CHARACTERIZATION AND MODELING OF CROSSTALK BOUNDED UNCORRELATED JITTER (BUJ) FOR HIGH-SPEED INTERCONNECTS

by

Andy Kuo
B.A.Sc., University of Toronto, 2002

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

The Faculty of Graduate Studies
Department of Electrical and Computer Engineering

We accept this thesis as conforming to the required standard:

The University of British Columbia
September 2004

Library Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Andy Kuo

Name of Author *(please print)*

4/10/2004

Date (dd/mm/yyyy)

Title of Thesis: Characterization and Modeling of Crosstalk Bounded Uncorrelated Jitter (BUJ) For High-Speed Interconnects

Degree: Master of Applied Science Year: 2004

Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, BC Canada

Abstract

As data rates move towards the Gbps regime, effects that may have been ignored at lower data rates are becoming significant. Such signal integrity issues decrease the timing budget of I/O interconnects exponentially and hence, place a stringent requirement on the total jitter budget. The issues that affect signal integrity also affect jitter as both share many common root causes. Jitter can be divided into different subcomponents each with different root causes and properties. *Crosstalk Jitter*, or commonly referred in the industry as *Bounded Uncorrelated Jitter (BUJ),* is a jitter subcomponent that is mostly caused by crosstalk coupling from the adjacent interconnects on printed-circuit boards (PCB). However, the characteristics of BUJ are still ill understood. In addition, a mathematical model of jitter and an algorithm to generate a histogram for BUJ have not been developed to this date.

The crosstalk-induced pulse characteristic from an aggressor signal is studied here. Based on the superposition principle, a jitter model to calculate the time difference between the distortion-free and the distorted edge crossings was developed. This model is also extended to calculate the worst-case timing difference. In addition, algorithms to generate the histogram distributions of BUJ are also developed.

Simulation and measurement results validate the BUJ model. Algorithms developed to generate the histogram for BUJ show reasonable accuracy with four aggressor traces or less. These algorithms have fast execution times of *5~20* seconds, compared to simulation and measurement times in the range of *10~30* minutes, which require data post-processing.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I wish to thank my academic advisor, Dr. André Ivanov, for providing the opportunity and guidance to work on this interesting project throughout my Masters. From Dr. Ivanov, I learned a great deal about how to conduct and present research and learned to be self-motivated. I would also like to thank Mr. Touraj Farahmand, Dr. Roberto Rosales, and Dr. Sassan Tabatabaei for their helpful discussions and technical support in this project.

I would like to thank the following members from Dr. Ivanov's research team, namely: Nelson Ou, Baosheng Wong, James Cicalo, and A.K.M. Kamruzzaman Mollah for their helpful insights and creating a friendly research environment. I would also like to thank the following members from SOC Research Group for their support and encouragement: Andy Yan, James Wu, Jess Chia, Stephen Shang, and Marwa Hamour.

I greatly appreciate the financial support that was provided by the Vector 12 Corporation, the BC Advanced Systems Institute (ASI), Micronet R & D, the Natural Sciences and Engineering Research Council of Canada (NSERC), and Gennum Corporation. Without their support, this work would have not been possible. Special thanks to Steve Hall of Agilent Technologies for providing us access to a real-time oscilloscope at a much-needed time.

Finally, I would especially like to thank my family for their support and encouragement throughout my years of schooling, and church friends for their motivations.

Chapter 1 Introduction

1.1 Motivation

The recent deployment of high-speed I/O interconnects introduces considerable signal integrity issues [1,2]. Figure 1.1 shows an example of this by comparing an ideal 500MHz clock signal with measurements obtained by the author from a pattern generator. Moreover, as data rates move towards the Gbps region, effects that may have been ignored at lower data rates now become significant. Such signal integrity issues include ringing, reflection, electro-magnetic interference (EMI), ground bounce, switching power supply noise, thermal noise, and crosstalk. Figure 1.2 from reference [3] illustrates that as bus speeds increase over the years, the timing budget of I/O interconnects decreases exponentially. This decrease in timing budget means that signal integrity issues must be properly taken care of as the signal propagates from the transmitting to receiving end. In addition, these signal integrity issues must also be modeled with high precision and accuracy [3].

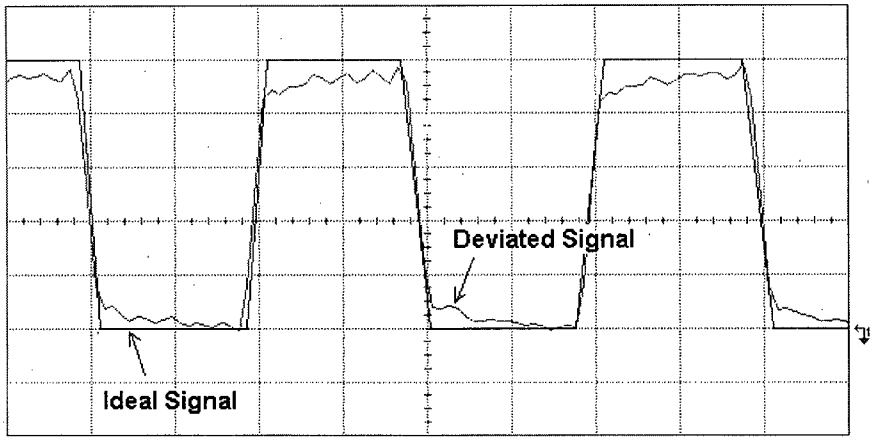


Figure 1.1: Ideal versus Measured 500MHz Signal

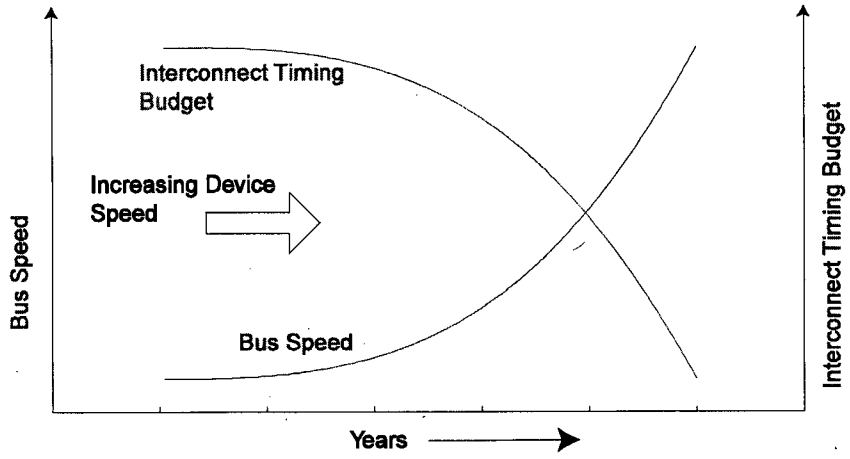


Figure 1.2: Interconnect Timing Budget and Bus Speed Trends

The decrease of the timing budget as the bus speed increases places a stringent requirement on the total jitter budget. The issues that affect signal integrity also affect jitter as both share many common root causes. For this reason, *jitter* is an important metric of signal integrity and is defined as the deviation of a timing event of a signal from its ideal occurrence in time [4]. This is as shown in Figure 1.3. Such deviation of timing events is also a stochastic process and can be represented as a histogram or probability density function (PDF) [5]. In jitter analysis, the PDF or histogram is by definition a measure of time [6], and for present high-speed applications has typically units of picoseconds. In practice, jitter can be used to estimate bit error rate (BER) since BER is essentially the cumulative distribution function (CDF) of the PDFs of the rising and falling edges of the signal [7].

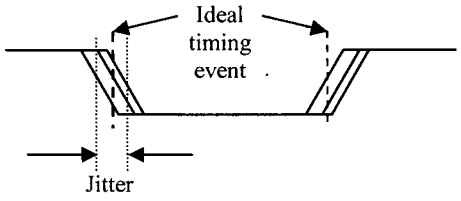


Figure 1.3: Jitter Definition

To better estimate the impact of jitter on BER performance of an I/O link, sub-components of jitter must be studied and identified. Different jitter subcomponents have different causes [8]. Understanding the characteristics and models of different jitter components allows one to devise test plans as well as improve the overall system design to reduce noise sources [9]. In addition, understanding different subcomponents can help predict the behavior of the transmission link of a system. Finally, having the ability to separate total jitter into different jitter components, i.e. de-convolution, reduces the time as well as difficulty of measuring each component directly [10].

1.2 Jitter Components

Jitter can be subdivided into two categories: *random jitter (RJ)* and *deterministic jitter (DJ)* [11, 12, 13, 14]. Figure 1.4 shows the subcomponents of total jitter [14]. The jitter specifications of a serial communication link normally specify *total jitter (TJ)* and either or both RJ and DJ. When jitter is expressed through a PDF, the PDF of the TJ is equal to the convolution of its RJ and DJ components [15].

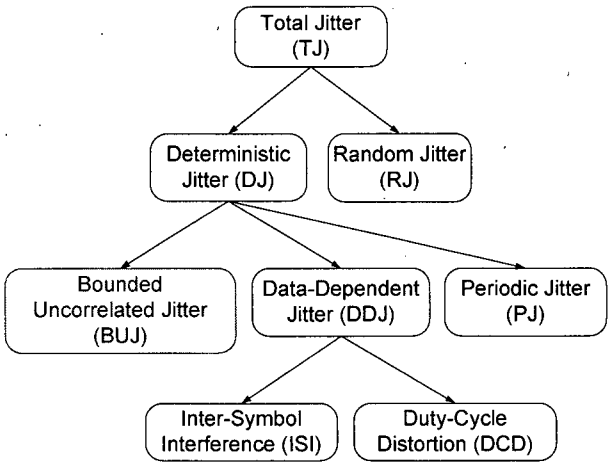


Figure 1.4: Subcomponents of Total Jitter

RJ is a random process that in theory can have any PDF shape but generally assumed to have a Gaussian distribution because thermal noise, the primary source of RJ, is also known to be Gaussian distributed [14,16, 17].

Deterministic jitter is repeatable and predictable. The peak-to-peak value of this jitter is bounded due to its predictable nature [12]. This jitter category in turn comprises the following subcomponents:

1) *Data-Dependent Jitter (DDJ)* is dependent on the bit pattern transmitted on the link under test [18]. DDJ can in turn be classified into two sub-components [18]: *Duty-Cycle Distortion (DCD)* and *Inter-Symbol Interference (ISI)*. DCD describes additional timing differences of signals having unequal pulse widths for high and low logic values. ISI is dependent on the transmitted patterns on the same trace under observation.

2) *Periodic Jitter (PJ)* refers to periodic variations of signal edge positions over time [19]. Ground bounce and other power supply variations are common PJ causes [20].

3) *Crosstalk Jitter,* also known as *Bounded Uncorrelated Jitter (BUJ)*, is typically due to coupling, e.g., from adjacent data-carrying links, or on-chip random logic switching [13]. BUJ is bounded due to finite coupling strength, and uncorrelated because there is no correlation from its own data pattern [20]. In other words, the correlation is from its adjacent traces, not the trace under study.

Mathematical models and analysis techniques are currently available to calculate each of the jitter subcomponents mentioned above, except for BUJ, the focus of this thesis. From now on, we will use the term BUJ to represent Crosstalk Jitter. However,

due to the scope of this thesis, these other techniques will not be further discussed in this report, and the reader is encouraged to read reference [20] that the author of this thesis co-authored.

Jitter analysis typically focuses on single serial data links such as *1*Gbps Ethernet or InfiniBand. Such analysis does not require knowledge of BUJ as there are no parallel data links that could introduce crosstalk. As the demand for I/O bandwidth increases, system buses are replaced with high-speed point-to-point interfaces that exceed Gbps data rates [21]. An example is the high-speed memory interface that interacts between the processor and memory dies. Such interface typically requires more I/O bandwidth, and hence multiple parallel traces between the interface and memory dies are needed. These multiple traces may introduce serious crosstalk issues and hence, BUJ needs to be studied.

Reference [22] demonstrates experimentally that BUJ is a deterministic jitter component caused by crosstalk. However, the characteristics of BUJ are still ill-understood [33]. In addition, a mathematical model of jitter and an algorithm to generate a histogram for BUJ have not been developed to this date. Such a model and an algorithm would help to predict BUJ without lengthy measurements or simulations. Furthermore, this knowledge of BUJ would empower designers to decide whether to reduce BUJ or other jitter subcomponents to achieve total jitter budget requirements [16].

This thesis characterizes BUJ through theoretical analysis of crosstalk on parallel traces in a Printed-Circuit Board (PCB) environment. A mathematical model for two traces is developed and used as the basis for algorithms capable of generating histograms for single and multiple traces. Simulation results were obtained by simulating the PCB

environment with many parallel interconnects. In addition, this thesis compares simulation results to measurement data. The results obtained include the following scenarios: different spacing between traces, different number of traces, different data patterns, different signal amplitudes and edge transition times, and timing skew differences between traces. This allows us to verify the models we developed, and validate the algorithms to generate histograms. Comparisons with simulation and measurement results validate the developed model and algorithms. In addition, the algorithms we developed can plot histograms in a period of *5~20 seconds compared to 10~30* minutes required for simulations and measurements. This allows designers to know and make quick decision on whether it is easier to reduce BUJ, or it is easier to reduce other sources of jitter to meet total jitter requirement. A study on the effect of time skew on BUJ is also carried out to explore the worst-case scenario.

1.3 Thesis Organization

Chapter 2 provides an overview of the fundamental crosstalk mechanisms. In addition, background work that explains the properties of crosstalk-induced pulses from the aggressor signal is discussed. In Chapter 3, mathematical models and algorithms of BUJ are developed. Simulation and measurement setup are explained in Chapter 4. In Chapter 5, results from simulations and measurements are compared with the models and algorithms developed. Finally, Chapter 6 presents conclusions and suggestions for future work.

Chapter 2 Background to Crosstalk and Jitter

In this chapter, we review the fundamentals of crosstalk noise on microstrip lines, from the basic concepts of self and mutual capacitance and inductance, to the prediction of the total induced crosstalk pulse on a victim trace. We also describe jitter analysis techniques.

2.1 Crosstalk Mechanisms

In a PCB environment, BUJ is a deterministic jitter component due to crosstalk from adjacent traces. By definition, *crosstalk* is the coupling of energy from one trace to another. This coupling is due to the electromagnetic fields generated by the propagating signals and its strength is dependent on the physical structure of the traces. As an example, Figure 2.1 shows the induced electric and magnetic fields around a cross section of a PCB trace. For simplicity, the substrate layer is not shown in Figure 2.1.

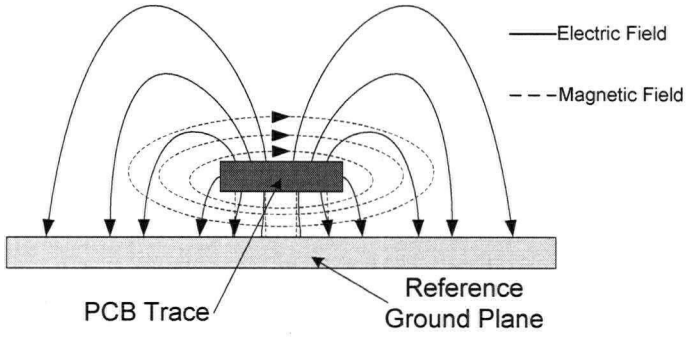


Figure 2.1: Cross Section of a PCB to Illustrate Electric and Magnetic Field [3]

Crosstalk is caused by two main effects: capacitive and inductive coupling. Before describing capacitive and inductive coupling, we need to introduce the concepts of trace capacitance, trace inductance, and crosstalk terminology.

