier the concept of matched filtering was introduced as the optimal way for enhancing link SNR. Interestingly, according to the definition of the matched filter, a bandpass filter could be considered a sub-optimal "match" to a band-limited clock signal. Recall first that the impulse response of the matched filter is the time-delayed conjugate of the transmit pulse response. Then consider that the impulse response of a bandpass filter, with center frequency tuned to the clock's fundamental frequency component, is a damped sinusoid oscillating at the clock frequency, while the clock itself is not much more than a sinusoid after its edges are rounded by the harmonic attenuation of the channel.

For the measured channel response shown in Fig. 7.2, corresponding to a six inch copper trace in an FR4-based printed circuit board (the target channel for

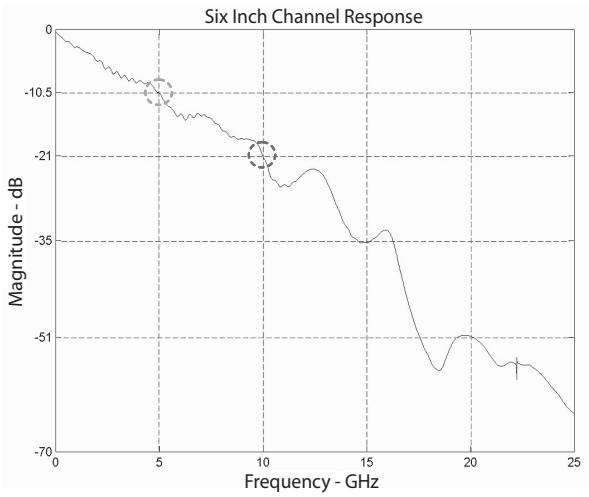


Figure 7.2: Target clock channel frequency response for a six inch FR4-based printed circuit board interconnect.

this chapter), the associated RJ and DCD amplification factors are presented versus clock frequency in Fig. 7.3.

These graphs identify some important characteristics of high frequency clock transmission. From Fig. 7.3 it is observed that RJ amplification tends to increase with operating frequency, and thus the frequency of the forwarded-clock should not be chosen lightly. The even faster increase in DCD amplification at higher frequencies was previously explained in light of the DCD accumulation that occurs due to the integrating nature of the channel. In the 20 Gb/s link presented in [27], two of the major factors driving the choice of clock frequency were the channel loss at the frequencies under consideration and the jitter amplification at those frequencies. Based on data like that found in Fig. 7.2 and Fig. 7.3, a quarter-rate clock (5 GHz) was chosen rather than the more commonly employed half-rate clock. Not only did this decision avoid the additional 10.5 dB of loss predicted in Fig. 7.2 to occur at 10 GHz, but it also avoided a random jitter amplification of nearly 2×, versus the jitter amplification anticipated at 5 GHz of just over 1×, as predicted by Fig. 7.3.

For comparison, the jitter amplification factors of two different bandpass filter configurations are presented in Fig. 7.4. Here again, the clock frequency is

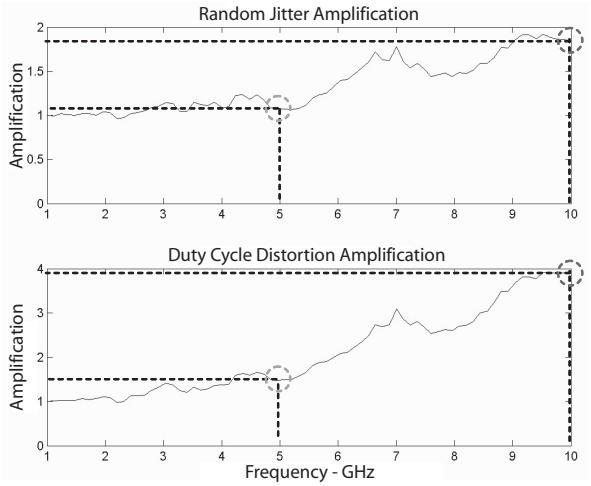


Figure 7.3: Anticipated RJ and DCD amplification at various clock frequencies for a six inch FR4-based printed circuit board interconnect.

swept, while the two filter's maintain a fixed center frequency of 5 GHz but distinct Qs of 2.5 and 5. Based on this figure, bandpass filtering should actually reduce the RJ present in a signal, as its jitter amplification factor is less than one. The figure also demonstrates that the jitter suppression provided by bandpass filtering improves with the filter Q, supporting a previous claim that bandpass filters may reduce cyclic phase noise and jitter by a factor of $\frac{\pi}{2Q}$ [122].

A final observation based on these last three figures is that even a relatively low-Q filter may not only counter the jitter amplification experienced across the channel, but may also remove much of the jitter that was present in the signal prior to the point of transmission, as will be demonstrated. Such is the case when the frequency response of the channel shown in Fig. 7.2 is followed by the response of the relatively low-Q bandpass filter to be presented shortly. The result is a combined jitter amplification of approximately 0.5, or the product of the channel and subsequent filter jitter amplification factors.

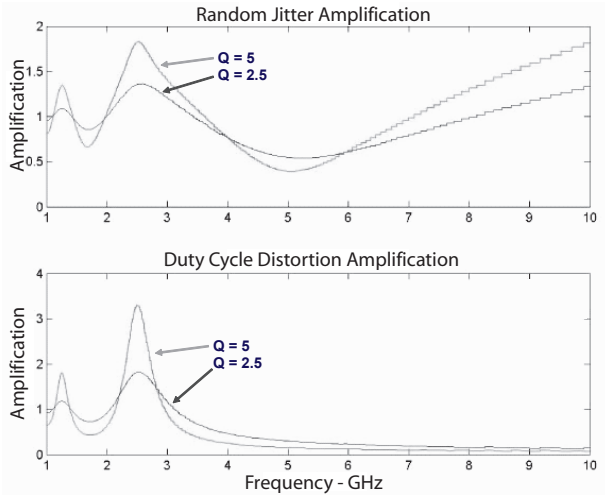


Figure 7.4: Anticipated RJ and DCD amplification for two bandpass filters with Qs of 2.5 and 5.

7.1.2 Suppression of DCD

Attenuation of DCD present in clock signals can be approached in two ways: attacking the source of the jitter (duty cycle error) and/or attacking the resulting jitter itself. Application of these two approaches may be separated into distinct operations on the low and high frequency components of the signal. As was discussed in Chapter 2, DCD results in a growing DC signal component through the integrating behavior of the lowpass channel. Yet at the same time, it was shown that DCD manifests itself as harmonic distortion, with the second harmonic being the dominant DCD component.

Thus, suppression of DCD requires the simultaneous attenuation of both the signal's DC component and frequencies equal to and greater than the second harmonic. One implication of this is that the common remedy of countering high frequency channel losses through highpass equalization not only fails to target DCD, but in fact tends to amplify this jitter component by amplifying the distorted higher order harmonics of the signal. A better solution would be a filter capable of amplifying the fundamental clock frequency while filtering off the corresponding harmonic

components. This could be accomplished with inductive peaking (high-Q, low-pass filtering) at the clock fundamental frequency, yet this would still fail to completely suppress the DC component of the signal, and therefore would sacrifice some potential attenuation of the duty cycle error as previously discussed. On the other hand, the inherent ability of a bandpass filter to amplify a narrow band of the signal's frequency spectrum, while completely removing the DC and unwanted higher order harmonic components, makes this filter an attractive candidate in the effort to mitigate DCD.

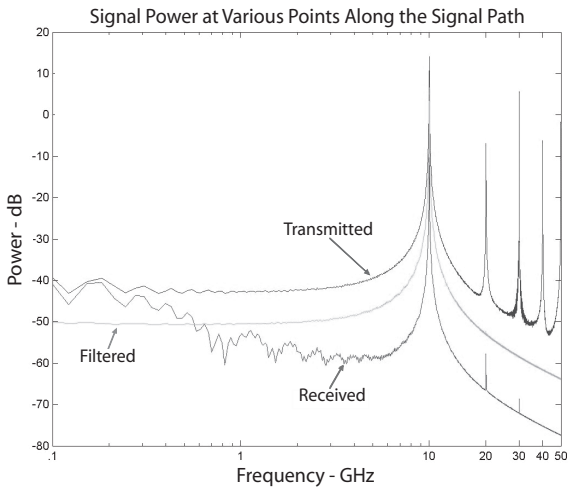


Figure 7.5: Power spectral densities at the transmitter, the receiver and following the bandpass filter.

The impact of bandpass filtering on the clock spectrum is illustrated in Fig. 7.5, where a clock initially exhibiting RJ and DCD is simulated passing over the six inch channel and then through a generic bandpass filter with a Q of five. As predicted by the earlier discussion on the harmonic components of DCD, it is observed that most even harmonics are comparable in magnitude to the odd harmonics at the point of transmission. After the channel, the received signal exhibits both a significant DC component and an attenuated, but still existing, second harmonic, implying the presence of DCD at the receiver. Following the bandpass filter, however, the DC

component of the clock is attenuated and the second harmonic is eliminated leaving a relatively pure and jitter-free sinusoid.

7.1.3 Periodic and Sinusoidal Jitter

While DCD has been shown to exhibit periodicity at frequencies 2× and above the clock fundamental, other lower frequency periodic jitter components are often observed in high-speed clock signals as well. For example, spread-spectrum clocking, which is simply a low frequency modulation of the transmitted clock phase used to reduce electro-magnetic emissions, is manifested in the time domain as a low frequency periodic jitter. For the most part, this particular jitter component is rarely a problem in that the modulated clock signal is used as the trigger for data serialization and transmission and cancels out during data capture at the receiving end, assuming reasonable channel matching. Even with the peak magnitude of the spread spectrum clock jitter specified in terms of nanoseconds, significantly greater than the fundamental clock period, its slow oscillation ($\approx$ 33kHz) provides tolerance to clock-data path mismatch. Periodic jitter components at higher frequencies, stemming from PLL jitter peaking or the excitation of IC package resonant frequencies, may be less tolerant to skew and must be addressed.

Fig. 7.6 provides an example of three sinusoidal jitter components which may result from the on-chip clock and data routing mismatch in the meso-synchronous topology. Even when the static timing offset is eliminated through adjusting the signal launch times at the transmitter, the propagation delay experienced by the clock, as it is distributed across the data port, can cause clock and data edges that were originally transmitted together to be several UI apart at the point of data capture, leaving a residual skew between the relative phases of clock and data periodic jitter components. As shown in the figure, the low frequency of the spread spectrum edge modulation contributes very little residual jitter even with a few nanoseconds of skew between the clock and data routing. On the other hand, higher frequency periodic jitter components can become completely out of phase or negatively correlated through the

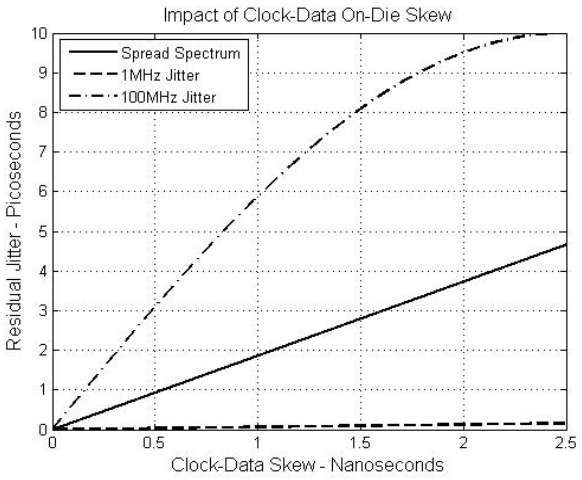


Figure 7.6: Residual sinusoidal jitter components that may result from on-chip clock and data routing mismatch.

same skew and directly add to the peak-to-peak jitter that must be tolerated by the capture operation.

To understand the bandpass filter's impact on periodic jitter, it is helpful to refer to the long-held approximation that "it takes Q cycles for a circuit to respond to changes at its input." Thus if a jitter event appears at the input of a high-Q filter, but is reversed within Q cycles, then the perturbation should not be observed at the circuit output. Such was the rebuttal found in [123] to criticism of the claim originally published in [122] that a bandpass filter could be incorporated into the feedback loop of a PLL to reduce cyclic phase noise and jitter by a factor of $\frac{\pi}{2Q}$. If this assumption holds, it would imply that clock jitter at frequencies above f_c/Q will be attenuated, and potentially eliminated, by a bandpass filter centered over the clock's fundamental frequency.

To verify this assumption, a sinusoidal jitter component with a peak-to-peak magnitude of 20 ps was superimposed onto a 5 GHz clock and passed through the bandpass filter to be presented, while sweeping the jitter frequency from 100 MHz to 10 GHz. Fig. 7.7 shows the results, with the simulated jitter amplification of the filter represented by the "*" symbols. Due to numerical issues the simulation

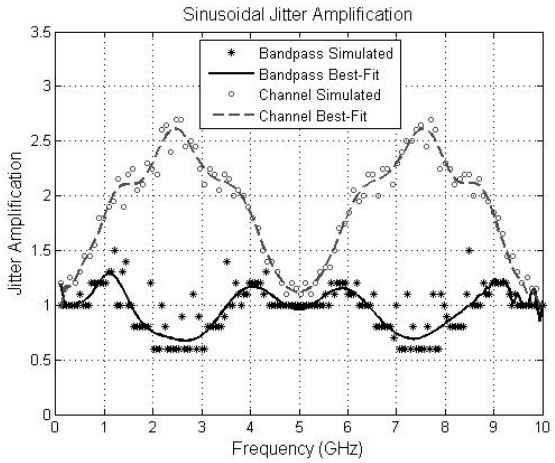


Figure 7.7: Sinusoidal jitter amplification of the proposed bandpass filter with clock frequency fixed at 5 GHz and sinusoidal jitter frequency swept from 100 MHz to 10 GHz.

produced several spikes depending on the phase relationship of the jitter and the underlying clock signal. To improve the read-ability of the data, best-fit curves are included. From the solid black line it appears that sinusoidal jitter amplification is symmetric about the clock frequency. This is due to the frequency relationship of the oscillating jitter and the underlying clock, which modulates the edge timing according to the ratio of the two frequencies, in other words, aliasing. When the clock and oscillating jitter frequencies are equal, the magnitude of the jitter will be the same at each clock edge and therefore will appear as a static phase shift or zero cycle-to-cycle jitter. Because the same jitter-to-clock frequency ratio exists at the output of the filter, the same static phase shift is observed in the output signal and the corresponding jitter amplification ($\text{jitter}_{out}/\text{jitter}_{in}$) equals unity. This does not imply that jitter amplification is worse at the bandpass filter's center frequency, but that the input jitter, and consequently the output jitter are both minimized at that point.

At the relative frequencies of $1/2f_c$ and $3/2f_c$, the filter reduces the peak sinusoidal jitter amplitude by as much as 40%, which is close to the predicted value

of $\frac{\pi}{2Q}$, or 0.5991 for this particular circuit implementation. It is actually possible to find frequencies at which the filter suppresses the sinusoidal jitter magnitude even further, as will be demonstrated near the conclusion of this presentation.

To further verify that the simulated results were not purely a numerical phenomenon, the simulation was repeated using the six inch channel response shown in Fig. 7.2, in place of the bandpass filter response. The data from this simulation is also included in Fig. 7.7 represented by the "o" symbols and the corresponding best-fit curve. Clearly the lowpass channel has a consistently negative impact on the magnitude of the sinusoidal jitter, regardless of frequency, though it does exhibit a similar symmetry. From these observations, it is clear that bandpass filters reduce unwanted periodic jitter over a range of frequencies, in which other filtering operations are likely to amplify the peak-to-peak jitter.

7.2 Existing Solutions for Reducing Clock Jitter

As was mentioned previously, PLLs are often employed within receivers to realign clock and data signals at the point of data capture and compensate for clock-data chip-to-chip routing mismatch and latency introduced by clock distribution networks. PLLs also commonly find their place in Process, Voltage and Temperature (PVT) compensation circuitry. One of the potentially positive side-effects of incorporating a PLL into the clock path is that when designed correctly the clock signal leaving the PLL may exhibit less high frequency jitter than the clock signal that was originally fed into the circuit.

This potential for high frequency jitter attenuation is associated with the PLL's phase tracking capability. One of the major considerations of the PLL design is the bandwidth of the control loop, which defines the frequency range over which changes in the input signal phase may be tracked by the circuit. Physically, the tracking bandwidth of the PLL is set by the cutoff frequency of an internal lowpass filter. Transition timing or phase variation at the PLL's input falling above the cutoff frequency of the loop filter are untrackable, and from the perspective of the tracking

mechanism, high frequency jitter is no different. Thus timing jitter beyond the bandwidth of the system is filtered off resulting in a lowpass jitter transfer characteristic from PLL input to output.

Unfortunately, jitter from the input signal is not the only component of timing error that may pass to the output of the PLL. Power supply noise and VCO phase noise both contribute to the total output jitter after being shaped by the jitter transfer characteristics of the system. According to [124], the jitter transfer of VCO phase noise through the output buffer is highpass in nature, while jitter stemming from the power supply sensitivity of the output buffer itself is bandpassed by the combination of lowpass and highpass functions associated with the loop filter and the output buffer respectively. Additionally, the phase detector, charge pump, and any frequency division circuitry will also contribute to the jitter reaching the PLL output. Thus it is possible for the PLL output to exhibit more jitter than the input, despite the I/O jitter filtering provided by the control loop. While several techniques to reduce the jitter generated from within the PLL have been studied, including a recently published work in which injection locking the reference clock to a slave oscillator was proposed and shown effective [125], most new methods under consideration add complexity to an already complicated circuit.

In addition to the possibility of contributing more jitter to the system than it removes, the very filtering nature of the PLL could prove detrimental to the communication system. For even though the jitter suppressing behavior of PLLs is often deemed essential, a case may easily be derived in which the jitter transfer characteristics of the PLL actually degrade the performance of the overall interconnect. For example if both the clock and data signals contain periodic jitter components, such as spread spectrum clocking or deterministic jitter resulting from the excitation of certain modes in the package resonance, then it would be critical to maintain the correlation between those components in both signals.

To apply numbers to this qualitative explanation, suppose both clock and data signals are transmitted exhibiting periodic jitter components at 500 kHz and 50 MHz. If the clock signal passes through a PLL with a loop bandwidth of 25 MHz

then the 50 MHz jitter on the clock will be filtered away and no longer correlated to the corresponding component of the data jitter. In addition, it is possible that the PLL will introduce new periodic components and certainly additional random jitter around the loop filter cutoff frequency due to a phenomenon known as jitter peaking. Thus it is reasonable to assume that the PLL will not only remove the 50 MHz jitter needed to match the data path, but it may also introduce jitter near 25 MHz that has no correlation to the data jitter, further degrading the performance sought through careful routing in the first place. In this particular case, the system performance may be improved by avoiding the inclusion of the PLL.

More often, the PLL designer must address the trade-off between filtering input signal jitter and tracking deterministic jitter components in the signal, through the selection of the loop bandwidth. If the loop bandwidth in the previous example was raised above 50 MHz to track the anticipated jitter component at that frequency, then additional random jitter between the original 25 MHz loop bandwidth and the current 50 MHz bandwidth would consequently pass to the output as well. In [27], the solution was to increase the loop bandwidth to 500 MHz to facilitate better jitter tracking, while at the same time filtering the incoming clock to compensate for the increased jitter passed by the high bandwidth PLL.

When maintaining jitter correlation between the clock and data signals is more important, a better solution may be to replace the PLL with a delay-locked loop (DLL), whose jitter transfer characteristics are very different. It is well known, and at times considered a negative characteristic, that DLLs pass jitter from input to output without attenuation. The jitter passing behavior of DLLs occurs because the waveform at the output is simply a delayed version of the input rather than a signal generated from within the system, as is the case with the VCO output of the PLL. In cases like that described above, such an allpass type of jitter transfer might be advantageous, as it maintains more of the clock-to-data jitter correlation while still providing for phase alignment and timing compensation.

To counter the increased random signal jitter which results with the DLL, a bandpass filter may be incorporated into the signal path to provide jitter filtering

above f_c/Q, where f_c is the center frequency of the filter, and ideally the frequency of the clock's fundamental component. This technique passes the lower frequency sinusoidal jitter, while reducing the high frequency jitter that has no correlated component in the data signal. The trade-off is that random jitter at frequencies between the alternative PLL bandwidth and f_c/Q will pass, though the noise filtering characteristics of the filter should provide additional benefit not accounted for in this discussion.

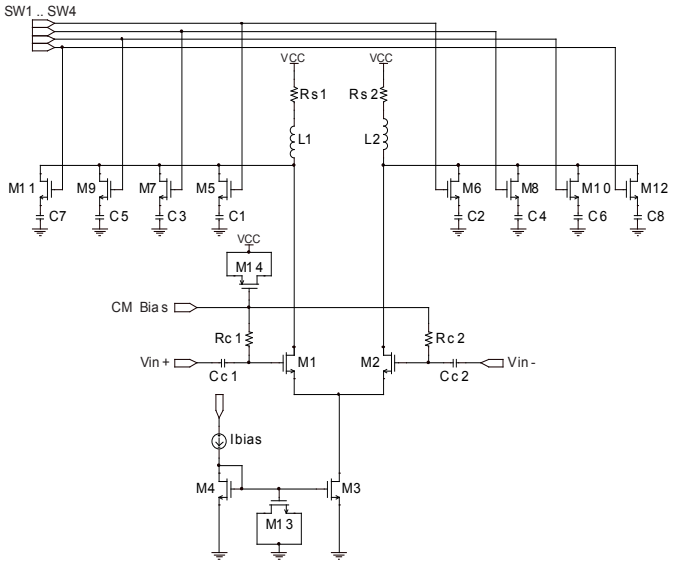


Figure 7.8: Schematic of the proposed bandpass filter.

7.3 Design of the Clock Filter

The design of a high frequency bandpass filter in standard CMOS requires several degrees of consideration. At the highest level, the trade-offs between digital and analog filter topologies are compared. In this case, the target center frequency of 5 GHz precludes the use of strictly digital techniques, due to the required circuit bandwidth. Even within the analog domain, the decision between discrete-time and

continuous-time architectures must be made. While discrete-time filters are routinely used at high frequency, for this implementation they are less attractive based on the large number of taps required to realize the filter response and the high level of noise expected from discrete-time implementation. At the next level, active versus passive filtering is considered. Based on the anticipated channel loss at 5 GHz (the target clock frequency), providing some gain within the circuit is desirable and implies that active filtering will be superior. The decision to achieve the filter frequency response through an LC-tank resulted from the need to minimize jitter generation from within the filter itself.