2.1.1 Trace Capacitance

By definition, *capacitance* is the ratio of the charge present on two conductors to the electric potential between them, as expressed by the following equation:

$$C = \frac{Q}{V} \tag{2.1}$$

where Q is the charge, C is the capacitance and V is the electric potential between them. Applying Gauss' Law, the capacitance can be alternatively expressed as:

$$C = \frac{\psi}{V} \tag{2.2}$$

where ψ is the total electric flux produced by the electric potential V in a closed surface [23]. In this chapter, the capacitance equations for a microstrip and two parallel microstrip configurations will be briefly discussed.

Equations 2.1 and 2.2 can be extended to a horizontal PCB trace on a reference ground plane, also known as a microstrip configuration. This configuration is shown in Figure 2.2, where a layer of PCB substrate material is stacked between the PCB trace and the reference ground plane. This configuration is commonly used in industry for two-layer PCBs.

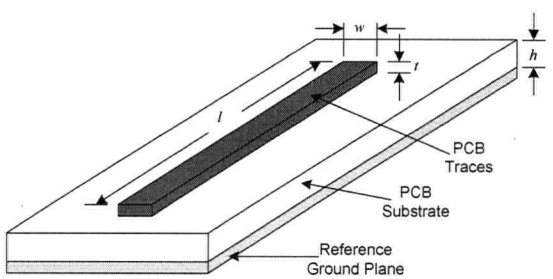


Figure 2.2: A Microstrip Configuration

Equation 2.3 can approximate the capacitance per unit length between the reference

ground plane and a rectangular horizontal trace:

$$\frac{C}{l} = \varepsilon_r \varepsilon_o K_C \left(\frac{w}{h} \right) \tag{2.3}$$

where ε_o is the dielectric constant of free space which is *1* F/m, ε_r is the effective dielectric constant of the PCB substrate, w is the width of trace, h is the vertical height of the substrate, l is the length of the conductor, and K_C is the capacitive fringing factor [24, 25]. The fringing factor is taken into account because the reference plane is assumed much wider than the width of the horizontal trace as shown in Figure 2.3. Please refer to [24, 25] for the derivation of the fringing factor. From Equation 2.3, the capacitance per unit length is proportional to the width of the trace.

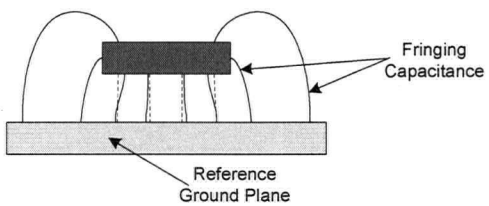


Figure 2.3: Effect of Fringing Capacitance

Besides the capacitance between the PCB trace and the reference ground plane, there is also a capacitance between two parallel traces near a ground plane as shown in Figure 2.4.

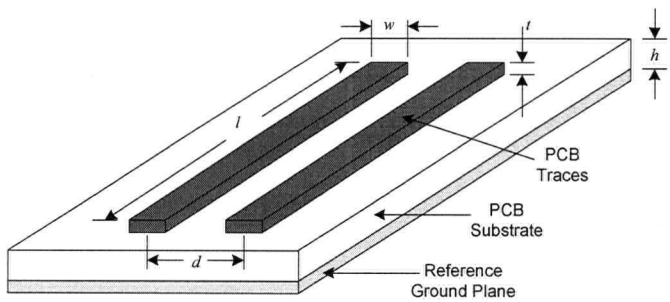


Figure 2.4: Two Microstrips Configuration

For the condition where $2h/d < 0.3$, then the capacitance per unit length equation is expressed as:

$$\frac{C_m}{l} \approx \frac{\varepsilon_r \varepsilon_o}{\pi} K_L K_C \left(\frac{w}{d}\right)^2 \tag{2.4}$$

where w, d, h and l are the dimensions specified in Figure 2.4, and K_L, K_C are the inductive and capacitive fringing factors respectively [24]. K_L in turn is also, and only, dependent on dimensions w and h. On the other hand, K_C depends on the dielectric constant of the PCB substrate. It is important to note that the capacitance per unit length is inversely proportional to the square of the distance, d, between two parallel traces.

2.1.2 Trace Inductance

Inductance relates magnetic flux to current, and can be calculated from:

$$L = \frac{N\Psi_m}{I} \tag{2.5}$$

where L is the inductance, ψ_m is the total magnetic flux due to electric current, I, flowing through the conductor, and N is the number of turns [24].

For the configuration of Figure 2.2 the inductance (self-inductance) per unit length equation is:

$$\frac{L}{l} = \frac{\mu_r \mu_o}{K_L}\left(\frac{h}{w}\right) \tag{2.6}$$

where μ_r is the relative permeability, μ_o is the magnetic permeability of free space which is $4\pi \times 10^{-7}$ H/m, K_L is the inductive fringing factor, and h and w are the dimensions specified in Figure 2.2 [24]. In this configuration, increasing the width of the trace and decreasing the height of the dielectric material, reduces the inductance per unit length

due to less magnetic fields surrounding the trace.

Figure 2.4 shows the same geometrical configuration for two parallel traces with a reference ground plane. The inductance per unit length (mutual inductance) in this configuration is essential to calculate the inductive crosstalk between circuits on the PCB. The equation for the inductance per unit length is [24]:

$$\frac{L_m}{l} \approx \frac{\mu_r \mu_o}{4\pi} \ln\left[1+\left(\frac{2h}{d}\right)^2\right] \tag{2.7}$$

2.1.3 Crosstalk Terminology

Figure 2.5 shows a typical aggressor and victim trace setup. This figure will be used to introduce crosstalk terminology. The *aggressor trace* is the trace whose transition affects the timing of the adjacent traces [26]. The *victim trace* is the trace being affected by the aggressor trace. In multiple parallel traces, there can be more than one aggressor and victim traces. In this thesis, only one victim trace is considered. The two ends of the aggressor trace are defined as the *load end* and the *source end*. The two ends of the victim trace are defined as the *far end* and the *near end*. It is also common to use the term *Far End Crosstalk (FECT)* and *Near End Crosstalk (NECT)* to represent crosstalk at the far end and near end. The capacitive and inductive couplings are commonly referred to as *mutual capacitance* and *mutual inductance*. There is also a secondary crosstalk effect. The crosstalk-induced pulse can act as an aggressor signal and induce a pulse signal back to the aggressor trace. In addition, the aggressor signal can affect another aggressor signal. When these effects are present, the coupling is referred to as *strong coupling*, and when they are absent or negligible, the coupling is referred to as *weak coupling*.

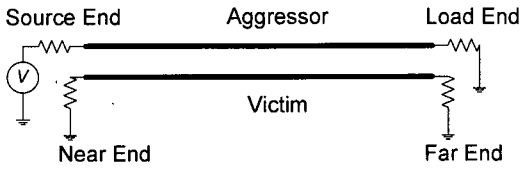


Figure 2.5: Aggressor and Victim Setup

A *quiet trace* is defined as a trace that has no signal propagating through it. The *crosstalk-induced pulse* is defined as the signal induced by the aggressor signal on the victim trace when the victim is a quiet trace. The *victim signal,* or the *distortion-free victim signal,* is the signal launched through the victim trace without any crosstalk distortion and has a notation of V_v. On the other hand, the *distorted victim signal* is the signal with crosstalk distortion. Finally, the *aggressor signal,* V_a, is the signal that propagates through the aggressor trace.

When the signal changes from one logic state to the other, it is referred to as an *edge transition.* The transition from logic '*0*' to logic '*1*' is referred to as a *rising edge transition.* Conversely, logic '*1*' to logic '*0*' is referred to as *falling edge transition.* The transition rise or fall times, from *0%* to *100%* or vice versa, are defined as *edge transition times,* T_{edge}. Note that T_{edge} is positive and that the rising edge and falling edge transitions can have different edge transition times. The time location of the signal when it crosses half of the signal amplitude value is referred to as an *edge crossing point.* This is as illustrated in Figure 2.6.

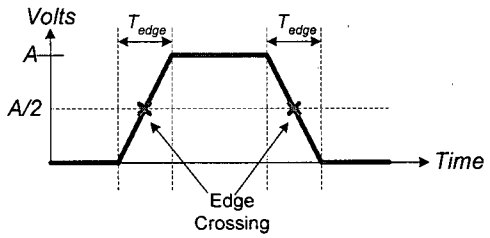


Figure 2.6: Edge Transitions

2.1.4 Mutual Capacitance and Mutual Inductance

In a PCB context, mutual capacitance is a measure of capacitive coupling between two traces. Figure 2.7 illustrates mutual capacitance per unit length between the aggressor and the victim trace.

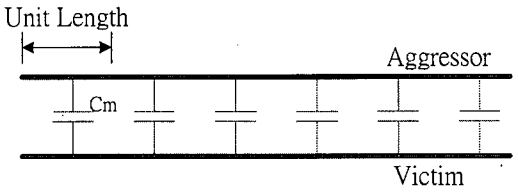


Figure 2.7: Capacitive Coupling per Unit Length

Mutual capacitance can be calculated from Equation 2.4 in subsection 2.1.1. As mentioned earlier, when two parallel traces are sufficiently close, the capacitance can become large enough to create significant coupling between the traces. This coupling can induce a current onto the victim, and can be calculated from:

$$I_{Cm} = C_m \frac{dV_a}{dt} \tag{2.8}$$

where I_{Cm} is the induced current per unit length by the aggressor trace on the victim trace, V_a is the driving voltage on the aggressor trace, and C_m is the mutual capacitance per unit length [3]. Since the induced current crosstalk is proportional to the rate of change of the voltage on the aggressor trace, as the edge transition time reduces, the current induced

onto the victim will increase.

Mutual inductance is the coupling from one trace to the other due to a magnetic field. Figure 2.8 shows the inductance per unit length.

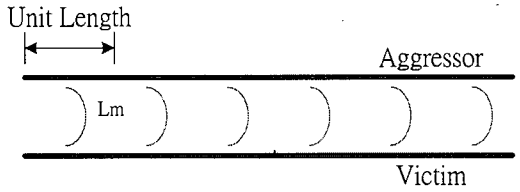


Figure 2.8: Inductive Coupling per Unit Length

Equation 2.9 predicts the crosstalk-induced voltage due to mutual inductance:

$$V_{Lm} = L_m \frac{dI_a}{dt} \tag{2.9}$$

where L_m is the mutual inductance per unit length and I_a is the driving current on the aggressor trace per unit length [3]. The induced crosstalk is proportional to the rate of change of the current on the aggressor trace. Similar to Equation 2.8, the induced voltage increases as the edge transition time reduces.

2.1.5 Crosstalk-Induced Pulse

The crosstalk-induced pulse trace travels both backward towards the near end, and forward towards the far end of the victim trace, as shown in Figure 2.9.

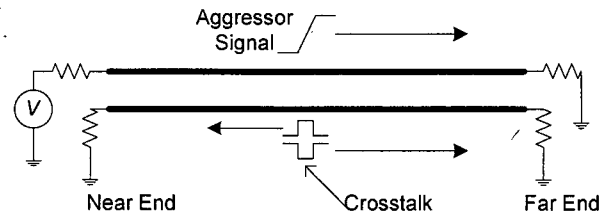


Figure 2.9: Forward and Backward Propagation

Table 1 summarizes the effect of edge transitions on the polarity of the crosstalk-induced pulse [27]. Note that at the far end, the crosstalk-induced pulses caused by the mutual capacitance and the mutual inductance have opposite polarities [27]. Since the receiver is connected to the far end, only the far end signals are studied in this work.

Edge Transition	Pulse induced only by Mutual Capacitance		Pulse induced only by Mutual Inductance	
	Polarity at the Near End	Polarity at the Far End	Polarity at the Near End	Polarity at the Far End
Rising Edge	Positive Pulse	Positive Pulse	Positive Pulse	Negative Pulse
Falling Edge	Negative Pulse	Negative Pulse	Negative Pulse	Positive Pulse

Table 1: Polarity of Crosstalk-Induced Pulses by either Capacitive or Inductive Coupling

In practice, mutual capacitance and inductance both exist. Assuming a weak-coupling system, there will be a subtraction between capacitive and inductive crosstalk on the far end due to the opposite polarities of the induced pulses. The amount of reduction depends on the magnitude of the mutual inductance and capacitance. Unfortunately, crosstalk reduction does not occur at the near end due to the identical crosstalk-induced pulse polarity.

Table 1 shows only the polarity of the crosstalk-induced pulse. Jarvis in [28] presented the equations to calculate the voltage amplitude of the crosstalk. The far end amplitude can be calculated from:

$$V_{p\_far} = -\frac{\Delta V_a l\sqrt{LC}}{2T_{edge}}\left(\frac{L_m}{L} - \frac{C_m}{C}\right) \qquad (2.10)$$

where ΔV_a is amplitude change of the aggressor signal, L_m is the mutual inductance per unit length, L is the self-inductance per unit length, C_m is the mutual capacitance per unit

length, C is the self-capacitance per unit length, and l is the length of the conductor. This equation is derived from Maxwell's Equation. It applies to PCB traces if they are properly terminated. Based on this equation, the amplitude of the crosstalk-induced pulse depends on the amplitude change of the aggressor, the edge transition time of the aggressor signal, and the capacitances and inductances. Note that ΔV_a has a positive polarity for a rising edge transition while a negative polarity for a falling edge transition. Reference [29] further presents the width of the crosstalk-induced pulse on the far end as that shown in Figure 2.10. As it can be seen from the figure, the width of the pulse is shown to be equal to the edge transition time of the aggressor signal.

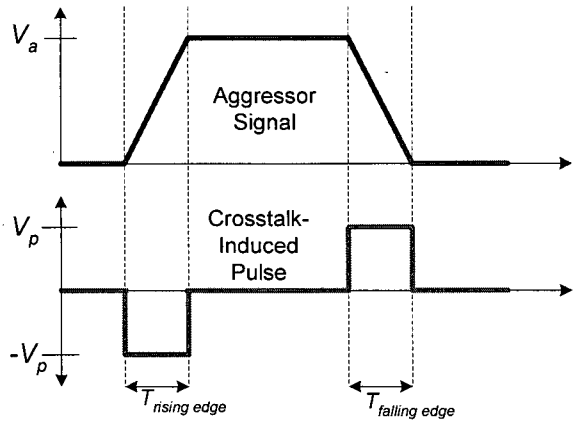


Figure 2.10: Crosstalk-Induced Pulse on the Far End

The above figure provides a good approximation for crosstalk-induced pulse predictions. In practice, the crosstalk-induced pulse shape is not exactly rectangular but trapezoidal with a rising and falling edge transitions. This rising and the falling transition times of the induced pulse corresponds to the time difference of the odd and even propagation modes on the victim trace. Please refer to reference [30] for details. Note that the total width of the crosstalk-induced pulse is equivalent to the edge

transition time of the aggressor. In this work, we will assume a rectangular pulse because this rising and falling edge transition of the trapezoidal crosstalk-induced pulse is calculated based on the method provided in [30] with only *3.2ps* when the edge transition time of the aggressor is *100ps*.

Note that in Equation 2.10, V_p can theoretically become infinitely large if the edge transition time of the aggressor signal is very small. In practice, this does not happen. Instead, the amplitude of V_p saturates at half the amplitude of the aggressor. Figure 2.11 shows the saturation phenomenon observed from *HSPICE fieldsolver* simulation. Details on the simulation methodology are explained in Chapter 4. In this figure, as the edge transition time is reduced below *0.03* ns, the crosstalk-induced pulse amplitude saturates to *~0.25V* for aggressor signal amplitude of *0.5V* with a *1*Gbps data rate. This same saturation phenomenon has been reported by [30, 31, 32] and explained in [30] in terms of the proportionality of the edge transition time to the time delay between odd and even signal propagation modes induced on the victim.

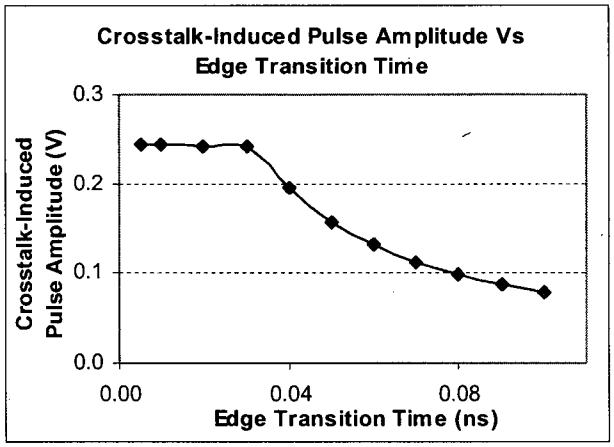


Figure 2.11: Crosstalk-Induced Pulse Amplitude at Far End

2.2 Jitter Analysis Techniques

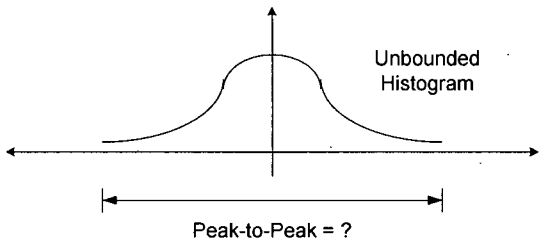
In this section, we will discuss jitter analysis issues that are relevant to this work such

as: extracting jitter from oscilloscope-captured data, and histogram analysis. For a broader coverage of additional jitter analysis techniques, the reader is encouraged to read references [20,33].

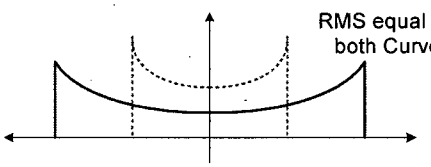
One difficulty with jitter analysis is identifying the different jitter components contributing to the total jitter. A common method for separating different jitter component is to perform mathematical analysis of the data captured with oscilloscope, either a real-time or an equivalent-time sampling oscilloscopes [33]. These analyses are capable of estimating random and deterministic jitter. Many mathematical analysis have been developed in the past few years, but few are reliable beyond *2.5~3.2* Gbps range [34]. Therefore, the development of these post-processing mathematical tools is still currently a topic of active research.

Specifying jitter and noise simply through *peak-to-peak* or *RMS* values is deemed insufficiently accurate [22]. A *peak-to-peak* value is sample size dependent and is inaccurate in the presence of random noise. This is because, by definition, random noise generally represented as a Gaussian distribution is unbounded. Figure 2.12 (a) shows the ineffectiveness of representing jitter with a peak-to-peak value when the distribution is unbounded. A peak-to-peak RJ measurement is ambiguous unless some boundary condition is established. Conversely, describing TJ simply via a *RMS* value is inaccurate in the presence of DJ. This is because a DJ PDF can take any form and thus cannot be uniquely describe with a RMS DJ value. A RMS value is only valid for describing a Gaussian distribution, i.e. one for random jitter (RJ). Therefore, a representation of jitter that includes the shape of the distribution is necessary. For that reason, histograms are used in the presence in the presence of DJ. Figure 2.12 (b) shows such an example when

there are two different periodic jitter (PJ) distributions but with equivalent RMS value.



(a) Drawback of Peak-to-Peak value



(b) Drawback of RMS value

Figure 2.12: Peak-to-Peak and RMS Values

A histogram is a diagram that plots the frequency of the occurrence of the measurements versus the measurement values. In jitter analysis, measurement values are usually in time units. Statistically, when the set of data approaches a large number, the histogram provides a good estimate of the shape of the PDF [5]. Since the oscilloscopes we used to observe the histogram do not show the number of data samples for the y-axis, it is difficult to choose the same number of data samples for simulation and predictions. Therefore, the histogram is normalized to show the probability of occurrences on the y-axis instead of the number of samples. The number of bins of a histogram determines how well the histogram will reflect the distribution. This value is adjusted in a case-by-case basis depending on number of samples. In this thesis, each bar, corresponding to one bin, is defined as a *delta line*. Histograms can be created from rising/falling edge time events, clock period variations, signal voltage variations, etc [16]. In this work, histograms are generated from the edge crossing point times of the victim signal.

Chapter 3 Prediction of BUJ

In this chapter, the superposition principle is applied to crosstalk to establish the BUJ jitter models. This model is extended to calculate the peak-to-peak BUJ, denoted as $BUJ_{p\text{-}p}$. In addition, algorithms to generate BUJ histograms are developed.

3.1 The Superposition Technique

Reference [1] suggests using superposition to sum up the victim signal with the induced crosstalk pulse to calculate the total distorted victim signal. Superposition can be applied to any linear system, where interconnects is a subset. Such a technique is illustrated in Figure 3.1. In this figure, an ideal victim signal is shown on the top left corner. This signal is added with the crosstalk-induced pulse induced by the aggressor signal shown on the top right corner of the figure. Both signals from the top left and top right corners are added together to form the distorted victim signal as shown in the bottom. For simplicity, there is no skew between the victim and the aggressor signals. In addition, the rising and falling edge transition times are equal.

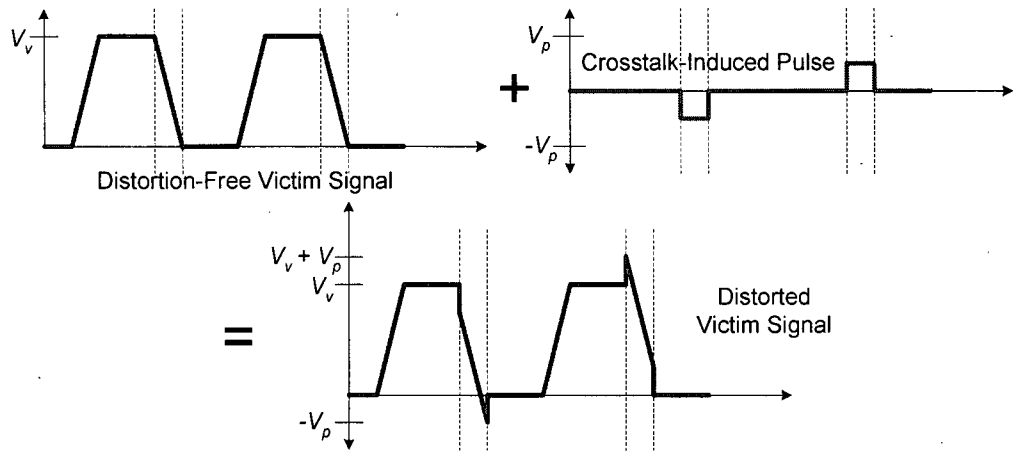
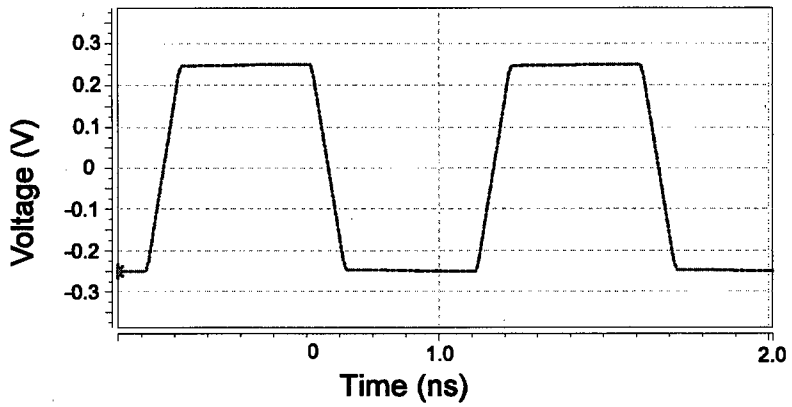
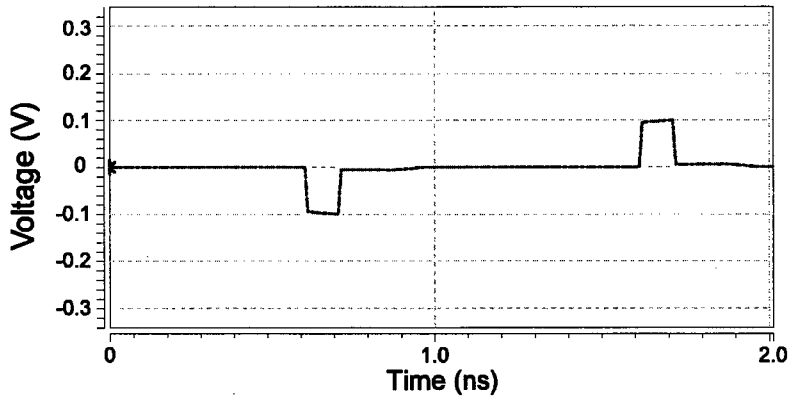


Figure 3.1: Distorted Victim Signal Obtained from Superposition

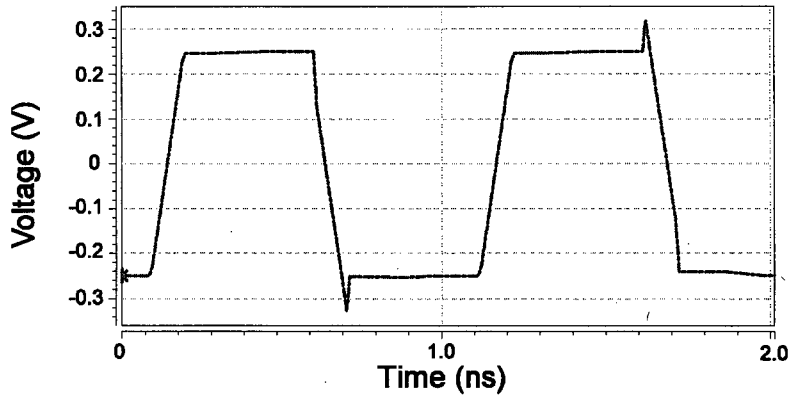
The similar illustration is also obtained from *HSPICE Fieldsolver* simulation with the victim signal operating at *1GHz*, and rising and falling edge transition time of 0.1ns.



(a) Distortion-Free Victim Signal



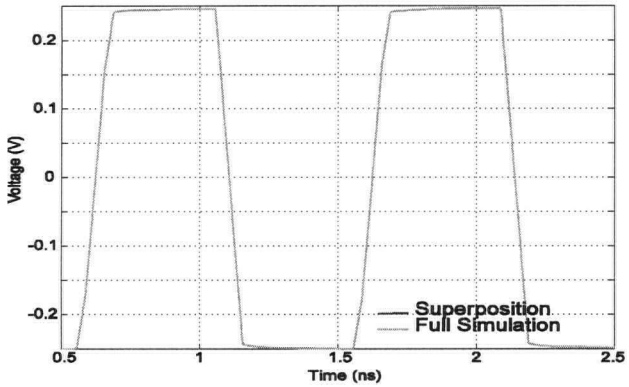
(b) Crosstalk-Induced Pulse



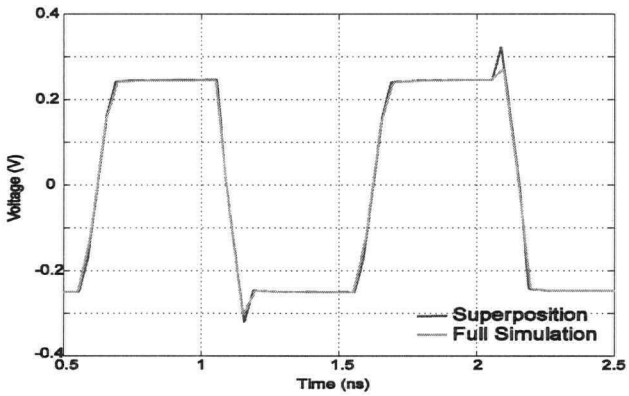
(c) Distorted Victim Signal, (a) + (b) = (c)

Figure 3.2: Simulation of the Distorted Victim Signal Obtained from Superposition

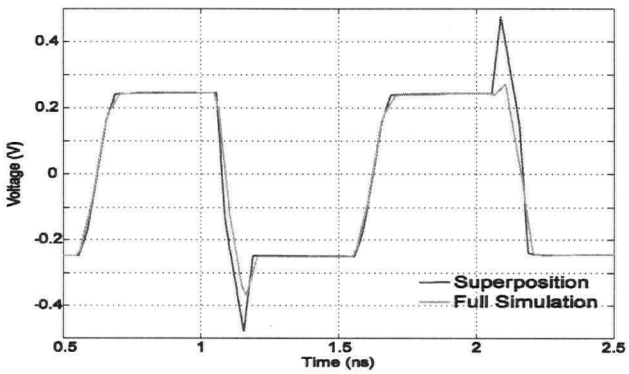
The resulting distorted victim signal obtained from superposition is accurate if the system is weakly coupled. In this subsection, Figure 3.3 is used to illustrate qualitatively the accuracy of applying superposition to determine the distorted victim signal. Figure 3.3 shows a comparison between distorted victim signals obtained from superposition and from *HSPICE* simulations. Note that all simulated signals correspond to the work described in Chapters 4 and 5. Also, note that since the simulated and calculated plots are generated using different CAD tools, they were super-imposed manually to match the rising edges and thus their alignment is subjective. Nonetheless, the figures are valid to illustrate qualitatively slight differences in timing and amplitude. Figure 3.3 (a) shows the distorted victim signals generated from one aggressor and one victim system. Note that both signals overlap and are indistinguishable. Hence, superposition is accurate in this situation. Figure 3.3 (b) shows the distorted victim signals for one victim trace sandwiched between two aggressor traces. Compared to Figure 3.3 (a), a slight timing difference can be observed, this is an indication of strong coupling between aggressors. When the number of parallel traces is expanded to five with two aggressor traces on each side of the victim trace, the time difference between the distorted victim signals becomes noticeable as shown in Figure 3.3 (c) due to strong coupling between the four aggressors. The reason the distorted victim signals in Figure 3.3 does not exhibit similar waveform as in Figure 3.1 due to *HSPICE fieldsolver* numerical analysis error.