Table 7.1: Final Filter Component Values

Device	**Width**	**Length**
M1-M2	20μm	90nm
M3	50μm	250nm
M4	1μm	250nm
M5-M6	10μm	90nm
M7-M8	20μm	90nm
M9-M10	40μm	90nm
M11-M12	80μm	90nm
M13	10μm	250nm
M14	50μm	250nm
Component	**Value**	**Units**
C1-C2	0.06	nF
C3-C4	0.12	nF
C5-C6	0.24	nF
C7-C8	0.48	nF
Cc1-Cc2	0.05	nF
Rc1-Rc2	40	Ω
Rs1-Rs2	13.8	Ω
L1-L2	1.92	nH

Fig. 7.8 presents the proposed fully differential, LC bandpass filter and corresponding component values are listed in Table 7.1. Prior to adding the input

AC coupling, formed by components R_{C1}, R_{C2}, C_{C1}, and C_{C2}, the corresponding filter transfer function is:

$$F(s) = \frac{\frac{C_{gd}}{C_{gd}+C_L}\left(s + \frac{R_s}{L}\right)\left(s - \frac{g_m}{C_{gd}}\right)}{s^2 + \left(\frac{R_s}{L} + \frac{1}{r_{ds}(C_{gd}+C_L)}\right)s + \frac{1}{L(C_{gd}+C_L)}} \tag{7.1}$$

where L, R_s, g_m, C_{gd}, and C_L are the inductance, the parasitic inductor resistance, the transconductance of the input devices, the parasitic gate-to-drain capacitance of the input devices, and the equivalent load capacitance created by various combinations of a 4-bit binary weighted capacitor array, respectively.

The transfer function in (7.1) represents a second-order lowpass filter with frequency zeros in both the left and right-half planes. The right-half-plane zero results from the parasitic gate-to-drain capacitance of the differential input devices M1-M2 and occurs above 50 GHz allowing it to be ignored for the remainder of the analysis.

The addition of the coupling capacitors and pull-up resistors to the circuit input produces two favorable results. First the full circuit transfer function becomes truly bandpass due to the pre-filtering of the input signal according to the expression:

$$G(s) = \frac{s}{s + \frac{1}{R_C C_C}}. \tag{7.2}$$

By setting $\frac{1}{R_C C_C} = \frac{R_s}{L}$ and cascading the AC coupling circuitry with the DC coupled amplifier, the full transfer function becomes:

$$H(s) = F(s)\,G(s) = \frac{\frac{C_{gd}}{C_{gd}+C_L}s}{s^2 + \left(\frac{R_s}{L} + \frac{1}{r_{ds}(C_{gd}+C_L)}\right)s + \frac{1}{L(C_{gd}+C_L)}}. \tag{7.3}$$

A second favorable condition provided by the AC coupling is that the common-mode bias voltage of the input devices may be optimized without any dependency on the DC level of the incoming signal, providing the highest gain for the lowest bias current.

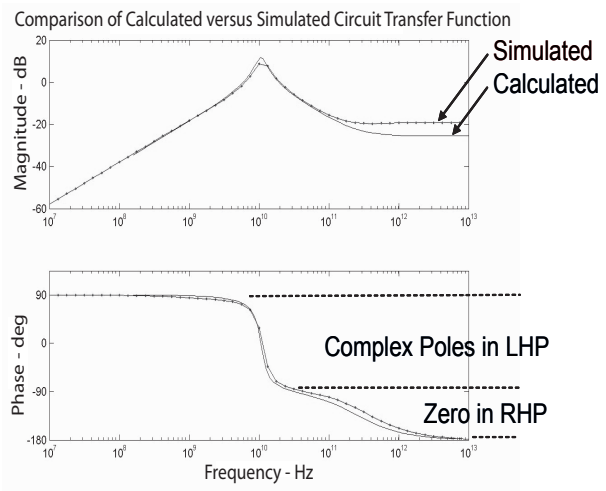


Figure 7.9: Comparison of the bandpass filter's frequency response with the expression found in (7.3).

The MOS-CAP (M13) connecting the gate of the tail device M3 to ground serves to improve the circuit's common-mode noise rejection by as much as 12 dB at higher frequencies by shunting noise from the current mirror and noise coupled through the parasitic gate-to-drain capacitor of the tail device to ground. In a similar way, the device M14 filters off high frequency noise on the common-mode bias node.

To reduce power dissipation, the positive power supply was set to 1.2V, while the bias current supplied by the current mirror is 100μA and is stepped up by the ratio of M3/M4 to provide a tail current of 5mA.

Devices M5-M12 are employed as switches to connect various combinations of the capacitor array in parallel with the inductor at the circuit output, thereby providing tuning of the filter's center frequency. MOS-CAPs were considered for finer tuning resolution, but were ruled out as the large voltage swing applied to the load would lead to nonlinear capacitance, and potential signal asymmetry.

Tuning resolution was improved by decreasing the unit capacitance of the capacitor array to 60fF, but this required an additional branch or 4-bits to achieve the same tuning range. Reducing the unit capacitance further provided no benefit in simulation, as the parasitic capacitance on the output node increases with each new

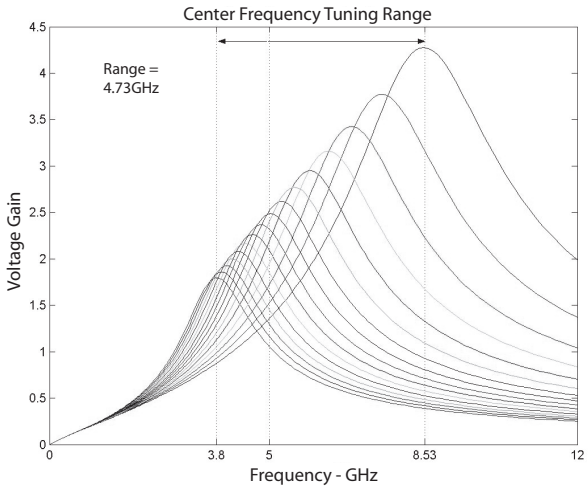
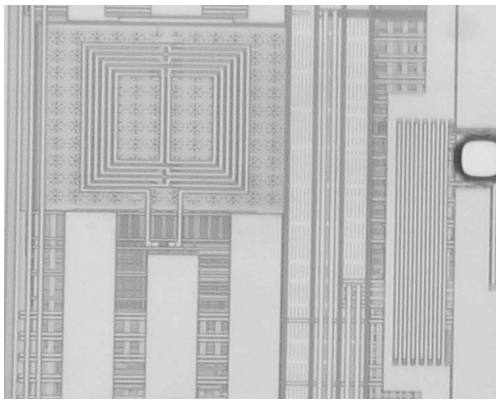


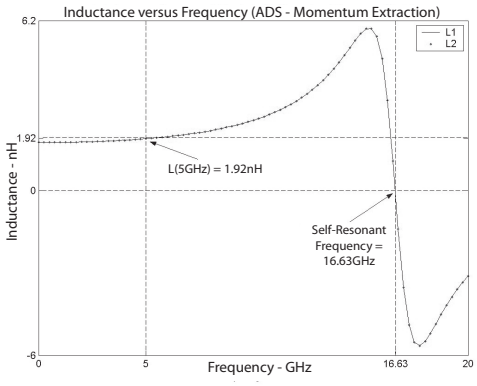
Figure 7.10: 4-bit tuning range of the proposed bandpass filter.

branch of the array and quickly becomes comparable in size to the least significant tuning bit. Due to the relatively low-Q value of the final filter (2.622) and the correspondingly wider passband, the required resolution in the center frequency tuning was relaxed. In the final implementation, with the unit load capacitance of 60fF, the frequency step from the ideal 5 GHz center frequency to the nearest settings above and below were on the order of 200-400 MHz, while the overall tuning range covered through the 16 steps was 3.8-8.53 GHz.

The differential inductive load was designed using Momentum, a 2-D solver available within ADS, and implemented in the form of a pair of interleaved spiral inductors, shown in Fig. 7.11a. Because the circuit was expected to provide good noise and jitter filtering even with a modest Q value, it was possible to approximate the target inductance of 2 nH and Q of 5 within a relatively small area (85 μm $\times$ 85 μm). However, achieving these values, while maintaining a self-resonant frequency 3x above the intended operating frequency of the inductor, was not trivial. Using similar values for the trace widths and the inter-trace spacing (1.8 μm and 1.5 μm respectively) resulted in lower parasitic capacitance at the expense of a slightly larger parasitic resistance, limiting the Q. After 2.5 interleaved loops, the simulated inductance was



(a) On-Chip Inductor



(b) Simulated Inductance

Figure 7.11: (a) Micro-photograph of the 85 μm $\times$ 85 μm differential, interleaved spiral inductors. (b) Simulated impedance response identifying an inductance of 1.92 nH at 5 GHz and a self-resonant frequency of 16.63 GHz.

only 1.5 nH, or 75% of the target value. To increase the inductance with an additional interleaved loop raised the inductance to 3.1 nH, but simultaneously reduced the self-resonant frequency to 11.5 GHz. The compromise was to follow the initial 2.5 interleaved loops with a pair of carefully matched individual loops within the left and right halves of the structure. This topology resulted in a final inductance of 1.92 nH, a Q of 4.38, and a self-resonant frequency of 16.63 GHz as presented in Fig. 7.11b. When placed within the circuit, the overall filter Q was reduced to 2.622, as mentioned, due to the switching devices and additional parasitics not associated with the inductor layout.

7.4 Bandpass Clock Filter Tuning Schemes

Perhaps the greatest challenge associated with implementing an analog filter is in the calibration or tuning of the filter frequency response. Fabrication process variation insures that the initial filter response will not match the intended or target response. Three of the more pervasive ways that process variation could impact the filter under consideration are:

1. Variations in trace widths and spacing, due to optical and feature etching phenomena, will impact the target inductance and parasitic resistance and capacitance of the spiral inductors and hence the center frequency and Q of the filter.

2. Variations in dielectric constants and thickness, due to irregular doping and layer growth, will effect parasitic resistance and capacitance, again altering the filter center frequency and Q.

3. Variations in the transistor characteristics (e.g. transconductance, drain-to-source resistance.) will effect the gain of the filter.

Thus, analog filters require tunability, and ideally self-calibration, if they are to be incorporated into high volume products. As bandpass filters are common in RF systems, several tuning techniques have already been proposed and explored.

7.4.1 Existing Tuning Solutions

In [126], it was suggested that two sinusoids, symmetrically offset about the desired filter center frequency, be passed through the filter and compared at the filter output. Measurable mismatch between the relative amplitudes of the two filtered signals would then correlate to an offset between the desired and actual center frequency. The filter response could then be adjusted until the amplitudes of the two filtered signals match, at which point the filter would be considered calibrated.

In [127], this signal balancing approach was enhanced by adding a third sinusoid at the desired center frequency, with the offset signals placed at frequencies where the signals were expected to be attenuated by a factor of two. The calibration circuitry was designed to keep track of the relative amplitudes of the three signals at the output of the filter. Calibration would be complete when the amplitude of the center signal equaled the sum of the two offset signals.

There are at least two difficulties inherent in these proposals. First, while they work well for high-Q filters, the asymmetric frequency response of a low-Q bandpass filter could lead to a systematic offset between the signals being balanced, making it difficult to judge when calibration is complete. A second issue, which is more a

problem for the first proposal, is that the uncalibrated filter center frequency must fall between the two offset signals, such that a measurable amount of each signal exists, otherwise the filter adjustment could be triggered in the wrong direction. By adding the third signal at the ideal center frequency, the uncalibrated center frequency constraint is relaxed to a degree, depending on the filter Q, because in this case, only two out of three signals must be present at the output.

To avoid this issue a third approach was proposed in [128]. In this case, however, several signals offset in frequency were passed through the bandpass filter. At the output of the filter, the frequency of the signal with the largest amplitude was identified and used to adjust the filter settings until the largest output signal resided at the desired filter center frequency. Thus, this technique would sift the many input signals through the filter and identify and adjust the filter's current tuning based on which signal passed through most easily. While this approach increases the amount of required circuitry, through operating with many signals rather than two or three, the result is better tolerance to the initial uncalibrated filter response.

In a similar approach, a technique was proposed in which white noise be passed through the filter, then passed through a limiting circuit, after which the dominant frequency component at the output could be measured. Filter tuning is continued until the frequency of the output align with the desired center frequency [129].