(a) One aggressor and one victim trace



(b) Victim trace sandwiched between two aggressor traces



(c) Two aggressor traces on each side of the victim trace

Figure 3.3: Comparison of Signals Generated by Superposition and Simulation

3.2 Crosstalk and Timing Jitter

Figure 3.4 is identical to Figure 3.2 (b) except for the addition of the distortion-free victim signal used in the simulation. This is done to illustrate the timing difference

between the distorted and distortion free signal, and thus, the generation of timing jitter from crosstalk. Because the aggressor signal (not shown in the figure) is operating at half of the data rate as the victim signal, the aggressor signal's edge transitions can only affect either the rising or the falling edge transitions. In this figure, only the falling edge transition is affected.

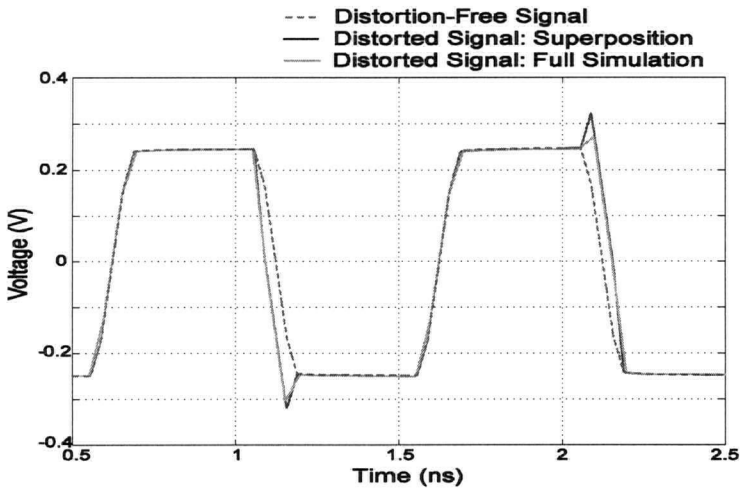
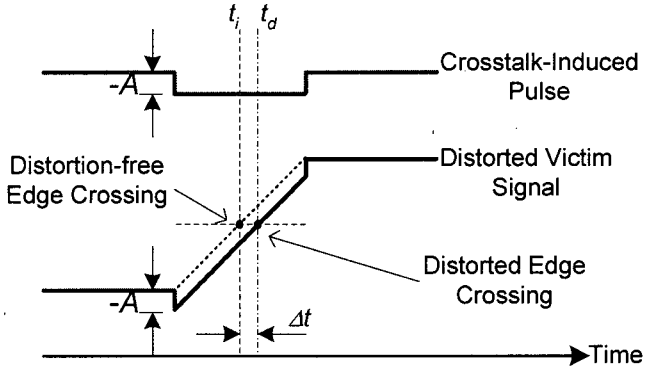


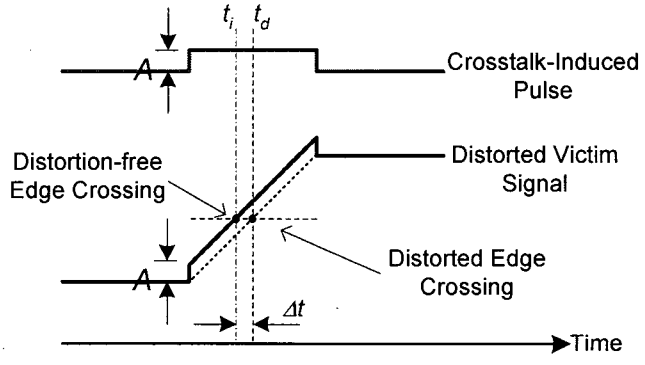
Figure 3.4: Comparison of the Distortion-free and the Distorted Victim Signals

Figure 3.5 further illustrates that the distorted victim edge transition can occur earlier or later than the distortion-free victim edge transition. If the crosstalk-induced pulse has negative voltage amplitude, the distorted victim rising edge will occur later than the distortion-free edge transition as illustrated in Figure 3.5 (a). On the other hand, the distorted victim rising edge transition will occur earlier than the distortion-free edge transition if the crosstalk-induced pulse has positive voltage amplitude as illustrated in Figure 3.5 (b). The time difference can be calculated from subtracting the edge crossings of the distortion-free and the distorted victim edge transitions. Two terms need to be introduced: *distortion-free edge crossing time* and *distorted edge crossing time*. *Distortion-free edge crossing time* refers to the distortion-free victim signal. *Distorted*

edge crossing time refers to the edge crossing of the distorted victim signal. They both indicate the moment in time when the edge crossings cross the logical threshold level. In this work, the logical threshold, the edge crossing point, is located at half the amplitude of the signal. The time difference, Δt, is simply the distorted edge crossing time, t_d, subtract the distortion-free edge crossing time, t_i.



(a) Distorted edge crossing occurs later than the distortion-free edge crossing

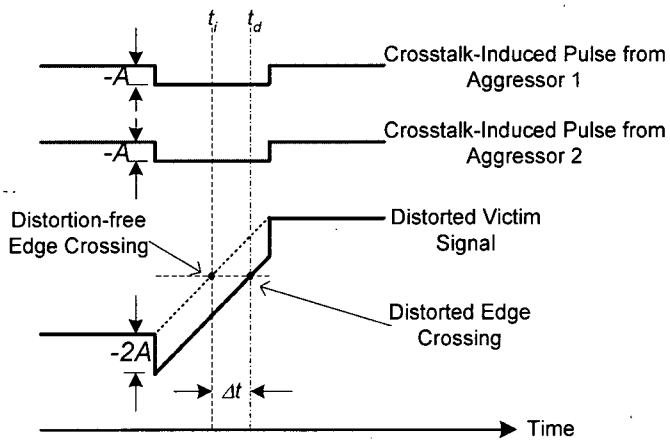


(b) Distorted edge crossing occurs earlier than the distortion-free edge crossing

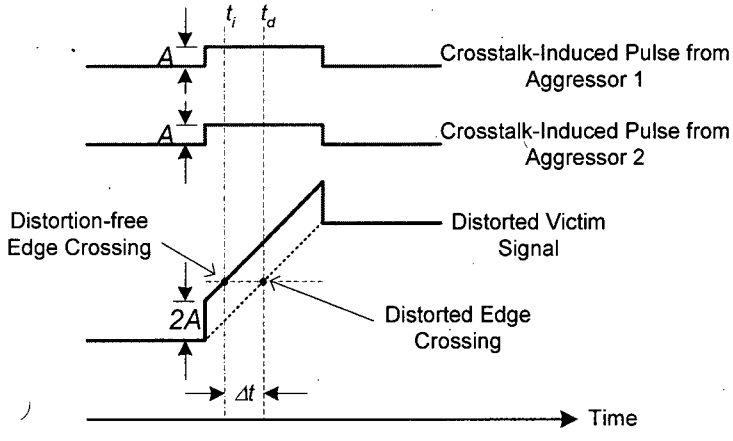
Figure 3.5: Earlier and Later Occurrences of the Distorted Edge Crossing Point

The previous discussion can be extended to one victim placed between two aggressors. This is shown in Figure 3.6. In Figure 3.6 (a), the distorted victim signal is calculated by adding two crosstalk-induced pulses each with amplitude $-A$. This creates a $-2A$ amplitude difference of the distorted victim signal, and consequently a longer Δt. Similarly, Figure 3.6 (b) shows the case when the two aggressors (victim trace

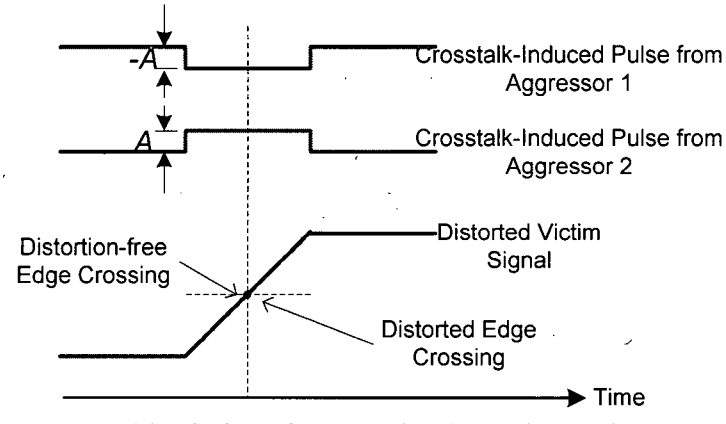
sandwiched in the middle of the two aggressors) produce an earlier edge crossing. Observe that, in comparison with Figure 3.5, Δt in Figure 3.6 (a) and (b) are twice as much. This graphical analysis is useful to understand how the amplitude of the total crosstalk-induced pulse sets the limit to the maximum timing difference in the victim edge crossings. Figure 3.6 (c) shows the additional case when the crosstalk-induced pulses have opposite polarities. The two crosstalk-induced pulses cancel each other and hence the victim signal is unchanged.



(a) Distorted edge crossing occurs later than the distortion-free edge crossing



(b) Distorted edge crossing occurs earlier than the distortion-free edge crossing



(c) Victim edge crossing is unchanged

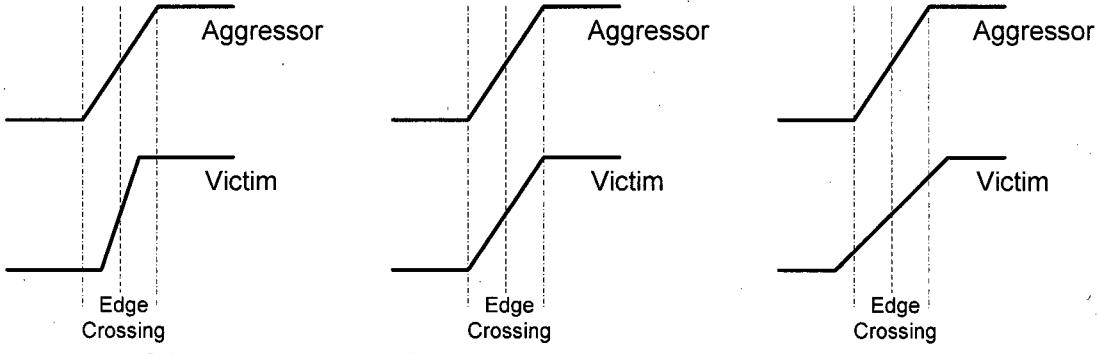
Figure 3.6: Illustration of the Distorted Edge Crossing Points Due to Crosstalk

3.3 BUJ Mathematical Model

From previous subsections, crosstalk will affect the edge transitions of the victim signal. It is possible to derive an equation to calculate such timing differences by knowing the signal shape of the crosstalk pulse based on superposition. In the rest of this section, we will use a graphical illustration to derive such equations.

3.3.1 Different Scenarios of Crosstalk Effects on the Victim

If both the aggressor and the victim signal have their edge crossings aligned, three scenarios exists as shown in Figure 3.7. In this figure, the edge transitions of both the aggressor and the victim are represented as rising edge transitions. They could instead be illustrated as falling edges.



(a) T_{edge} of the victim < T_{edge} of the aggressor

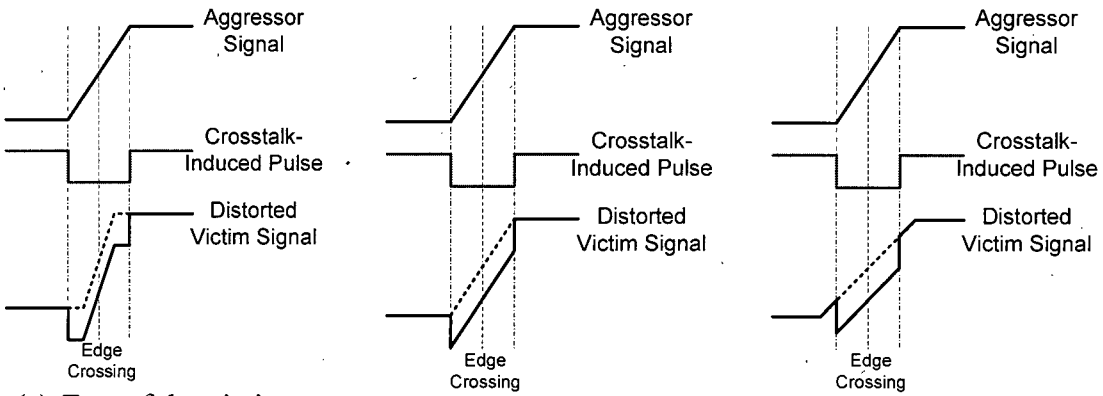
(b) T_{edge} of the victim = T_{edge} of the aggressor

(c) T_{edge} of the victim > T_{edge} of the aggressor

Figure 3.7: Different Aggressor and Victim Edge Transition Scenarios

Figure 3.8 below shows the resulting distorted victim signals of the three scenarios in Figure 3.7. In Figure 3.8 (a), the crosstalk-induced pulse is wider than the victim signal's edge transition width. Therefore, more than just the edge transition time of the victim signal is affected by the crosstalk. In Figure 3.8 (b), both the aggressor and the victim have the same edge transition time, therefore, the crosstalk-induced pulse only affects the

edge transition of the victim signal. In Figure 3.8 (c), only a portion of the victim signal's edge transition time is affected because the width of the crosstalk-induced pulse is shorter than the victim's signal edge transition.



(a) T_{edge} of the victim < T_{edge} of the aggressor

(b) T_{edge} of the victim = T_{edge} of the aggressor

(c) T_{edge} of the victim > T_{edge} of the aggressor

Figure 3.8: Graphical Representation of the Distorted Victim Signals with Different Edge Transition Scenarios

For the sake of discussion from now on, we will describe the middle section of the distorted victim signal as the *ramp*, illustrated in Figure 3.9, and the left and right instantaneous amplitude changes as the left and right *vertical ends*.

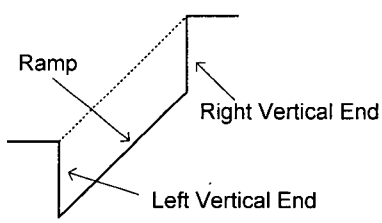
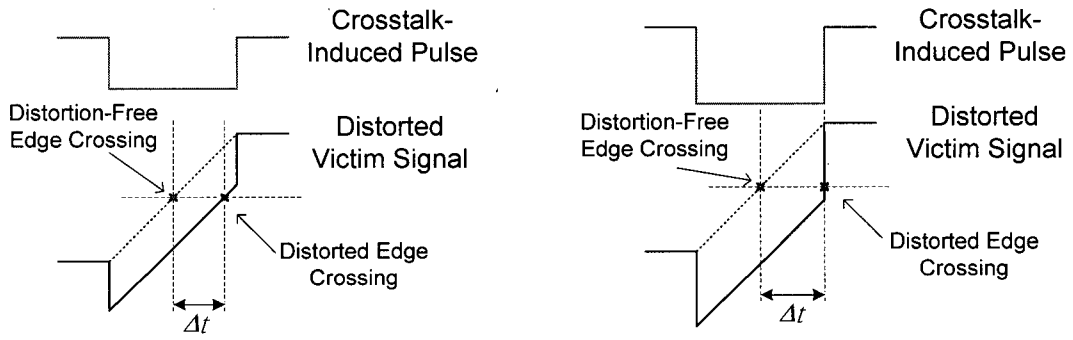


Figure 3.9: Ramp and Vertical End Illustration

Now consider two additional sub-scenarios of Figure 3.8 (b) that emphasize the amplitude of the crosstalk-induced pulse. The first sub-scenario is shown in Figure 3.10 (a) where the distorted edge crossing lies on the ramp of the victim signal. The second sub-scenario is shown in Figure 3.10 (b) where the distorted edge crossing lies on the

right vertical end of the victim signal.

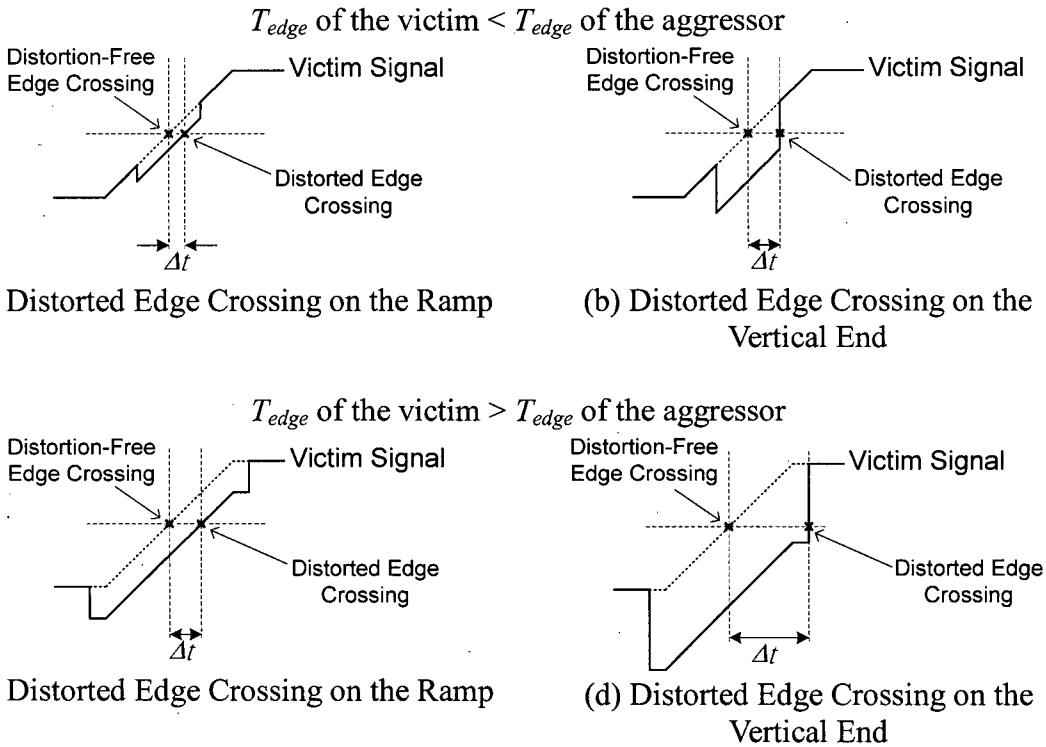


(a) Small crosstalk-induced pulse amplitude

(b) Large crosstalk-induced pulse amplitude

Figure 3.10: Effect of Crosstalk-Induced Pulse Amplitude on the Victim Edge

The two sub-scenarios of Figure 3.10 also apply to the cases of Figure 3.8 (a) and (c). This is illustrated in Figure 3.11. As can be seen from all these cases, Δt is bounded at the left or right vertical ends.



(a) Distorted Edge Crossing on the Ramp

(b) Distorted Edge Crossing on the Vertical End

(c) Distorted Edge Crossing on the Ramp

(d) Distorted Edge Crossing on the Vertical End

Figure 3.11: Edge Transition Scenarios due to Crosstalk-Induced Pulse

3.3.2 Deriving the Time Location of the Distorted Edge Crossing

As shown in Figure 3.10 and Figure 3.11, when the distorted edge crossings lay on the vertical ends, then the time difference between the distorted and the distortion-free edge crossing times is half of the crosstalk-induced pulse width time. This is because the crosstalk-induced pulse width time is equal to the edge transition time of the aggressor, T_a, and the distortion-free edge crossings are aligned with the center of the crosstalk-induced pulse width. This time difference can be described as:

$$\Delta t = \frac{T_a}{2} \tag{3.1}$$

where T_a is the edge transition time of the aggressor signal.

Also, from Figure 3.10 and Figure 3.11, when the distorted edge crossing lays on the ramp of the victim, then there are two points to observe as illustrated in Figure 3.12. First, the time difference between the distorted and the distortion-free edge crossing times depends on the amplitude of the crosstalk-induced pulse. As the amplitude of the crosstalk-induced pulse increases, the time difference increases. Secondly, the ramp of the distorted victim signal has the same slope as the edge transition of the distortion-free victim signal.

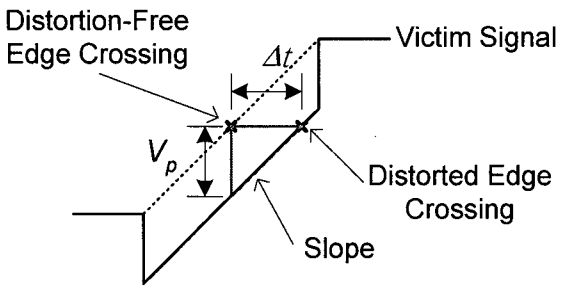


Figure 3.12: Relationship of Slope and Time Difference

The slope, m, as illustrated in Figure 3.12 is defined as:

$$m = -\frac{V_p}{\Delta t} \tag{3.2}$$

where V_p is the amplitude of the crosstalk as introduced in Chapter 2 and is similar to A in Figure 3.5 and Figure 3.6, and Δt is the time difference between the distorted and the distortion-free edge crossing times. To find out the time difference between the distorted edge crossing and the distortion-free edge crossing, Equation 3.2 is rewritten as:

$$\Delta t = -\frac{V_p}{m} \tag{3.3}$$

Since the slope of the victim signal can be computed from the amplitude, V_v, and the edge transition time, T_v, of the distortion-free victim signal, Equation 3.3 can also be written as:

$$\Delta t = -\frac{V_p}{\Delta V_v / T_v} = -\frac{V_p T_v}{\Delta V_v} \tag{3.4}$$

Equation 3.4 is the general equation to calculate the time difference, Δt, of the victim edge. This equation also reflects the fact that the distorted edge can occur earlier or later than the distortion-free edge crossing. For example, in Figure 3.13, the crosstalk-induced pulse has a negative amplitude polarity. In the falling edge transition of the distorted victim signal, the distorted edge crossing time occurs earlier than the distortion-free edge crossing time and results in a negative time difference.

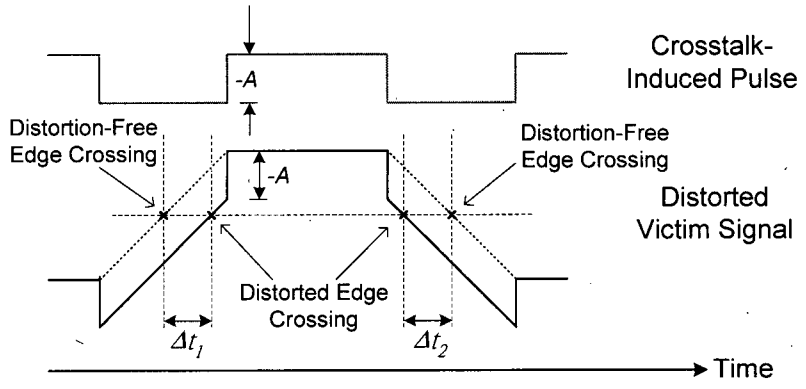


Figure 3.13: Faster/Slower Victim Edge Crossing due to Crosstalk-Induced Pulse

As the amplitude of the crosstalk-induced pulse increases, the distorted edge crossing will eventually reach the vertical end of the victim signal. Therefore, Equation 3.1 is used to calculate the time difference instead of Equation 3.4. However, this equation does not take into consideration the slope of the edge transition and the polarity of the crosstalk-induced pulse as Equation 3.4 did. Therefore, to be able to determine whether the distorted edge crossing will occur earlier (a negative time difference), or later (a positive time difference), Equation 3.1 is expanded as follows:

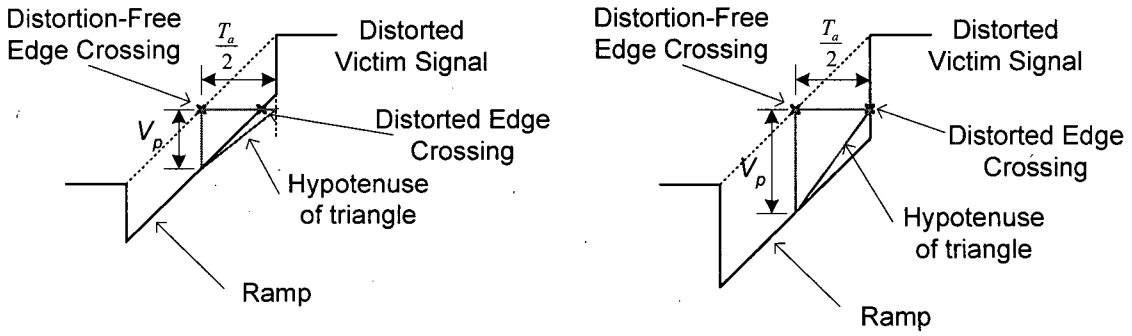
$$\Delta t = \frac{T_a}{2}, \qquad if \frac{V_p}{m_v} < 0 \tag{3.5a}$$

$$\Delta t = -\frac{T_a}{2}, \qquad if \frac{V_p}{m_v} < 0 \tag{3.5b}$$

where T_a is the edge transition time of the aggressor, V_p is the amplitude of the crosstalk-induced pulse, and m_v is the slope of the victim's edge transition. Equation 3.5a is for the condition when the edge crossing of the distorted victim signal lies on the right vertical end. Conversely, Equation 3.5b is for the condition when the edge crossing of the distorted victim signal lies on the left vertical end.

By analyzing the distorted signal graphically, we can decide whether to choose

Equation 3.4 or 3.5. However, for computation purposes, a split equation that applies to all of the conditions is necessary. Figure 3.14 (a) and (b) help devise such equation. Both Figure 3.14 (a) and (b) have a triangle drawn whose slope is computed as $\left|\frac{V_p}{T_a/2}\right| = \left|\frac{2V_p}{T_a}\right|$. Figure 3.14 (a) shows the scenario when the distorted edge crossing lies on the ramp of the victim edge transition. On the other hand, Figure 3.14 (b) shows the scenario when the distorted edge crossing lies on the vertical end of the distorted victim signal.



(a) Distorted edge crossing locate on the ramp of the distorted victim signal

(b) Distorted edge crossing locate on the vertical end of the distorted victim signal

Figure 3.14: Boundary Conditions of the BUJ Model

For the distorted edge crossing to lay on the ramp of the distorted victim signal, the slope of the hypotenuse must be smaller than the slope of the edge transition of the distorted victim signal as illustrated in Figure 3.14 (a). On the other hand, for the distorted edge crossing to lay on the vertical end, the slope of the hypotenuse must be greater than the slope of the edge transition of the distorted victim signal as illustrated in Figure 3.14 (b). Therefore, the split-equation interval conditions are expressed as:

$$\left|2V_p/T_a\right| < \left|m_v\right| \tag{3.6a}$$

$$\left|2V_p/T_a\right| > \left|m_v\right| \tag{3.6b}$$

where m_v is the rate of change of victim voltage over time.

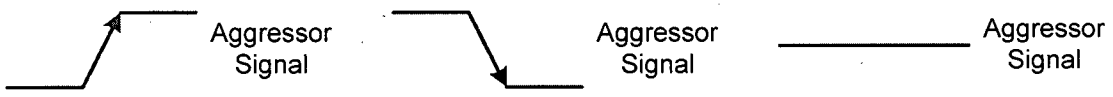
Equations 3.1 to 3.6 can be summarized as follows:

$$\Delta t = \begin{cases} -\dfrac{V_p T_v}{\Delta V_v}, & if \left|2V_p/T_a\right| < \left|m_v\right| \\ \dfrac{T_a}{2}, & if \left|2V_p/T_a\right| > \left|m_v\right| \ and \ \dfrac{V_p}{m_v} < 0 \\ -\dfrac{T_a}{2}, & if \left|2V_p/T_a\right| > \left|m_v\right| \ and \ \dfrac{V_p}{m_v} < 0 \end{cases} \qquad (3.7)$$

where V_p is the amplitude of crosstalk-induced pulse, m_{victim} is the rate of change of victim voltage over time, and T_a and T_v are the edge transition time of the aggressor and victim signal respectively. V_p can be computed by Equation 2.10. In the next section, Equation 3.7 is extended to calculate $BUJ_{p\text{-}p}$. In Sections 3.6 and 3.7, this equation is used as the basis for algorithms that generate the histogram of BUJ.

3.4 Peak-to-Peak BUJ ($BUJ_{p\text{-}p}$)

Normally, in digital signals there are only two types of edge transition, the rising and falling edge transitions. In this thesis, the inactivity between consecutive '0's or '1's is also considered as an edge transition. This is illustrated in Figure 3.15.



(a) A rising edge transition (a) A falling edge transition (c) No edge transition

Figure 3.15: Three Types of Edge Transitions

When the aggressor signal is a Pseudo-Random Bit Stream (PRBS) pattern, all these transition events will occur. Whenever a rising or a falling edge transition occurs, through crosstalk, these will either delay or advance the edge crossings of the victim

signal. Whenever two consecutive logic *'0'*s or *'1'*s occur, there is no crosstalk and hence the timing of the victim's edge transition is unaffected.

In a two parallel traces configuration, the time difference between the earlier and later occurrences of the victim's edge crossing is referred to as $BUJ_{p\text{-}p}$. $BUJ_{p\text{-}p}$ can be written as:

$$BUJ_{\text{p-p}} = \left|\Delta t_{aggressor\_rise}\right| + \left|\Delta t_{aggressor\_fall}\right| \qquad (3.8)$$

where Δt is calculated from Equation 3.7 using either the aggressor's falling and rising edge transition time. This peak-to-peak value is adequate to describe BUJ for two parallel traces. However, this equation does not take into consideration the probability of occurrence of the different types of edge transitions described before and represented in Figure 3.15. In the following section, algorithms to generate BUJ histogram will be developed.

3.5 BUJ Histogram Generation

3.5.1 Two Parallel Traces

The pseudo-code to generate the histogram for two parallel traces is presented below:

```
1     calculate the self and mutual capacitance and inductance;
2     select one edge type "z" from the victim pattern;
3     for each occurrence i of z in the victim pattern
4         if the aggressor has a rising edge
5             determine V_p;
6             time_diff(i) = Δt_aggressor_rise ;
7         else if the aggressor has a falling edge
8             determine V_p;
9             time_diff(i) = Δt_aggressor_fall ;
10        else if the aggressor has no edge transition
11            time_diff(i) = 0;
12        end if;
13    end for;
14    plot histogram from time_diff;
```

Table 2: Pseudo-Code to Plot the Histogram for Two Parallel Traces

In Table 2, first, the self and mutual capacitance, and inductance need to be calculated (*line 1*). This can be calculated by using the equations introduced in Chapter 2, or alternatively, by using a fieldsolver program. A fieldsolver is a simulation-based program that can compute the resistance, capacitance, and inductance matrix as well as other parameters by providing the geometrical configuration and the material properties of the PCB traces. Details about fieldsolvers will be discussed in Chapter 4.