A still more radical tuning approach was proposed in [130], in which a fifth-order lowpass Bessel filter was to be tuned. In this case, the known phase response of the filter was incorporated into the calibration scheme. Initially, the authors considered passing the signal to be filtered through the uncalibrated filter, and then adjust the filter response until the appropriate I/O phase relationship was achieved. After further consideration it was recognized that, while the phase relationship between a sinusoidal input and a sinusoidal output could be accurately measured, the phase relationship of the harmonics of the full squarewave signal to be filtered, would be more difficult to track. Thus, rather than pass the actual signal through the filter, a sinusoidal VCO was designed to serve as a temporary input to the filter. Then

through measurement of the I/O phase relationship of the sinusoidal signal, the filter was tuned accurately.

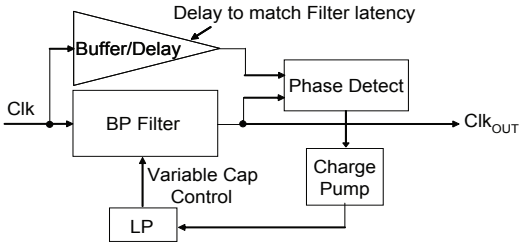
An additional approach incorporating both phase information and signal balancing was suggested in [131]. In this case the signal phase angles being compared were not the I/O phases, but rather the phase difference between signals at low and high offsets from the center frequency. Through sophisticated signal processing, down converters tuned to offset frequencies above and below the target center frequency are used to derive a phase angle error metric based on the phase relationship which should occur when that filter is tuned correctly.

Still another example of filter calibration based on phase-locking is found in [132], in which the authors tuned a bandpass filter through zeroing out the phase difference between the current entering the LC tank and the voltages at the terminals of the tank.

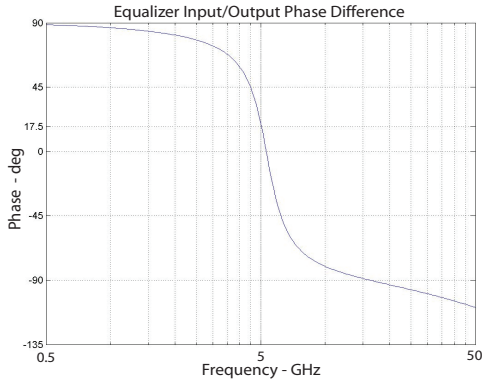
7.4.2 Proposed Filter Tuning Schemes

For the bandpass clock filter being presented, three self-calibration schemes were considered based on phase-locking, LC current cancellation, and peak amplitude detection.

The phase-locking topology, as shown in Fig. 7.12a, exploits the same principles used in [130], namely that the signal phase shift through the filter has a known value, which in the case of the bandpass filter is ideally zero as the reactive components of the filter transfer function cancel at the center frequency. In the physical realization of this circuit, however, there will be some residual signal phase skew due to propagation delay through the filter. To avoid a systematic phase offset at the phase detector, the reference path incorporates a delay cell intended to match the latency through the filter. The clock to be filtered is input to the system and passes through both the filter and the delay path. Then by comparing the phase at the output of the delay cell with the phase at the output of the filter, feedback is generated and used to zero out the phase discrepancy, which should ideally occur when the filter center frequency has reached the desired setting.



(a) Block Diagram



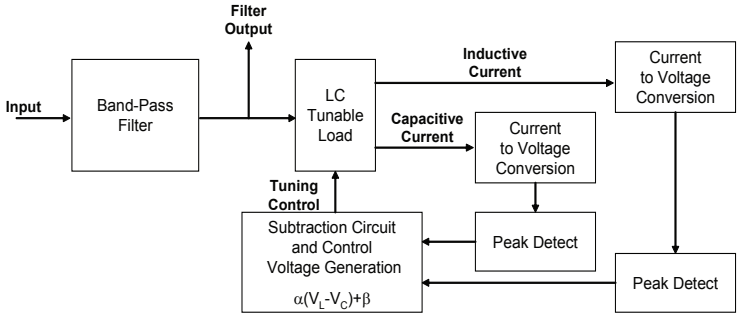
(b) Filter Phase Response

Figure 7.12: Phase Tuning: (a) Block diagram of a center frequency tuning scheme based on phase-locking. (b) Simulated filter phase response identifying the residual phase offset at the center frequency due to the signal propagation delay through the filter circuitry and the impact of the inductor's series resistor.

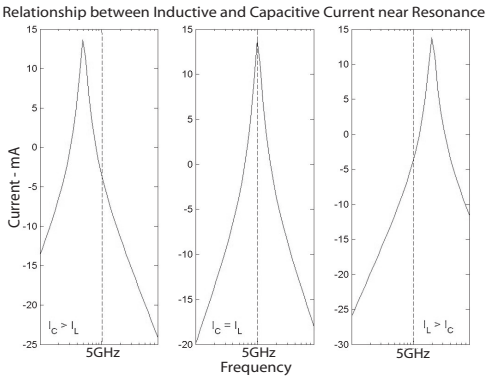
Fig. 7.12b presents the simulated phase response the filter. In this particular example, the observed residual phase offset at the desired center frequency results from the propagation delay spoken of and is canceled by the explicit delay placed in the reference path.

This phase-locking approach is attractive for several reasons. First, both signals compared by the phase detector, the filtered clock and the reference clock, are derived from the incoming signal avoiding the need for additional signal generation (extra VCOs, etc). Second, once the initial calibration is complete, the tuning circuitry does not negatively impact the quality of the clock signal and therefore may be left connected to provide continuing adjustments of the filter response to compensate for environment changes (temperature, etc.). Finally, the phase-locking system employs a standard PLL, with the exception of the reference signal derived from the input rather than provided by a VCO. With insignificant circuit modifications, the PLL which likely would already be included into the clock distribution network, could be optioned in a way to serve double-duty by also meeting the needs of the tuning

system, thereby resulting in increased system efficiency. Interestingly, the greatest drawback to this approach is that it requires the design of a full PLL, which if not needed for other reasons, is an added system complexity to be avoided if possible.



(a) Block Diagram



(b) Waveforms

Figure 7.13: LC Tuning: (a) Block diagram of a center frequency tuning scheme based on inductive/capacitive current comparison. (b) Waveforms corresponding to the calibration algorithm.

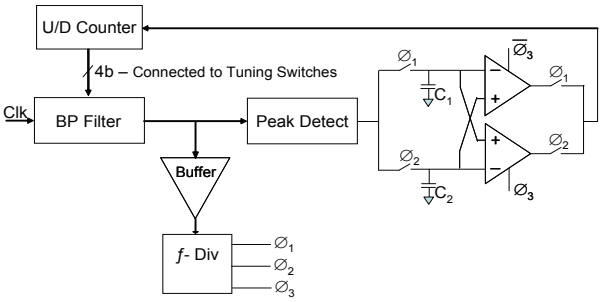
The LC current cancellation technique presented in Fig. 7.13a, similarly takes advantage of signal phase characteristics, but in this case it is the 180 degree phase shift between the inductive and capacitive currents circling within the LC

tank that is exploited. A sinusoid at the desired center frequency is passed through the filter and the current from the inductive and capacitive paths are converted to voltages across carefully matched resistors. The resulting AC voltages are then fed through peak detection circuitry to simplify the comparison process. A subtraction circuit computes the net voltage difference, if one exists, and scales the computed value to provide a control signal for subsequent filter tuning. The direction of the filter adjustment depends on which current is larger. As illustrated in Fig. 7.13b when the center frequency of the filter is lower than the frequency of the input signal, capacitive current will dominate as the capacitive reactance of the filter is smaller than the corresponding inductive reactance in that condition. Conversely, when the filter is tuned too high, the inductive current should dominate. The feedback loop attempts to zero out the net current extracted from the tank circuit, at which point the filter center frequency should be set correctly.

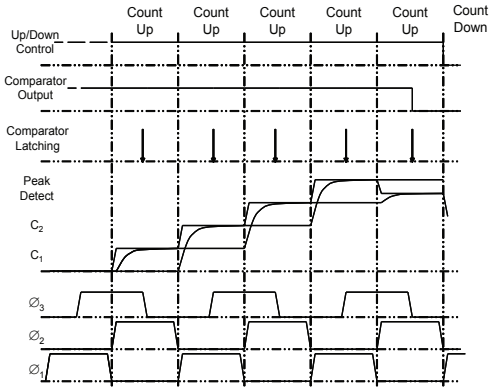
The final approach considered, and the one chosen for the prototype system, is based on the peak amplitude of the filtered signal. The nature of the bandpass filter response predicts that the amplitude of a passing signal will be greatest when the frequency of that signal is aligned with the filter's center frequency. The block diagram shown in Fig. 7.14a demonstrates how a single clock signal could be used to dial in the filter response.

The clock signal is first passed through the filter and consequently driven down two paths. In the first path, the filtered clock encounters a buffer, which isolates the circuitry to follow from the output of the bandpass filter. This insures that the additional tuning circuitry will not impact the quality of the final clock output negatively. Following the buffer, the signal is fed into a frequency divider circuit which outputs three lower frequency non-overlapping clock signals ϕ_1, ϕ_2, and ϕ_3 with 90 degrees of phase shift between them to be used for the sampling of the filtered clock amplitude. These signals are also shown at the bottom of the timing diagram found in Fig. 7.14b.

The second path taken by the filtered clock signal passes through a peak detection circuit which produces a DC voltage whose DC level is relative to the



(a) Block Diagram



(b) Waveforms

Figure 7.14: Peak Tuning: (a) Block diagram of a center frequency tuning scheme based on peak detection. (b) Waveforms corresponding to the calibration algorithm.

peak voltage of the alternating filtered clock signal. The signals ϕ_1 and ϕ_2 are then alternately used to sample the peak level of the filtered clock. The timing diagram in Fig. 7.14b provides and example of how the calibration might proceed.

1. First the peak detector output is sampled on the rising edge of ϕ_1 and stored on capacitor C_1.

2. The Up/Down counter, which is initially set to zero, corresponding to the highest center frequency setting, is increased by one and signal ϕ_2 samples the new peak level and stores it on capacitor C_2.

3. The two comparators shown in the schematic then compare the two sampled levels on the falling edge of signal ϕ_3 and the output of the upper comparator passes its value to the counter which consequently steps the tuning setting up or down accordingly. A high comparator output signifies that the second sampled value was greater and therefore the last adjustment brought the filter response closer to the desired response.

4. The counter is incremented and the new peak level is sampled by signal ϕ_1 and stored on C_1.

5. The two samples are again compared and the output of the lower comparator is passed to the counter on the rising edge of ϕ_3.

6. The process continues until the most recently sampled value is lower then the previous sample, indicating that the filter is diverging from the optimal setting, at which point the counter is decremented once to return to the previous tuning setting and calibration is disabled.

The resolution of this approach is somewhat limited by the fact that the difference in signal amplitude from setting to setting is relatively smaller for the few steps just on either side of the desired center frequency. Thus it is possible that the calibration will disable prematurely. This is not of great concern, however, considering the low-Q nature of the filter, where a center frequency tuning error of a few hundred Megahertz is significantly overshadowed by the wide bandwidth of the filter.

7.5 Performance of the Clock Filter

To verify the filter's response to common noise events, simulations were run in which power supply and common-mode noise were superimposed onto the passing clock signal and the peak-to-peak output jitter was noted.[1] The worst-case common-mode noise sensitivity occurred near 250 MHz, and resulted in approximately 30 fs of jitter per millivolt of input common-mode noise. Power supply noise sensitivity

[1] All of the simulations reported correspond to the extracted characteristics of the circuit, including an s-parameter representation of the inductor layout.

peaked at the filter's center frequency, and led to approximately 6 fs of jitter per millivolt of power supply noise. As an additional experiment, an artificial input offset of 50 mV along with a peak-to-peak power supply noise of 25 mV at the frequency of maximum sensitivity was applied to the circuit. Simultaneously a 5 GHz clock exhibiting 25 mV of common-mode noise at the frequency of maximum sensitivity was passed through the filter and the simulated peak-to-peak output jitter was observed to be 2.043 ps.