Next, a particular edge transition *"z"* is selected among all the edges of the victim pattern (*line 2)* as shown in the example in Figure 3.16. In this example, the victim signal has a '1001110' data pattern that repeats *n* times (*line 3*). An edge "*z"* is selected from the data pattern.

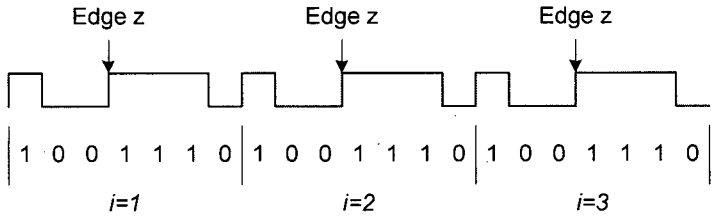


Figure 3.16: A Repeating Victim Data Pattern

The selection of a particular edge *"z"* is done to observe the effect of the different types of aggressor edge transitions on this specific edge. Figure 3.17 illustrates this concept by adding an aggressor data pattern. Note that the aggressor and the victim data patterns have different pattern lengths. In the first repeating pattern of the victim, that is *i=1*, edge *z* is affected by an aggressor's rising edge transition (*line 4*). When *i=2*, edge *z* is not affected, as there is no edge transition in the aggressor (*line 10*). When *i=3*, edge *z* is affected by a falling edge transition of the aggressor signal (*line 7*). In theory, the larger the amount of samples collected over time, the better the histogram represents the probability density function.

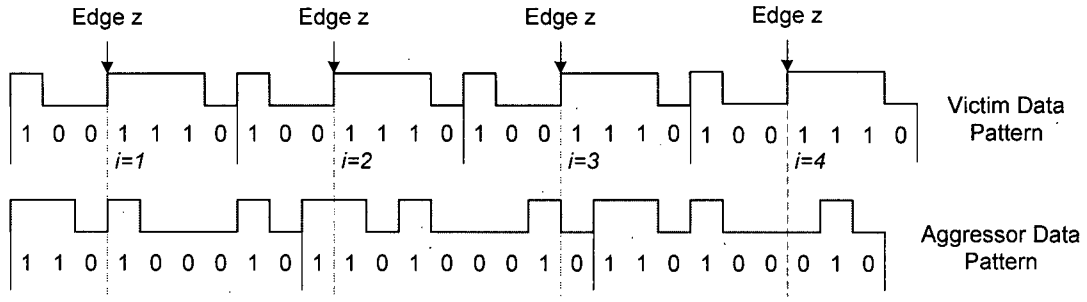


Figure 3.17: Edge z Affected by Different Aggressor Edge Transitions

The crosstalk-induced amplitude, V_p, is then calculated (*lines 5 and 8*) using Equation 2.10. This amplitude depends on the rising or falling edge transition time. In *line 6*, the time difference, *time_diff* is calculated with Equation 3.7 from the aggressor's rising edge transition time, whereas in *line 9*, the time difference is calculated from the aggressor's falling edge transition time. If the aggressor has no edge transition, the *time_diff* is simply set to zero, as there will be no crosstalk impact on edge *z (line 11)*.

Finally, after having computed and stored the *time_diff* for *n* repetitive patterns, the stored information is used to plot the histogram (*line 14*).

3.5.2 Multiple Parallel Traces

The algorithm discussed in Table 2 focused on one aggressor and one victim. In that algorithm, the aggressor could affect the victim only through *3* types of edge transition events. If there were two aggressors, there would be $3 \times 3 = 9$ different possible transition scenarios. If the number of aggressor traces were increased to four, there would be $3^4 = 81$ possible transition scenarios.

For multiple aggressors, superposition can again be used to compute the amplitude of the total crosstalk-induced pulse as shown in Equation 3.11:

$$V_p = V_{p1} + V_{p2} + \cdots + V_{pn} \qquad (3.11)$$

where V_{p1} *to* V_{pn} are the crosstalk amplitudes induced on the victim individually by aggressor *1* to *n*, and V_p is the total crosstalk amplitude due to all the aggressor traces. Note that when there are more than two aggressors, they are further away from the victim signal with equal distance between adjacent aggressor traces. The further the aggressor away from the victim trace, the smaller the magnitude of the crosstalk-induced pulse, V_p.

Feeding V_p into Equation 3.7 with the result obtained from Equation 3.11 will provide the time difference that results from the effect of all aggressor traces. This calculation assumes that all aggressors have the same rising and falling edge transition times.

To fit Equation 3.11, the algorithm in Table 2 can be modified to accommodate multiple aggressors. This new algorithm is as follows:

```
1    calculate the self and mutual capacitance and inductance;
2    select one edge type "z" from the victim pattern;
3    for each occurrence i of z in the victim pattern
4        for j =1..m aggressors
5            if aggressor j has a rising edge
6                determine V_pj;
7            else if aggressor j has a falling edge
8                determine V_pj;
9            else if aggressor j has no edge transition
10               V_pj=0;
11           end if;
12       end for;
13       V_p = V_p1 + V_p2 + ... + V_pm ;
14       calculate Δt;
15       time_diff(i) = Δt;
16   end for;
17   plot histogram from time_diff;
```

Table 3: Pseudo-Code to Plot the Histogram for Multiple Parallel Traces

First, the capacitance and inductance values need to be calculated (*line 1)*. Next, similarly to the algorithm in Table 2, a specific edge "*z*" of the victim data pattern is selected. Then, the amplitude of the crosstalk-induced pulse, V_p, of each aggressor is computed (*line 6, 8, 10*). When all amplitudes V_p of all aggressors have been computed, they are added together based on superposition principle to obtain a total V_p (*line 13*). Finally, Δt is calculated based on the total V_p and stored in an array called *time_diff*. When *time_diff* has been computed and stored for n repetitive patterns, the stored information can be used to generate the histogram.

Chapter 4 Simulation and Measurement Setup

An objective of this thesis is to validate the models and algorithms developed in Chapter 3. Therefore, it is necessary to perform simulations and measurements to compare their results with those obtained from the models. In this chapter, we describe first a simulation setup able of accounting for crosstalk between user-defined signals transmitted through a user-defined PCB environment. Next, we describe experimental setups to observe crosstalk on custom designed PCBs. Finally, we describe the post-processing algorithm developed to obtain BUJ from simulated and measured data.

4.1 Simulation Program

To setup a simulation platform, it is important to select a software tool able to compute the inductance and capacitance of an arbitrary microstrip configuration in an accurate and rapid manner. The most accurate numerical tools used to perform these calculations are field simulators, generally referred to as *fieldsolvers* [1]. A second software tool is necessary to read the matrix output of the *fieldsolver* and perform time domain transient analysis with user defined victim and aggressor signals. The simulation results obtained from transient analysis will be compared with the results obtained from the algorithms and models presented in Chapter 3. In this section, the concept of inductance and capacitance matrix is first introduced. These matrices are necessary for any simulator to perform transient analysis. Finally, the selection of *fieldsolvers* will be briefly discussed.

4.1.1 Inductance and Capacitance Matrix

In Figure 4.1, the cross section of two parallel microstrips is illustrated. In this

configuration, there are both self and mutual capacitances, C and C_m, respectively. In addition, self and mutual inductances, L and L_m, also exist. These four parameters can be placed into the capacitance and inductance matrices:

$$[C]=\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{4.1a}$$

$$[L]=\begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \tag{4.1b}$$

In Equation 4.1a, C_{11} and C_{22} represent the self-capacitances of trace *1* and *2* respectively. Similarly, L_{11} and L_{22} represent the self-inductances of trace *1* and *2* respectively. C_{12} and C_{21} are the mutual capacitances and L_{12} and L_{21} are the mutual inductances between the two traces.

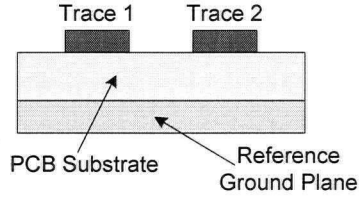


Figure 4.1: Cross Section of Two Parallel Microstrips

When there are n parallel traces as illustrated in Figure 4.2, the size of the matrix becomes $n \times n$. Then, the capacitance and inductance matrices are:

$$[C]=\begin{bmatrix} C_{11} & C_{11} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n1} & \cdots & C_{nn} \end{bmatrix} \tag{4.2a}$$

$$[L]=\begin{bmatrix} L_{11} & L_{11} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n1} & \cdots & L_{nn} \end{bmatrix} \tag{4.2b}$$

In Equation 4.2, self-capacitances and inductances for traces *1* to *n* are represented as C_{11} to C_{nn} and L_{11} to L_{nn} respectively. The mutual capacitance between traces *1* and *2* is represented as C_{12} or C_{21}. Similarly, the mutual inductance is represented as L_{12} or L_{21}. These capacitance and inductance matrices can be generated by a fieldsolver given the dimensions and electrical properties of a PCB structure. These matrices are later on used not only by the transient analysis simulator, but also by the BUJ mathematical models and algorithms developed in Chapter 3, e.g. *line 1* of Table 2 and Table 3.

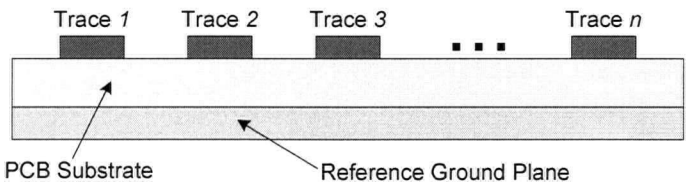


Figure 4.2: Cross Section of *n* Parallel Microstrips

4.1.2 Select a Fieldsolvers

A fieldsolver is a simulation program that is used to model the electro-magnetic field interactions between transmission paths and can calculate parameters such as trace impedances, propagation velocity, self, mutual capacitances and inductances [3]. Its calculation is based on Maxwell's equations. There are two categories of fieldsolvers: Two-Dimensional (*2-D*) and Three-Dimensional (*3-D*).

A *2-D* fieldsolver performs the necessary calculations to extract the parameters by analyzing the cross section of the PCB and traces. The advantages of *2-D* fieldsolvers are ease of use, fast computation times, and reasonable accuracy [1, 3]. On the negative side, 2-D fieldsolvers can only simulate relatively simple geometries and cannot calculate frequency dependent effects [3].

A *3-D* fieldsolver has the advantage of simulating *3-D* geometries. In addition, it can

predict frequency dependent effects such as skin effects[1]. Furthermore, it is much more accurate but also very difficult to use and takes significantly more computing times [3].

There are many commercially and academically available fieldsolvers. To name a few: *Maxwell 3D*; *HSPICE Fieldsolver*, and *Fast Model*. In this work, a *2-D* fieldsolver was used, namely *HSPICE Fieldsolver*. This allows us to integrate directly the inductance and capacitance matrices calculated by the fieldsolver into *HSPICE* transient analysis.

4.2 Experimental Setup

As mentioned earlier, an experimental setup is necessary to confront practical results with the models and the algorithms developed in Chapter 3. Such a setup must be able to generate crosstalk on multiple PCB traces, at speeds of *0.5~3*Gbps, and allow capturing of the distorted victim signal for the extraction of BUJ. In this section, we will describe first, custom PCB designs for high-speed interconnects used as the crosstalk test-bed, and second the experimental setups used in this work.

4.2.1 PCB Design

Figure 4.3 shows the physical *3-D* view of the PCB used for the experimental setup. The thickness of the FR4 substrate, the copper trace width and the trace thickness are selected based on PCB manufacturer's capabilities as well as to satisfy the characteristic impedance. The copper trace thickness, *1.68mils*, and substrate thickness, *10mils*, are limited by the selected manufacturer's capabilities. Note that *1mil* is equal to *0.0254mm*. For a dielectric constant of *4.0* and a characteristic impedance of *50Ω* the width of the

[1] The tendency of alternating current, as its frequency increases, to travel only on the surface of a conductor.

trace is calculated to be approximately *17mils*.

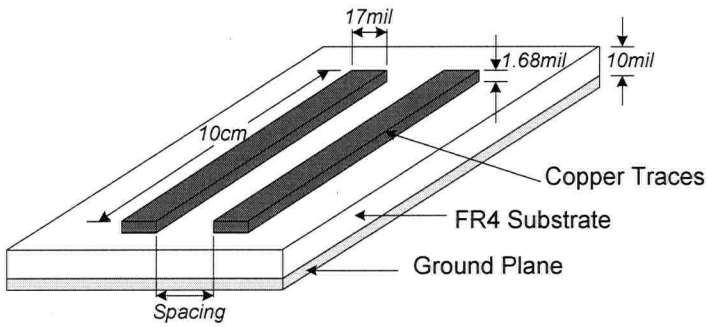


Figure 4.3: PCB Dimensions

Figure 4.4 shows the manufactured version of the PCB designed. On this PCB, there are three different sets of parallel traces. The top and bottom set of traces on this figure have only two parallel traces each. The length of the parallel traces is *10cm*. The spacing between the two traces on the top of the Figure 4.4 is labeled as *1x spacing*, to indicate that the spacing between the two traces is equal to the width of the trace. Similarly, the bottom set has a *2x spacing*, indicating the spacing between the traces is twice the width of the trace. Note that as the number of traces increases, the spacing stays the same between adjacent traces. Both ends of the parallel traces are accessible through SMA connectors. The middle set of four parallel traces in the figure has a *1x spacing* between traces. The length of each trace is *16cm* long. Other similar PCB configurations that include *2x* spacing and *3x* spacing of four parallel traces as well as other configurations were also designed but are not shown in the figure.

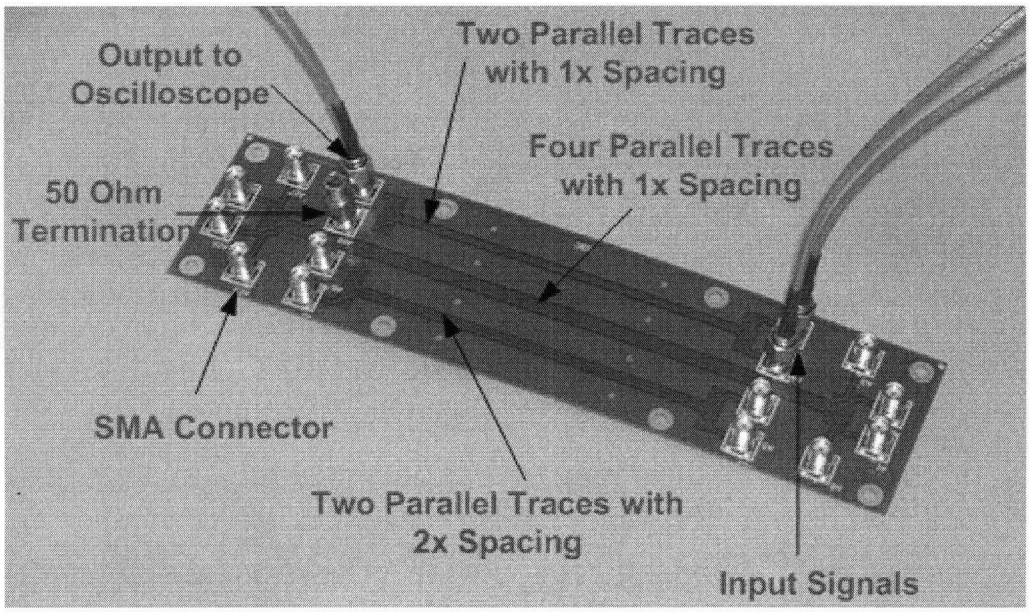


Figure 4.4: Photo of a Manufactured PCB

On the right hand side of Figure 4.4, the input signals from the signal sources are connected to both traces through SMA connectors. In this setup, one trace acts as the aggressor and the other acts as the victim. On the left hand side of the figure, the victim trace is connected to an oscilloscope to observe the victim signal behavior at the far end while the aggressor trace is simply terminated by a *50Ω* termination to avoid reflections.