By integrating the simulated thermal noise at the output to derive an equivalent rms noise level, and following the discussed approach of scaling the rms noise level by the inverse of the signal slewrate to approximate the rms value of the jitter, the anticipated RJ generated by the circuit was 40.49 fs.

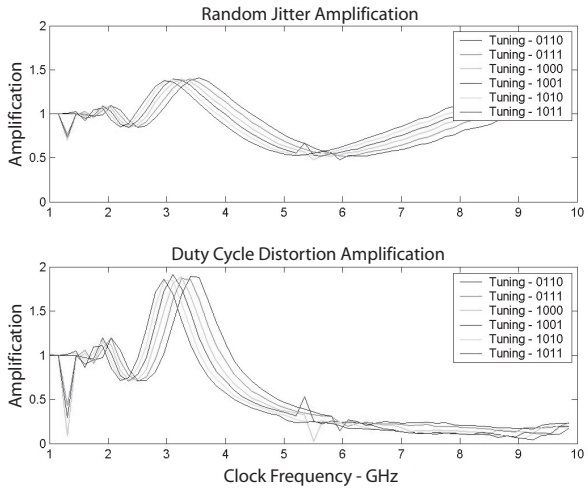
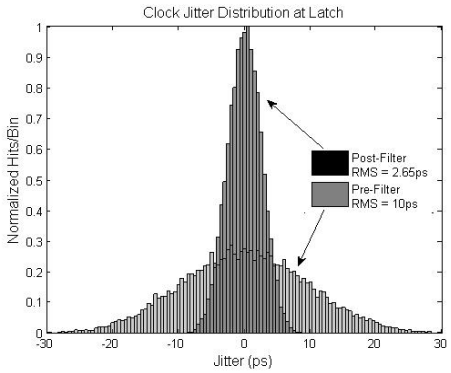
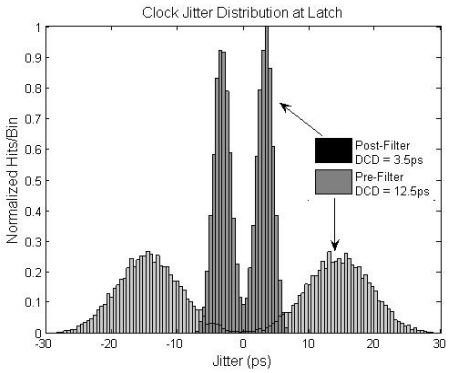


Figure 7.15: Simulated jitter amplification versus filter center frequency tuning.

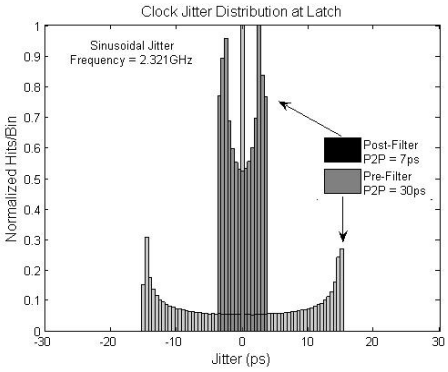
Jitter amplification was also studied. Fig. 7.15 presents the jitter amplification of the circuit for several input clock frequencies, with each curve corresponding to a distinct 4-bit center frequency tuning setting. When the filter center frequency is tuned to 5 GHz, the peak jitter amplification around 3 GHz, as seen in the diagram, results from an amplification of the clock's second harmonic through the filtering



(a) RJ Attenuation



(b) DCD Attenuation



(c) Sinusoidal Jitter Attenuation

Figure 7.16: Simulated impact of the proposed bandpass filter on various clock jitter components. (a) Gaussian distributed RJ. (b) DCD. (c) Sinusoidal jitter.

process. Conversely, when the clock frequency equals the filter's center frequency of 5 GHz, the random jitter and DCD amplification are predicted to be 0.45-0.5 and 0.25 respectively.

Fig. 7.16 illustrates the impact of the bandpass filter on specific components of the overall clock jitter. In Fig. 7.16a the bandpass filter is shown to reduce the rms level of the RJ by a factor of 3.77. In a similar way, Fig. 7.16b and Fig. 7.16c

Table 7.2: Simulated Filter Characteristics and Performance

Feature	**Value**
Center Frequency (f_c)	5 GHz
Power Dissipation	5.695 mW
Gain at f_c	7.924 dB
Tuning Range	3.8-8.53 GHz
On-chip Spiral Inductor Value	1.92 nH
Inductor Dimensions	85 μm $\times$ 85 μm
Inductor Quality Factor	4.38
Total Filter Quality Factor	2.622
RJ Jitter Amplification Factor	0.45
DCD Jitter Amplification Factor	0.25
Jitter Generation	40.69 fs
Common-mode Noise Sensitivity	30 fs/mV
Power Supply Sensitivity	6 fs/mV

show the filter reducing peak-to-peak DCD and sinusoidal jitter at a given frequency by factors of 3.57 and 4.29, respectively.

Table 7.2 presents the final characteristics of the bandpass filter and Table 7.3 compares the filter performance with on-chip bandpass filter designs previously published. According to the data reported in Table 7.3, the design presented here, while claiming the lowest Q factor, provides the highest center frequency and a wider tuning range than any of the previous designs with minimal power consumption. The low Q factor is not considered a negative quality as it was initially predicted that a bandpass filter exhibiting a Q of 2-5 would be very effective in reducing jitter, a fact that was corroborated by the many simulation results presented.

Table 7.3: Comparison of Filter Performance with Previously Published Work

Reference	**Technology**	f_C **Tuning**	**Filter Q**	V_{DD}	I_{Supply}
[133]	Bipolar	1 GHz	4-400	5 V	13.6 mA
[134]	SiGe	1.6-2 GHz	3-350	2.8 V	8.7 mA
[135]	SiGe	1.882 GHz	12.5467	2.7 V	18 mA
[136]	CMOS	194-203 MHz	2.3-∞	3V	2.94 mA
[137]	CMOS	829.6 MHz	3.4-629	2 V	22.9 mA
[138]	CMOS	850 MHz	47.2222	2.7 V	77 mA
[139]	CMOS	2.14 GHz	35.6667	2.5 V	2 mA
[140]	CMOS	2.19 GHz	40	1.3 V	4 mA
This Work	CMOS	3.8-8.53 GHz	2.62	1.2 V	5.695 mA

Chapter 8

Conclusion

This dissertation represents the culmination of an effort to provide practical yet novel enhancements to "state of the art" high-speed electrical signaling. With few exceptions, the proposals detailed here have either been published in respected engineering journals [115, 141], been presented at international conferences [120, 142], or have led to patent filings [143, 144]. In addition, two corresponding journal papers are currently in review [61, 145], four patent disclosures have been approved for filing, and at least one more potential article is awaiting submission.

As chip-to-chip signaling reaches and exceeds the physical bandwidth of the commodity PC board channel, interconnect designers not only face the challenge of restoring signal integrity through noise filtering and channel equalization, but must also address the difficulties associated with modeling the growing impact of timing jitter. This dissertation addresses both challenges by providing algorithms for generating jittery signals with statistical precision, new algorithms for channel equalizer calibration, and the design of an LC bandpass forwarded clock filter.

8.1 Summary of Contributions

1. **Methods for generating realistic clock and data waveforms with statistically definable jitter characteristics.**

   The steady rise in chip-to-chip signaling frequency has turned the focus of signal integrity from the vertical data eye-closing effects of ISI and other amplitudinal noise sources to the horizontal eye closure associated with timing uncertainty or jitter. As a result, recent signal integrity publications have focused on jitter,

while failing to recognize or account for the synergistic way in which noise and jitter cooperate to close high frequency data eyes along both dimensions. Overcoming the tendency to model and simulate noise and jitter independently required the capability to generate signals with simultaneous voltage and timing degradation.

A technique is presented here which provides the needed functionality through applying Fourier theory to the signal generation problem. Both periodic clock signals and aperiodic data signals may be derived with complete control over the noise and jitter characteristics of the waveforms. By constructing the waveforms from their respective frequency components, sub-femptosecond jitter resolution is achieved even when the simulation time step is several orders of magnitude larger.

A second pair of signal generation algorithms was also developed to overcome the constraints on data jitter magnitude inherent in the first approach, while significantly increasing the speed of the signal generation process, at the admitted cost of flexibility.

Both methodologies allow the derived signal to exhibit any combination of statistical characteristics. To combine the precision of these realistic signals with transistor level simulation requires only a few additional lines of code to write the time versus voltage waveform values to the appropriate format of the target simulation engine. By so doing, timing critical circuits may be more readily identified, characterized, and compensated for to produce more robust designs.

2. **Simple and novel methods for calibrating continuous-time data channel equalizers.**

   The study of channel equalization has been motivated for decades by the inevitable clash between performance demands and available channel bandwidth. To compensate for both manufacturing tolerance and environment changes over time, most equalizers require a corresponding calibration scheme. Continuous-time equalizers are particularly sensitive to environmental changes, and are

unfortunately difficult to tune. It is not uncommon for more complexity, on-chip area, and power draw to be associated with the adaptation circuitry than with the equalizer itself.

A new calibration scheme, initially targeting second-order continuous-time equalizers, has been presented, wherein two new error terms, derived from the measured or simulated pulse response of the channel during a training sequence, are used to calibrate and fix the equalizer coefficients. The results show dramatic improvement in data eye quality following equalization, which in turn translates to higher achievable datarates for a given channel and BER specification.

3. **The design of a high frequency, tunable bandpass forwarded clock filter.**

   Past and present approaches to reducing clock jitter in digital communication systems have typically relied on the inherent high frequency jitter filtering provided by PLLs. Unfortunately, the assumption that incorporating a PLL into the clock path will enhance signal integrity is not an absolute, as the jitter filtering characteristics of the PLL may reduce the correlation between jitter events initially common to both the data signals and their associated forwarded sampling clocks. In addition, the PLL may actually contribute more uncorrelated jitter to the passing clock than it removes, due to oscillator phase noise and supply noise sensitivity.

   In some cases, the use of a relatively low-Q bandpass filter may serve to reduce clock jitter significantly without the complexity of the PLL, as the slow transient response of such filters has an averaging effect on the incoming edge timing, reducing both random and high frequency deterministic jitter components. To verify these assumptions, a fully differential, tunable bandpass filter with on-chip spiral inductors was designed. Compared with previously published designs, the current approach draws minimal power while achieving both a high center frequency and a wider tuning range. In addition to suppressing

RJ and PJ, as anticipated, the filter proved effective in reducing the DC and high frequency components of DCD.

8.2 Areas of Future Interest

Without detracting from the value of the contributions just discussed, there are several ways in which the material presented in this thesis might be built upon, through extension to other applications, enhancements, etc.

- **Waveform Generation**

  While the signal generation techniques presented here provide functionality currently unavailable in industry standard tools, several enhancements to the methods could be made. First of all, the generality of the methods can be improved by extending the models to alternative modes of signal encoding. While 2-PAM encoding is the standard for high-speed electrical interconnects, other signaling methodologies exist and some, including multilevel pulse amplitude modulation (M-PAM) are popular. The methods may also be extended to RF communication systems by addressing phase-shift keying (PSK) and other forms of signal modulation.

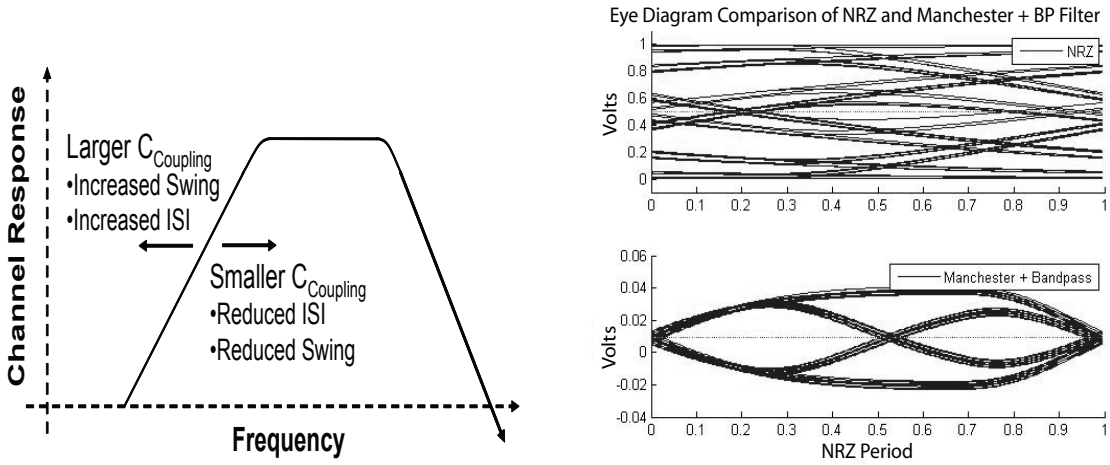
  As was discussed earlier, the value of the Fourier-based signal generation methodology may also be improved by incorporating a more realistic waveform into the underlying model. Exponential functions may be built into the derivation to round the corners of the signal in an effort to more realistically model the RC and RLC filtering experienced by the signal and reduce the number of harmonic computations.

  In addition, while the current waveform generation process introduces voltage noise and jitter independently, counting on noise-to-jitter correlation to develop as the signal passes through the various blocks of the system, it should be possible to build a controlled level of noise-to-jitter correlation directly into the model.

Again as was suggested within the text, the ability to simultaneously vary the characteristics of the generated signals on a cycle-to-cycle basis may allow for more accurate modeling of oscillators and PLLs. One of the challenges associated with PLL modeling is to account for the random walk in output phase that occurs during the pause between feedback-controlled phase adjustments. Using the Fourier-based signal generation technique, it should be possible to let the signal transition timing vary with each cycle, based on a specified variance, and periodically apply a control signal to zero out the absolute phase offset. This would allow for jitter peaking and other important PLL characteristics to be simulated and not just discussed in terms of jitter transfer functions.

- **Channel Equalization**

  As high-speed data communication becomes the standard, rather than the goal, channel equalization will likely become a common part of every chip-to-chip interconnect. While the theory of equalization is well understood and architectures are mature, there is always room for improvement. One of the best forms of improvement is simplification. If the method of tuning an equalizer's frequency response based on single pulse and double pulse amplitude measurements, as presented here, can be extended to other equalizer topologies, then the ominous task of equalizer realization may be alleviated. Specifically, it would be interesting to study the extension of the calibration algorithms presented here to analog discrete-time FIR-based equalizers and continuous-time equalizers of higher order. It may also be prudent to study the response of the proposed tuning methods to simultaneous classical adaptation of discrete-time transmit equalizer circuits to see if the two distinct methods interfere with each other during training. And finally, while the methodology presented here did not specifically provide for dynamic or continuous adaptation, it may be possible to employ either data encoding or the time-multiplexing of training patterns into the data path to facilitate continuous equalizer recalibration and account for changing environmental conditions without degrading link throughput.



(a) AC Coupled Interconnect

(b) NRZ vs. Manchester + BP

Figure 8.1: (a) Impact of narrowband filtering broadband data signals. (b) Simulated eye diagrams of band-limited NRZ data and Manchester encoded data followed by a bandpass filter.

- **Bandpass Clock Filtering**

  For years DLLs have been considered less attractive than PLLs in high-speed clock distribution networks as DLLs offer no jitter filtering. Yet at high datarates, the allpass jitter transfer of DLLs may provide better link performance, as clock-to-data jitter correlation is maintained. Based on a preliminary study, it may be possible to balance the trade-offs between jitter matching and jitter filtering by following the DLL with a bandpass filter. If designed correctly, this combination will provide a level of noise and jitter filtering comparable to that of the PLL, while maintaining a greater degree of correlation between jitter common to both clock and data signals.

  An additional area of interest is the application of bandpass filtering to data channels. Intuitively, such filtering of the broadband data would be destructive, but based on an earlier study of AC coupled interconnects [146], intentional

attenuation of low frequency signal content can be exploited to reduce ISI. Fig. 8.1a summarizes the findings of the previous study. According to the figure, a reduction in coupling capacitance shifts the frequency zero of the system transfer function to higher frequencies, the result of which is a simultaneous reducing in ISI and signal swing. To regain lost signal power, the capacitor may be increased, thereby shifting the zero lower, at the cost of additional ISI.

A preliminary study has shown that broadband data may be narrowbanded through data encoding techniques, such that passage through a narrowband filter should only attenuate out-of-band noise and ISI, as previously observed, rather than the low frequency components of the signal. Manchester encoding is one method for narrowbanding the data, but leads to a 2× reduction in datarate for a given clock frequency. Still, as demonstrated in Fig. 8.1b, by bandpass filtering Manchester encoded data at the receiving end of a band-limited channel, the data eyes are well defined (lower window), while a corresponding NRZ encoded signal, at half the frequency, exhibits a tremendous amount of ISI (upper window). Alternative narrowband encoding techniques exist, and new encoding may be developed, to facilitate the application of bandpass filtering to data channels while minimizing the impact on link throughput.

If adopted, the technology presented within this work, as well as the pursuit of the areas suggested for further consideration, should enhance the performance and robustness of developing interconnect systems and extend the life of electrical signaling.

Bibliography

[1] G. Papadopoulus, "Future of computing, NOW Workshop", Lake Tahoe, California, July 27, 1997.

[2] "Video game industry", RocSearch, Ltd., [Online], Available: http://www.rocsearch.com/pdf/Video%20Game%20Industry.pdf, [Accessed: July 14, 2006].

[3] G. Papadopoulus, "Keynote Address NC03Q4 at SunNetwork", Berlin, Germany, December 4, 2004].

[4] Agilent Technologies Technical Staff, ", Question and Answer Period following the Seminar on Signal Integrity Solutions for High Speed Design, Boise, ID, May, 2006.

[5] C. Werner, C. Hoyer, A. Ho, M. Jeeradit, F. Chen, B. Garlepp, W. Stonecypher, S. Li, A. Bansal, A. Agarwal, E. Alon, V. Stojanovic, and J. Zerbe, "Modeling, simulation, and design of a multi-mode 2-10 Gb/sec fully adaptive serial link system", in *Proceedings of the IEEE Custom Integrated Circuits Conference*, September 2005, pp. 709–716.

[6] "Press Release: Micron introduces the worlds first 1 Gigabit DDR3 memory for computing applications", [Online], Available: http://www.micron.com/about/news/pressrelease.aspx?id= 3DEDAAC1EFA2B68E, [Accessed: January 14, 2007].

[7] M. M. K. Liu, *Principles and Applications of Optical Communications*, Irwin Professional Publishing, Chicago, first edition, 1996.

[8] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*, Cambridge University Press, New York, first edition, 1998.

[9] J. F. Buckwalter, *Deterministic Jitter in Broadband Communications*, PhD thesis, January 2005.

[10] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, third edition, 1995.

[11] B. K. Casper, M. Haycock, and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling scheme", in *Digest of Technical Papers from the IEEE Symposium on VLSI Circuits*, June 2002, pp. 54–57.

[12] C. E. Shannon, "A mathematical theory of communication", *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[13] C. E. Shannon, "A mathematical theory of communication", *The Bell System Technical Journal*, vol. 27, pp. 623–656, October 1948.

[14] J. Buckwalter and A. Hajimiri, "Crosstalk-induced jitter equalization", in *Proceedings of the IEEE Custom Integrated Circuits Conference*, September 2005, pp. 409–412.

[15] J. Buckwalter and A. Hajimiri, "Cancellation of crosstalk-induced jitter", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 621–632, March 2006.

[16] C. T. Chen, J. Zhao, and Q. Chen, "A simulation study of simultaneous switching noise", in *Proceedings of the IEEE Electronic Components and Technology Conference*, May 2001, pp. 1102–1106.

[17] Y. J. Kim, S. W. Han, K. W. Park, J. K. Wee, and J. S. Kih, "Analysis of simultaneous switching noise by short-circuit current and CMOS-single ended driver", in *Proceedings of the IEEE Electronic Components and Technology Conference*, May 2005, pp. 1748–1751.

[18] C. S. Choy, C. F. Chan, and M. H. Ku, "A feedback control circuit design technique to suppress power noise in high speed output driver", in *Proceedings of the IEEE*, 1995, pp. 307–310.

[19] R. Senthinathan, J. L. Prince, and S. Nimmagadda, "Effects of skewing CMOS output driver switching on the "simultaneous" switching noise", in *Proceedings of the IEEE/CHMT Electronics Manufacturing Technology Symposium*, 1991September, pp. 342–345.

[20] R. Senthinathan and J. L. Prince, "Application specific CMOS output driver circuit design techniques to reduce simultaneous switching noise", *IEEE Journal of Solid-State Circuits*, vol. 28, no. 12, pp. 1383–1388, December 1993.

[21] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*, John Wiley & Sons, Inc., New York, first edition, 1997.

[22] R. Hartley, "Base materials for high speed, high frequency pc boards", 2002, [Online], Available: http://www.speedingedge.com/PDF-Files/Materials_RickH2.pdf, [Accessed: November 25, 2006].

[23] M. Li, Y. Tao, S. Wang, and T. Kwasniewski, "Studies on FIR filter pre-emphasis for high-speed backplane data transmission", Carleton University, Ottawa, and the Altera Corporation.

[24] R. W. Lucky, L. N. Holzman, F. K. Becker, and E. Port, "Automatic equalization for digital communication", *Proceedings of the IEEE (Correspondence)*, vol. 53, pp. 96–97, January 1965.

[25] S. Reynolds, P. Pepeljugoski, J. Schaub, J. Tierno, and D. Beisser, "A 7-tap transverse analog-FIR filter in 0.13 μm CMOS for equalization of 10 Gb/s fiber-optic data systems", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2005, pp. 330–331.

[26] M. E. Said, J. Sitch, and M. Elmasry, "A 0.5 μm SiGe pre-equalizer for 10 Gb/s single-mode fiber optic links", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2005, pp. 224–225.

[27] B. Casper, J. Jaussi, F. O'Mahony, M. Mansuri, K. Canagasaby, J. Kennedy, E. Yeung, and R. Mooney, "A 20 Gb/s forwarded clock transceiver in 90nm CMOS", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2006.

[28] J. Jaussi, B. Casper, M. Mansuri, F. O'Mahony, K. Canagasaby, J. Kennedy, and R. Mooney, "A 20 Gb/s embedded clock transceiver in 90nm CMOS", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2006.

[29] J. E. C. Brown and P. J. Hurst, "Continuous-time forward equalization for the decision-feedback-equalizer-based read channel", *IEEE Transactions on Magnetics*, vol. 34, no. 4, pp. 2372–2381, July 1998.

[30] J. E. C. Brown, P. J. Hurst, B. C. Rothenberg, and S. H. Lewis, "A CMOS adaptive continuous-time forward equalizer, LPF, and RAM-DFE for magnetic recording", *IEEE Journal of Solid-State Circuits*, vol. 34, no. 2, pp. 162–169, February 1999.

[31] J. W. M. Bergmans, "Digital magnetic recording systems", *IEEE Transactions on Magnetics*, vol. 24, no. 1, pp. 683–688, January 1988.

[32] R. S. Kajley, J. E. C. Brown, and P. J. Hurst, "A mixed-signal decision-feedback equalizer that uses a look-ahead architecture", *IEEE Journal of Solid-State Circuits*, vol. 32, no. 3, pp. 450–459, March 1997.