4.2.2 Measurement System Overview

For our experiments, the ideal signal source must have the lowest possible total jitter; this is to reduce the introduction of additional jitter on top of the BUJ produced in the PCB. In addition, the ideal signal source must support multiple channels. We devised two experimental setups based on the availability of two different signal sources. The first setup employs an Agilent Error Performance Analyzer *86130A* as signal generator that provides the best jitter performance, *8ps* peak-to-peak, but only two signal channels. The second setup uses Lattice's ORT82G5 high-speed evaluation platform, which

provide multi-channel signals but with a larger total jitter per channel of 28ps peak-to-peak [35].

The first setup is shown in Figure 4.5. This setup includes a signal generator, a PCB under test, an oscilloscope and the Agilent *86130A*. The oscilloscope shown in the figure is a Tektronix *TDS8000* Equivalent-Time Digital Sampling Oscilloscope (ET-DSO). An ET-DSO can measure frequencies higher than its sampling rate but does not allow individual edge analysis. Another type of oscilloscope used in this work that solves the above-mentioned drawback is called a real-time sampling oscilloscope [20]. In this work, an Agilent *Infiniium 54856* real-time sampling oscilloscope is primarily used to perform individual edge and histogram analysis.

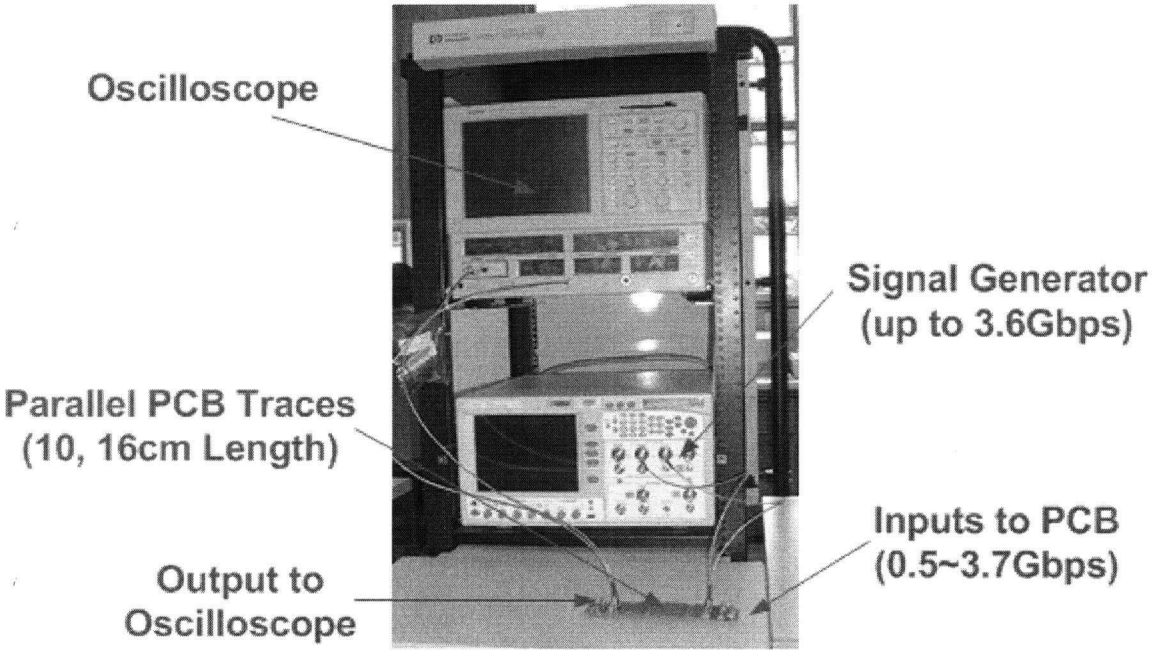


Figure 4.5: Measurement Test Setup with Two-Channel Signal Source

Figure 4.6 shows the setup to perform multi-channel crosstalk experiments. Lattice Semiconductor's high-speed multi-channel evaluation platform, *ORT82G5 Evaluation Board*, was used as the signal source. Because the evaluation platform requires a

differential reference clock signal, the *86130A* is used instead as the differential reference clock to the board. In the figure, four signal channels are connected from the evaluation platform to the PCB. The output of the PCB is connected to the oscilloscope.

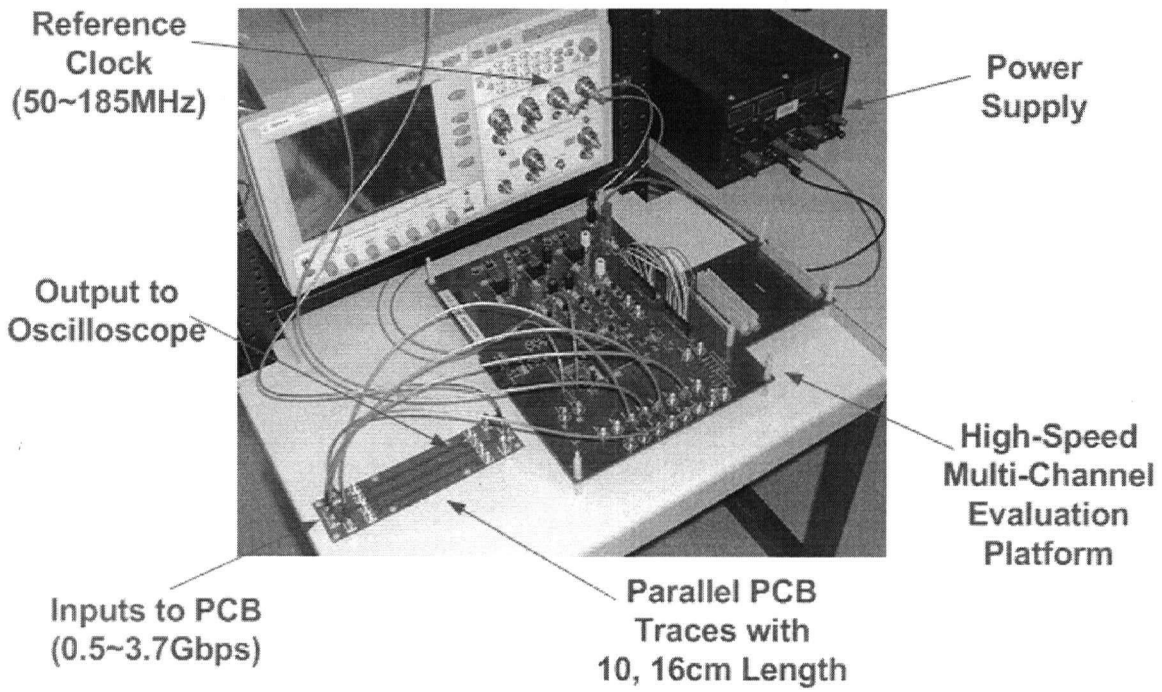


Figure 4.6: Measurement Test Setup with Multi-Channel Signal Source

4.3 Data Post-Processing

Both timing and amplitude data are recorded into files from either simulation or measurements. The sampling rate of the test instrument determines how frequent these values are sampled and stored. In our case, the real-time oscilloscope used has a maximum sampling rate of *20GHz*, i.e. *50ps* interval. On the other hand, simulations do not have such time interval constraints but reducing the time interval increases the simulation time.

The recorded data needs to be post-processed to remove unwanted jitter sources, such as RJ and PJ, to remove edge transition glitches, and to plot the BUJ histogram. The

pseudo-code of the algorithm that was used to post-process the recorded data is presented in the following table:

1	*Parse the victim signal data from simulation or measurement;*
2	*find all the edge crossing times*
3	*remove glitches in crossings;*
4	*select one edge type "z" from the victim pattern;*
5	$z_{crossing}[\ ]$ *= locate all subsequent occurrences of edge z crossings;*
6	*for i=1..n*
7	$\Delta t\,|_{z_i} = T_{measured/simulated}(z_i) - T_{distortion\text{-}free}(z_i)$*;*
8	*find the jth aggressor edge (out of m edges from the aggressor pattern) that corresponds to this victim edge z;*
9	*add* $\Delta t\,|_{z_i}$ *to array of time_diff(j);*
10	*end for;*
11	*for j=1..m edges of the aggressor pattern*
12	*average = the average of each row of the time_diff(j);*
13	*replace all the* $\Delta t\,|_{z_i}$ *values in the jth row, time_diff(j), with average calculated;*
14	*end for;*
15	*generate histogram from time_diff;*

Table 4: Pseudo-Code to Perform Data Post-Processing

In *line 1*, all the timing and amplitude data is first parsed. Next, all the edge crossing times are determined and saved. The edge crossing level is simply half the amplitude of

the victim signal. Because the recorded timing and amplitude values do not necessarily match the edge crossing point, interpolation is necessary to estimate the location of the edge crossings as illustrated in the example in Figure 4.7. In this example, the edge crossing is located between the two sampling points. Linear interpolation is applied to find the exact time location.

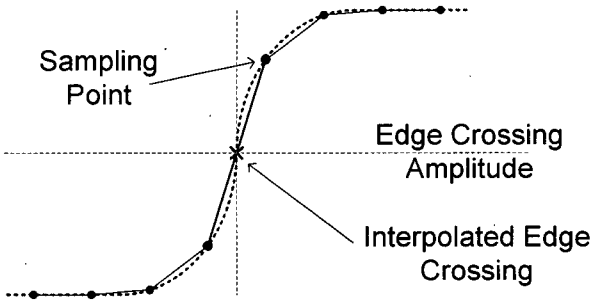


Figure 4.7: Linear Interpolation

Glitches exist in the signal waveform due to amplitude noises. Figure 4.8 shows three edge crossings for a single victim edge transition. After finding all the edge crossings (*line 2*), glitches that occur in the edge crossing need to be considered by taking the mean of these three edge crossing times (*line 3*).

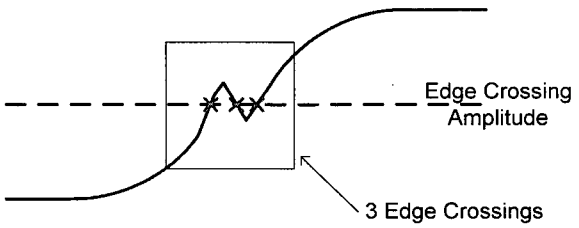


Figure 4.8: Glitch in Edge Transition

Similar to the pseudo-codes of the algorithms in Chapter 3, an edge transition *"z"* is selected among all the edges of the victim data pattern (*line 4*). If the victim signal was simply a clock, then this edge *"z"* would represent either all the rising or all the falling

edges of the clock signal. This is the case in this work where we focused on the crosstalk effect only on the falling edges of a high-speed clock. However, the victim could as well be any periodic arbitrary data pattern, and then the edge "*z*" represents the periodic occurrences of one unique transition event in the data pattern. When the victim is an arbitrary data pattern, the algorithm presented in this section must be used iteratively to focus on one edge "*z*" at the time. Since the data patterns in the aggressor(s) and the victim are not synchronized, as the pattern in the victim repeats, the selected edge "z" will be affected by different types of events, rising edge, falling edge and no transition, in the aggressor trace. The rest of the algorithm that will be explained gathers all these events to calculate their effect on the timing of edge *"z"*.

In *line 5*, all the occurrences of the *"z"* edge crossings, $z_{crossing}$, are now located and saved. In *line 7*, the time differences, $\Delta t|_{z_i}$, between the measured/simulated and the distortion-free edge crossing times are calculated. These values are then stored in the *j*th row of a multi-dimensional array called *time_diff* (*lines 8-9*). Figure 4.9 is used as an example to explain *lines 6~10*. In this figure, the victim data pattern is *'110'* whereas the aggressor data pattern is *'1000.'* Note that the victim data pattern should be *'10'* or *'01'* to avoid DDJ. However, to illustrate this algorithm, the victim data pattern is *'110.'* In the example, edge "*z*" of the victim data pattern is selected as the falling edge transition after the second consecutive *'1.'* First, the time difference between the actual edge crossing and the expected distortion-free edge crossing of the first edge *z* is calculated (*line7*). Next, the aggressor edge corresponding to this edge *z* of the victim data pattern is recorded (*line 8*). In this figure, it is the third edge of the aggressor data pattern that corresponds to the first edge *z* of the victim data pattern. The time difference, $\Delta t|_{z_i}$, is

then stored at the third row of the *time_diff* array (*line 9*). This completes one loop. Next, the time difference, $\Delta t|_{z_i}$, of the second repeating edge *"z"* is calculated and stored in the second row of the *time_diff* array as illustrated in Figure 4.9. When the time difference, $\Delta t|_{z_i}$, of the fifth edge "z" is calculated, this information is stored in the third row of the *time_diff* array. Since the first column is already occupied, this calculated time difference will be stored in the second column of the third row.