[33] Wavecrest Technical Staff, "Jitter fundamentals", 2003, [Online], Available: http://www.wavecrest.com/technical/jitterfund.htm, [Accessed: October 1, 2004].

[34] Agilent Technologies Technical Staff, "Appl. Note 1448-1, measuring jitter in digital systems", [Online], Available: http://cp.literature.agilent.com/litweb/pdf/5988-9109EN.pdf, [Accessed: October 1, 2004].

[35] "Understanding and characterizing timing jitter", [Online], Available: http://www2.tek.com/cmswpt/tidetails.lotr?ct=TI&cs=Primer&ci=2244&lc=EN, [Accessed: January 1, 2007], Tektronix Technical Staff.

[36] Agilent Technologies Technical Staff, "Appl. Note 1448-1, jitter analysis: The dual-Dirac model, RJ/DJ, and Q-SCALE", [Online], Available: http://cp.literature.agilent.com/litweb/pdf/5989-3206EN.pdf, [Accessed: Januaryagilent dual-Di 1, 2007].

[37] B. Sklar, *Digital Communications - Fundamentals and Applications*, Prentice Hall, New Jersey, first edition, 1988.

[38] Q. Dou and J. A. Abraham, "Jitter decomposition by time lag correlation", in *Proceedings of the IEEE International Symposium on Quality Electronic Design*, March 2006, pp. 525–530.

[39] J. Buckwalter, B. Analui, and A. Hajimiri, "Predicting data-dependent jitter", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 9, pp. 453–457, September 2004.

[40] B. Analui, J. Buckwalter, and A. Hajimiri, "Estimating data-dependent jitter of a general LTI system from step response", in *Proceedings of the IEEE*, 2005, pp. 1841–1844.

[41] B. Analui, J. Buckwalter, and A. Hajimiri, "Data-dependent jitter in serial communications", *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 11, pp. 3388–3397, November 2005.

[42] J. Buckwalter and A. Hajimiri, "A 10 Gb/s data-dependent jitter equalizer", in *Proceedings of the IEEE Custom Integrated Circuits Conference*, September 2004, pp. 39–42.

[43] J. Buckwalter and A. Hajimiri, "Analysis and equalization of data-dependent jitter", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 607–619, March 2006.

[44] International Business Strategies, "Analysis of the relationship between eda expenditures and competitive positioning of ic vendors: A custom study for eda consortium", [Online], Available: http://edac.org/downloads/resources/profitability/HandelJonesReport.pdf, [Accessed: August 11, 2006].

[45] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, Inc., New York, first edition, 1996.

[46] Agilent Technologies Technical Staff, "Guide to harmonic balance simulation in ads", [Online], Available: http://eesof.tm.agilent.com/docs/adsdoc2004A/pdf/adshbapp.pdf, [Accessed: November 10, 2005].

[47] R. Telichevesky, K. Kundert, and J. White, "Receiver characterization using periodic small-signal analysis", in *Proceedings of the IEEE Custom Integrated Circuits Conference*, May 1996, pp. 449–452.

[48] "RF Design Environment closes verification gap, in *Microwaves & RF for Designers at Higher Frequencies*", [Online], Available: http://www.mwrf.com/Articles/Index.cfm?ArticleID=6854, [Accessed: November 10, 2005].

[49] C. Werner, C. Hoyer, A. Ho, M. Jeeradit, F. Chen, B. Garlepp, W. Stonecypher, S. Li, A. Bansal, A. Agarwal, E. Alon, V. Stojanovic, and J. Zerbe, "Modeling, simulation, and design of a multi-mode 2-10 Gb/sec fully adaptive serial link system", in *Proceedings of the IEEE Custom Integrated Circuits Conference*, September 2005, pp. 709–716.

[50] D. B. Leeson, "A simple model of feedback oscillator noise spectrum", in *Proceedings of the IEEE*, February 1966, vol. 54, pp. 329–330.

[51] "IEEE standard definitions of physical quantities for fundamental frequency and time metrology - random instabilities", 1139-1999, IEEE, 1999.

[52] Synapticad Technical Staff, "WaveFormer Pro", [Online], Available: http://www.syncad.com, [Accessed: August 20, 2006].

[53] G. Balamurugan and N. Shanbhag, "Modeling and mitigation of jitter in multi-Gbps source-synchronous I/O links", in *Proceedings of the 21st International Conference on Computer Design*. IEEE, October 2003, pp. 254–260.

[54] P. K. Hanumolu, B. K. Casper, R. Mooney, Gu-Yeon Wei, and Un-Ku Moon, "Analysis of PLL clock jitter in high-speed serial links", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 11, pp. 879–886, November 2003.

[55] K. K. Kim, J. Huang, Y. B. Kim, and F. Lombardi, "On the modeling and analysis of jitter in ATE using Matlab", in *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, October 2005, pp. 285–293.

[56] "Appl. Note 61W-19431-2, Controlled jitter generation for jitter tolerance and jitter transfer testing", [Online], Available: http://www.tek.com/Measurement/App_Notes/61_18431/eng/61W-18431-2.pdf, [Accessed: November 18, 2005], Tektronix Technical Staff.

[57] D. Hong and K. T. Cheng, "BER estimation for high-speed serial links", in *Proceedings of the Gigalscale Systems Research Center Annual Symposium*, September 2006.

[58] S. Tabatabaei, M. Lee, and F. B. Zeev, "Jitter generation and measurement for test of multi-Gbps serial IO", in *Proceedings of the ITC International Test Conference*, October 2004, pp. 1313–1320.

[59] P. K. Hanumolu, B. K. Casper, R. Mooney, G. Y. Wei, and U. K. Moon, "Jitter in high-speed serial and parallel links", in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 2004, pp. 425–428.

[60] A. Sanders, M. Resso, and J. D'Ambrosia, "Channel compliance testing utilizing novel statistical eye methodology", in *Proceedings of Design Con 2004*. International Engineering Consortium, February 2004.

[61] T. M. Hollis, D. J. Comer, and D. T. Comer, "Abbreviated steady state analysis - efficient simulation of high-speed clock channels", *IEEE Transactions on Circuits and Systems I*, (In Review).

[62] G. Esch and T. Chen, "Design of CMOS IO drivers with less sensitivity to process, voltage, and temperature variations", in *Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications*, January 2004, pp. 1–6.

[63] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series*, Wiley, New York, first edition, 1949.

[64] R. E. Kalman, "A new approack to linear filtering and predication problems", *Transactions of the ASME, Journal of Basic Engineering*, pp. 34–45, March 1960.

[65] L. Litwin, "Matched filtering and timing recovery in digital receivers", 2001, [Online], Available: http://www.rfdesign.com, [Accessed: March 1, 2006].

[66] D. G. Long, "Matched filter discussions", Brigham Young University, 2006, private communication.

[67] T. E. Tuncer, "ISI-free pulse shaping filters for receivers with or without a matched filter", *Proceedings of the IEEE*, pp. III2269–III2272, 2002.

[68] A. Kisel, "An extension of pulse shaping filter theory", *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 645–647, May 1999.

[69] A. Kisel, "Nyquist 1 universal filters", *IEEE Transactions on Communications*, vol. 48, no. 7, pp. 1095–1099, July 2000.

[70] N. Alagha and P. Kabal, "Generalized raised-cosine filters", *IEEE Transactions on Communications*, vol. 47, no. 7, pp. 989–997, July 1999.

[71] C. Tan and N. Beaulieu, "An investigation of transmission properties of Xia pulses", in *Proceedings of the IEEE International Conference on Communications*, Vancouver, British Columbia, June 6–10 1999, pp. 1197–1201.

[72] C. Tan and N. Beaulieu, "Transmission properties of conjugate-root pulses", *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 553–558, April 2004.

[73] S. Kesler and D. Taylor, "Research and evaluation of the performance of digital modulations in satellite communications systems", Tech. Rep. CRL Report No. 92, McMaster University, Hamilton, Ontario, Canada, 1981.

[74] H. Baher and J. Beneat, "Design of analog and digital data transmission filters", *IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications*, vol. 40, no. 7, pp. 449–460, July 1993.

[75] E. Hassan and H. Ragheb, "Design of linear phase Nyquist filters", *IEE Proceedings – Circuits, Devices and Systems*, vol. 143, no. 3, pp. 139–142, June 1996.

[76] S. Mneina and G. Martens, "Maximally flat delay Nyquist pulse design", *IEEE Transactions on Circuits and Systems – II: Express Briefs*, vol. 51, no. 6, pp. 294–298, June 2004.

[77] P. Chiang, W. J. Dally, M. E. Lee, R. Senthinathan, Y. Oh, and M. A. Horowitz, "A 20-Gb/s 0.13-μm CMOS serial link transmitter using an LC-PLL to directly drive the output multiplexer", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 1004–1011, April 2005.

[78] A. Ho, V. Stojanovic, F. Chen, C. Werner, G. Tsang, E. Alon, R. Kollipara, J. Zerbe, and M. A. Horowitz, "Common-mode backchannel signaling system for differential high-speed links", in *Digest of Technical Papers from the IEEE Symposium on VLSI Circuits*, June 2004, pp. 352–355.

[79] G. Sheets and J. D'Ambrosia, "The impact of environmental conditions on channel performance", in *Proceedings of DesignCon 2004*. International Engineering Consortium, February 2004.

[80] J. Zerbe, Q. Lin, V. Stojanovic, A. Ho, R. Kollipara, F. Lambrecht, and C. Werner, "Comparison of adaptive and non-adaptive equalization methods in high-performance backplanes", in *Proceedings of DesignCon 2004*, February 2004.

[81] J. E. Jaussi, G. Balamurugan, D. R. Johnson, B. K. Casper, A. Martin, J. Kennedy, N. Shanbhag, and R. Mooney, "8-Gb/s source-synchronous I/O link with adaptive receiver equalization, offset cancellation, and clock de-skew", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 80–88, January 2005.

[82] C. Langton, "Inter symbol interference (ISI) and raised cosine filtering", 2006, [Online], Available: http://www.complextoreal.com, [Accessed: March 1, 2006].

[83] O. J. Zobel, "Electrical network and method of transmitting electrical currents", US Patent #1,603,305, 1926.

[84] H. W. Bode, "Attenuation equalizer", US Patent #2,096,027, 1936.

[85] R. W. Lucky and H. R. Rudin, "Generalized automatic equalization for communication channels", *Proceedings of the IEEE (Letters)*, pp. 439–440, March 1966.

[86] M. E. Austin, "Decision-feedback equalization for digital communication over dispersive channels", Tech. Rep. 461, Massechusetts Institute of Technology: Research Laboraory of Electronics, August 1967.

[87] M. Sorna, T. Beukema, K. Selander, S. Zier, B. J, P. Murfet, J. Mason, W. Rhee, H. Ainspan, and B. Parker, "A 6.4 Gb/s CMOS SerDes core with feedforward and decision-feedback equalization", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2005, pp. 62–63.

[88] R. Payne, B. Bhakta, S. Ramaswamy, S. Wu, J. Powers, P. Landman, U. Erdogan, A. Yee, R. Gu, L. Wu, Y. Xie, B. Parthasarathy, K. Brouse, W. Mohammed, K. Heragu, V. Gupta, L. Dyson, and W. Lee, "A 6.25 Gb/s binary adaptive DFE with first post-cursor tap cancellation for serial backplane communications", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2005, pp. 68–69.

[89] J. Freedman and J. Margolin, "Signal-to-noise improvement through integration in a delay-line filter system", Tech. Rep. 22, Massechusetts Institute of Technology: Lincoln Laboraory, May 1953.

[90] J. H. Winters and S. Kasturia, "Adaptive nonlinear cancellation for high-speed fiber-optic systems", *IEEE Journal of Lightwave Technology*, vol. 10, no. 7, pp. 971–977, July 1992.

[91] V. Stojanovic, A. Ho, B. Garlepp, F. Chen, J. Wei, E. Alon, C. Werner, J. Zerbe, and M. A. Horowitz, "Adaptive equalization and data recovery in a dual-mode (PAM2/4) serial link transceiver", *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, pp. 348–351, June 2004.

[92] J. H. Winters and S. Kasturia, "A multigigabit backplane transceiver core in -.13-μm CMOS with a power-efficient equalization architecture", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2658–2666, December 2005.

[93] J. H. Winters and R. D. Gitlin, "Electrical signal processing techniques in long-haul fiber-optic systems", *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1439–1453, September 1990.

[94] D.J. Comer, *Active and Passive Filters*, Brigham Young University, Utah, 2004.

[95] M. Banu and Y. Tsividis, "Fully integrated active RC filters in MOS technology", *IEEE Journal of Solid-State Circuits*, vol. SC-18, no. 6, pp. 644–651, December 1983.

[96] Y. Tsividis, M. Banu, and J. Khoury, "Continuous-time MOSFET-C filters in VLSI", *IEEE Journal of Solid-State Circuits*, vol. CAS-33, no. 2, pp. 125–140, February 1986.

[97] "Designing a simple, small, wide-band and low-power equalizer for FR4 copper links", Tech. Rep. HFTA-06.0, Maxim Integrated Products, February 2003.

[98] C. D. Holdenried, J. W. Haslett, and M. W. Lynch, "Analysis and design of HBT Cherry-Hooper amplifiers with emitter-follower feedback for optical communications", *IEEE Journal of Solid-State Circuits*, vol. 39, no. 11, pp. 1959–1967, November 2004.

[99] A. J. Baker, "An adaptive cable equalizer for serial digital videl rates to 400Mb/s", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 1996, pp. 174–175.

[100] J. C. Park and L. R. Carley, "High-speed CMOS continuous-time complex graphic equalizer for magnetic recording", *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 427–437, March 1998.

[101] Y. Tomita, M. Kibune, J. Ogawa, W. W. Walker, H. Tamura, and T. Kuroda, "A 10-Gb/s receiver with series equalizer and on-chip ISI monitor in 0.11-μm CMOS", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 986–993, April 2005.

[102] H. Higashi, S. Masaki, M. Kibune, S. Matsubara, T. Chiba, Y. Doi, H. Yamaguchi, H. Ishida, K. Gotoh, and H. Tamura, "A 5-6.4-Gb/s 12-channel transceiver with pre-emphasis and equalization", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 978–985, April 2005.

[103] J. Sewter and A. C. Carusone, "A CMOS finite impulse response filter with a crossover traveling wave topology for equalization up to 30 Gb/s", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 909–917, April 2006.

[104] R. Sun, J. Park, F. O'Mahony, and C. P. Yue, "A low-power, 20-Gb/s continuous-time adaptive passive equalizer", in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 2005, pp. 920–923.

[105] F. Bien, Y. Hur, M. Maeng, H. Kim, E. Gebara, and J. Laskar, "A reconfigurable fully-integrated 0.18-μm CMOS feed-forward equalizer IC for 10-Gb/sec backplane links", in *Proceedings of the International Symposium on Circuits and Systems*. IEEE, May 2006, pp. 2117–2120.

[106] S. Ariyavisitakul N. R. Sollenberger and L. J. Greenstein, "Tap-selectable decision-feedback equalization", *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1497–1500, December 1997.

[107] J. F. Buckwalter, M. Meghelli, D. J. Friedman, and A. Hajimiri, "Phase and amplitude pre-emphasis techniques for low-power serial links", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 6, pp. 1391–1399, June 2006.

[108] J. H. R. Schrader, E. A. M. Klumperink, J. L. Visschers, and B. Nauta, "Pulse-width modulation pre-emphasis applied in a wireline transmitter, achieving 33 dB loss compensation at 5-Gb/s in 0.13-μm CMOS", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 990–999, April 2006.

[109] B. Analui, A. Rylyakov, S. Rylov, M. Meghelli, and A. Hajimiri, "A 10-Gb/s two-dimensional eye-opening monitor in 0.13-μm standard CMOS", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2689–2699, December 2005.

[110] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications*, John Wiley & Sons, Inc., Singapore, first edition, 1998.

[111] E. A. Vittoz, *Micropower Techniques*, Design of MOS VLSI Circuits for Telecommunications. Prentice Hall, 1994.

[112] D. J. Comer and D. T. Comer, "Using the weak inversion region to optimize input stage design of op amps", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 51, no. 1, pp. 8–14, January 2004.

[113] D. J. Comer and D. T. Comer, "Operation of analog MOS circuits in the weak or moderate inversion region", *IEEE Transactions on Education*, vol. 47, no. 4, pp. 430–435, November 2004.

[114] D. M. Binkley, C. E. Hopper, S. D. Tucker, B. C. Moss, J. M. Rochelle, and D. P. Foty, "A cad methodology for optimizing transistor current and sizing in analog CMOS design", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 22, no. 2, pp. 225–237, February 2003.

[115] T. M. Hollis, D. J. Comer, and D. T. Comer, "Optimization of MOS amplifier performance through channel length and inversion level selection", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 9, pp. 545–549, September 2005.

[116] R. P. Sallen and E. L. Key, "A practical method for designing RC active filters", *IRE Transactions on Circuit Theory*, vol. CT-2, pp. 74–85, March 1955.

[117] D. J. Comer and J. E. McDermid, "Inductorless bandpass characteristics using all-pass networks", *IEEE Transactions on Circuit Theory*, vol. CT-17, pp. 501–503, December 1968.

[118] C. D. Holdenried, J. W. Haslett, and M. W. Lynch, "Analysis and design of HBT Cherry-Hooper amplifiers with emitter-follower feedback for optical communications", *IEEE Journal of Solid-State Circuits*, vol. 39, no. 11, pp. 1959–1967, November.

[119] Y. Chang, J. Choma, and J. Wills, "The design of CMOS gigahertz-band continuous-time active lowpass filters with Q-enhancement circuits", in *Proceedings of the Ninth Great Lakes Symposium on VLSI*. IEEE, March 1999, pp. 358–361.

[120] T. M. Hollis, D. J. Comer, and D. T. Comer, "Self-calibrating continuous-time equalization targeting inter-symbol interference", in *Proceedings of the IEEE Northeast Workshop on Circuits and Systems*, June 2006.

[121] J. Kim, J. Yang, S. Byun, H. Jun, J. Park, C. S. G. Conroy, and B. Kim, "A four-channel 3.125 Gb/s/ch CMOS serial-link transceiver with a mixed-mode adaptive equalizer", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 2, pp. 462–471, February 2005.

[122] D. T. Comer, "VCO jitter reduction with bandpass filtering", in *Electronics Letters*. IRE, January 1995, vol. 31, pp. 11–12.

[123] D. T. Comer, "Comment on "VCO jitter reduction with bandpass filtering"", in *Electronics Letters*. IRE, May 1995, vol. 31, p. 848.

[124] V. Stojanovic and M. A. Horowitz, "Modeling and analysis of high-speed links", in *Proceedings of the IEEE Custom Integrated Circuits Conference*, September 2003, pp. 589–594.

[125] H. Ng, R. Farjad-Rad, M. J. Lee, W. J. Dally, T. Greer, J. Poulton, J. H. Edmondson, R. Rathi, and R. Senthinathan, "A second-order semidigital clock recovery circuit based on injection locking", *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2101–2110, December 2003.

[126] Y. Chang, J. Wills, and J. Choma, "A front-end filter with automatic center frequency tuning circuitry", *Proceedings of the IEEE*, pp. 28–31, 2001.

[127] H. Liu and A. I. Karsilayan, "Frequency and Q tuning of active-LC filters", *Proceedings of the IEEE*, pp. II65–II68, 2002.

[128] H. Yamazaki, K. Oishi, and K. Gotoh, "An accurate center frequency tuning scheme for 450-kHz CMOS Gm-C bandpass filters", *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1691–1697, December 1999.

[129] A. R. Holden, A. Montalvo, and R. H. Myers, "Apparatus and methods for tuning bandpass filters", US Patent #6,266,522, 2001.

[130] J. M. Khoury, "Design of a 15-MHz CMOS continuous-time filter with on-chip tuning", *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 1988–1997, December 1991.

[131] T. J. Hoffmann and M. M. Mulbrook, "Method and apparatus for automatic center frequency tuning of tunable bandpass filters", US Patent #7,039,385, 2003.

[132] J. Phinney and D. J. Perreault, "Filters with active tuning for power applications", *IEEE Transactions on Power Electronics*, vol. 18, no. 2, pp. 636–647, March 2003.

[133] W. Gao and W. M. Snelgrove, "A linear integrated LC bandpass filter with Q enhancement", *IEEE Transactions on Circuits and Systems II*, vol. 45, no. 5, pp. 635–639, May 1998.

[134] S. Pipilos, Y. P. Tsividis, J. Fenk, and Y. Papananos, "A Si 1.8GHz RLC filter with tunable center frequency and quality factor", *IEEE Journal of Solid-State Circuits*, vol. 31, no. 10, pp. 1517–1525, October 1996.

[135] D. Li and Y. P. Tsividis, "A 1.9-GHz Si active LC filter with on-chip automatic tuning", in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2001, pp. 368–369.

[136] W. B. Kuhn, F. W. Stephenson, and A. E. Riad, "A 200 MHz CMOS Q-enhanced LC bandpass filter", *IEEE Journal of Solid-State Circuits*, vol. 31, no. 8, pp. 1112–1122, August 1996.

[137] W. S. T. Yan, R. K. C. Mak, and H. C. Luong, "2-V 0.8-μm CMOS monolithic RF filter for GSM receivers", *IEEE MTT-S Microwave Symposium Digest of Technical Papers*, vol. 2, pp. 569–572, June 1999.

[138] W. B. Kuhn, N. K. Yanduru, and A. S. Wyszynski, "Q-enhanced LC bandpass filters for integrated wireless applications", *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, pp. 2577–2586, December 1998.

[139] T. Soorapanth and S. S. Wong, "A 0 dB-IL 2140 $\pm$ 30 MHz bandpass filter utilizing Q-enhanced spiral inductors in standard CMOS", *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, pp. 15–18, June 2001.

[140] F. Dulger, E. S. Sinencio, and J. S. Martinez, "A 1.3-V 5-mW fully integrated tunable bandpass filter at 2.1 GHz in 0.35-μm CMOS", *IEEE Journal of Solid-State Circuits*, vol. 38, no. 6, pp. 918–928, June 2003.

[141] T. M. Hollis, D. J. Comer, and D. T. Comer, "Mitigating ISI through self-calibrating continuous-time equalization", *IEEE Transactions on Circuits and Systems I*, vol. 53, no. 9, pp. 545–549, 2005.

[142] T. M. Hollis, D. J. Comer, and D. T. Comer, "Reduction of duty cycle distortion through bandpass filtering", in *Proceedings of the IEEE Conference on PhD Research in Microelectronics and Electronics*, July 2005, vol. 2, pp. 67–70.

[143] T. M. Hollis, "Generation and manipulation of signals for circuit and system verification", US Patent Applied For, 2006.

[144] B. K. Casper, T. M. Hollis, J. Jaussi, S. R. Mooney, F. O'Mahony, and M. Mansuri, "Forwarded clock filtering", US Patent Applied For, 2005.

[145] T. M. Hollis and D. J. Comer, "Bandpass equalization of high-speed forwarded clocks", *IEEE Transactions on Circuits and Systems I*, (In Review).

[146] L. Luo, J. M. Wilson, S. E. Mick, J. Xu, L. Zhang, and P. D. Franzon, "3 Gb/s AC coupled chip-to-chip communication using a low swing pulse receiver", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 287–296, 2006.