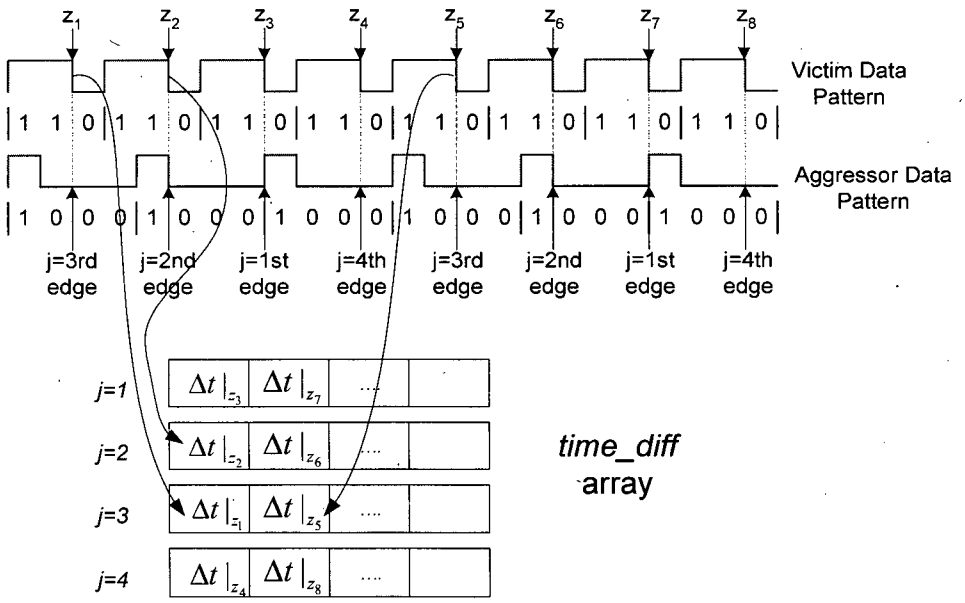
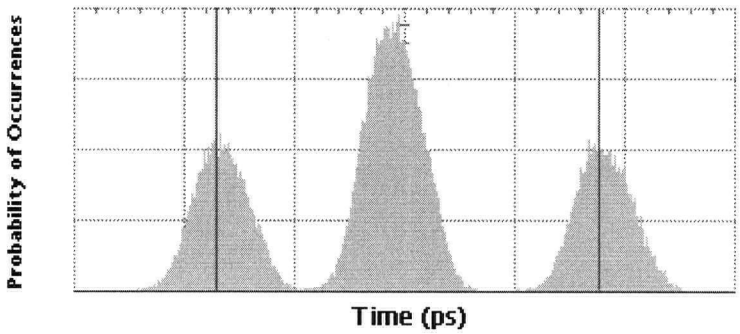


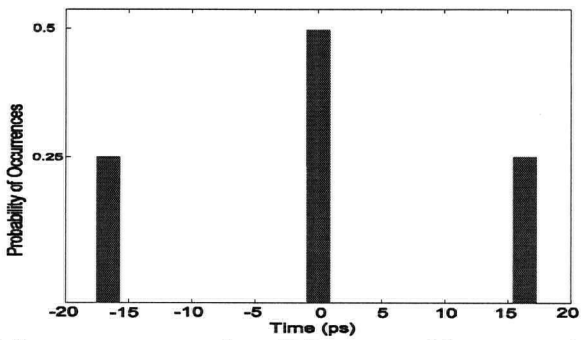
Figure 4.9: Example to Illustrate *lines 6~10* of Algorithm in Table 4

If the histogram is plotted immediately before any post-processing filtering, other jitter components will still exist. As an example, Figure 4.10 (a) shows the histogram measured with a real-time oscilloscope that also includes RJ, which is Gaussian distributed. Due to RJ's symmetrical distribution, it can be removed by averaging. Another high-frequency jitter component PJ, which is not shown in the figure, is also symmetrical and thus can be removed by averaging. *Lines 11~14* in the post-processing algorithm remove other jitter components that are symmetrical in their distribution. In

line 12, the average of each row of the array *time_diff* is computed; each column in the same row is replaced with the resulting average value. After going through all the rows of the array *time_diff*, the histogram is plotted. The histogram of Figure 4.10 (a) is filtered to simply three delta lines as shown in Figure 4.10 (b). This is under the assumption that non-symmetrical jitter components such as DDJ are small in the aggressor, which is true for experiments in this thesis. In Chapter 5, we analyze and discuss the results from both simulations and measurements for several crosstalk scenarios.



(a) Before post-processing (RJ included)



(b) After post-processing (RJ removed by averaging)

Figure 4.10: Histograms Plotted Before and After Post-Processing

In Figure 4.10, the histogram is plotted from data recorded for one aggressor and one victim traces. In part (b), the left and right delta lines represent the effect of the aggressor's rising and falling edge on the victim's falling edge. The left delta line

corresponds to the probability of occurrences of distorted victim edge crossing times that occurs earlier than the distortion-free victim edge crossing. This probability event depends on the aggressor data pattern. Conversely, the right delta line corresponds to the probability of occurrences of distorted victim edge crossing times that occurs later than the distortion-free victim edge crossing. The middle delta line represents the probability of occurrences of the victim edge crossing due to no edge transitions from the aggressor.

Chapter 5 Results and Discussion

In this chapter, we first report results of inductive and capacitive coupling on a PCB. This trend is reported for different spacing between parallel traces, increasing number of aggressors, and different aggressor data patterns. Next, the model predictions developed in Chapter 3 are compared against measurements and simulations. The algorithms to plot BUJ histograms will also be presented and compared with results obtained from simulations. Finally, the impact of time skew between signals will be presented.

5.1 Effect of Spacing between Parallel Traces

In this section, we show the impact of capacitive and inductive coupling on BUJ. In addition, results of increasing the number of aggressor traces are also studied. Furthermore, the results of different data patterns on $BUJ_{p\text{-}p}$ values are reported to illustrate the existence of DDJ.

5.1.1 The Trend of Capacitances and Inductances with Trace Spacing

The mutual capacitance and inductance for different trace spacings are presented in Figure 5.1. These values are obtained from simulations performed with *HSPICE*'s 2-D fieldsolver. The left side axis of Figure 5.1 represents the mutual capacitance in *pF*. Similarly, mutual inductance axis is represented on the right side in *nH*. As the spacing between two traces increases, both mutual inductances and capacitances decrease and eventually converge to zero. Self-inductances and capacitances are identical for different spacing between traces, and they are approximately *303nH* and *103pF* respectively.

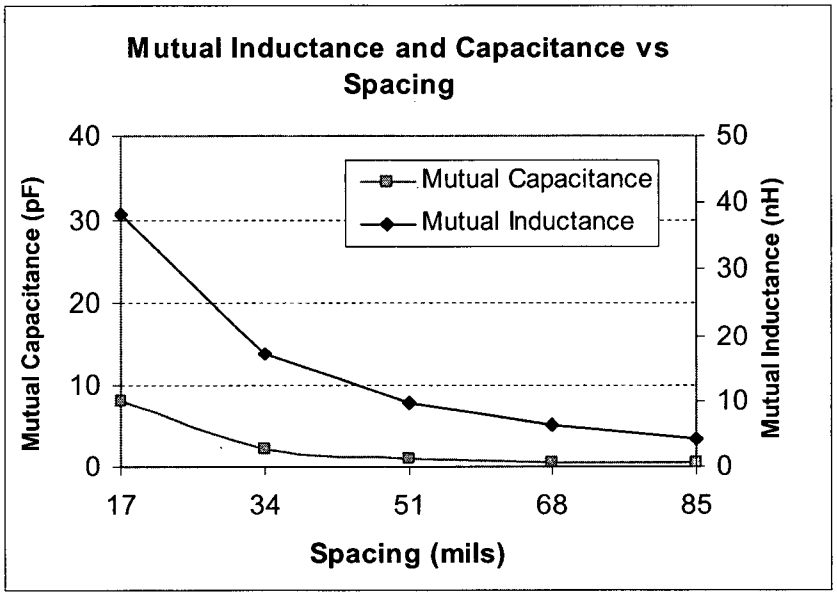


Figure 5.1: Simulation of Mutual Inductance and Mutual Capacitance Vs Trace Spacing

The polarity of the crosstalk-induced pulse equation in Equation 2.10 depends on the aggressor's amplitude change during transition, and on the factor $\alpha = \left(\frac{L_m}{L} - \frac{C_m}{C}\right)$. Figure 5.2 illustrates a comparison between the inductances ratio L_m/L and the capacitances ratio C_m/C at different parallel trace spacings. From this, we can predict the effect of α on the polarity of the crosstalk-induced pulse. These values are again obtained using *HSPICE's fieldsolver*. The results show that for our PCB setup, the term α is always greater than zero. As can be seen from the magnitudes, mutual inductances dominate mutual capacitances. Consequently, only the changes of the aggressor amplitude determine the polarity of the crosstalk-induced pulse. For example, Equation 2.10 will produce negative crosstalk-induced pulse amplitudes for rising edge aggressor transitions. Conversely, the crosstalk-induced pulse amplitude will be positive for an aggressor falling edge transition. Note that the ratios in Figure 5.2 are only applicable to

the PCB used in this work. It is possible to design a PCB such that α is minimized; for example, by using a stripline configuration.

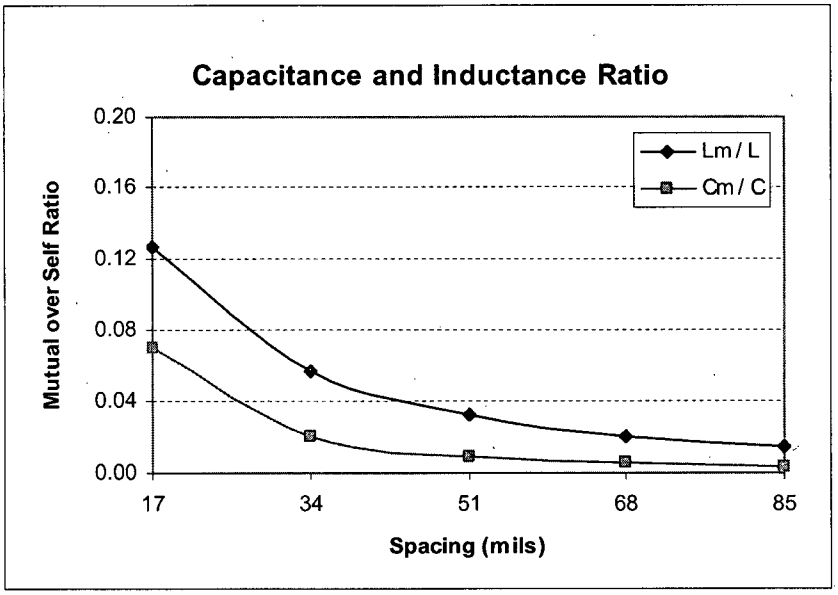


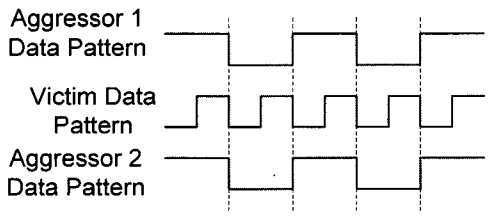
Figure 5.2: Simulation of Mutual over Self-Capacitance and Inductance ratios Vs Trace Spacing

5.1.2 Peak-to-Peak BUJ with Multiple Aggressors

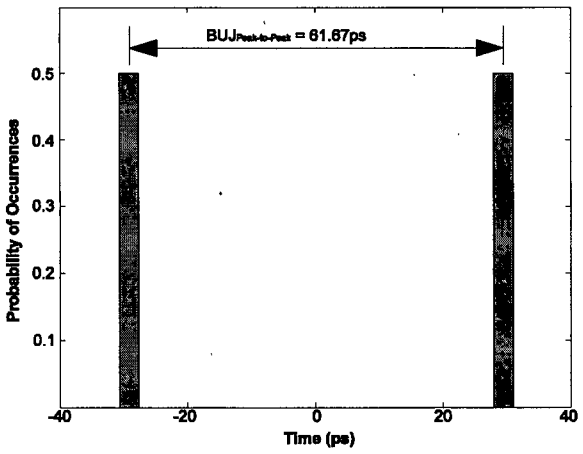
In this subsection, we explain the concept of $BUJ_{p\text{-}p}$ with the aid of a histogram. Then, we study the effect of increasing the number of aggressors on the $BUJ_{p\text{-}p}$. Due to measurement limitations of the available number of data channels, only simulation results are presented.

Figure 5.3 (a) illustrates clock-like data patterns of two aggressors and a victim signal running simultaneously. The aggressor signals are operating at half the data rate of the *1*Gbps victim signal. Notice that these two aggressor data patterns are identical simply to show a worst-case coupling scenario. Perturbation of the victim's timing can be determined from the time location of all rising and/or falling edges of the victim signal. The histogram shown in Figure 5.3 (b) is constructed based on the time difference of the

victim's falling edges. Since the edge transitions of the victim signal can only be affected by either the simultaneous falling or rising edge transitions of the aggressor signals, there are only two delta lines in the histogram. The right and left delta line in (b) are due to aggressor rising and falling edge transitions respectively. According to Equation 3.8, the time difference from the left delta line to the right delta line is a measure of $BUJ_{p\text{-}p}$.



(a) Aggressor and Victim Signals



(b) Histogram

Figure 5.3: Clock-Like Data Patterns and the Resulting BUJ Histogram

Since $BUJ_{p\text{-}p}$ is proportional to α, then when changing the spacing between traces, it should follow the trend of Figure 5.2; this is shown in Figure 5.4. For simplicity, the aggressor signals are all synchronized clock signals. Moreover, the victim trace is always sandwiched between parallel traces except when there are only one aggressor and one victim trace. As the number of aggressor traces increases from one to two, the $BUJ_{p\text{-}p}$

doubles. As the number of aggressors increases beyond two, the $BUJ_{p\text{-}p}$ increases but with lesser contribution from traces further away, hence, the cumulative effect of additional aggressor traces on $BUJ_{p\text{-}p}$ is not linear. As the number of aggressor traces increased to above ten, the $BUJ_{p\text{-}p}$ converges to a limit. This sets a practical bound to how many traces are worth considering when analyzing multiple-trace crosstalk. Figure 5.4 also allows designers to estimate how much $BUJ_{p\text{-}p}$.

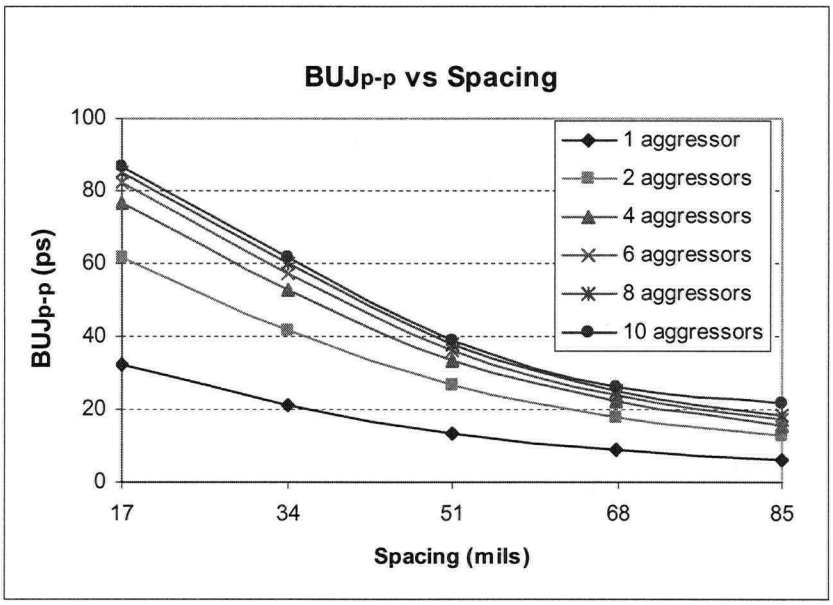
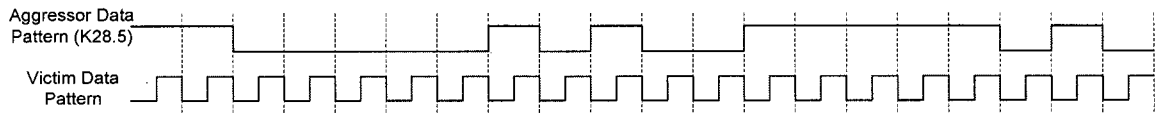


Figure 5.4: Simulation of $BUJ_{p\text{-}p}$ for Multiple Aggressors

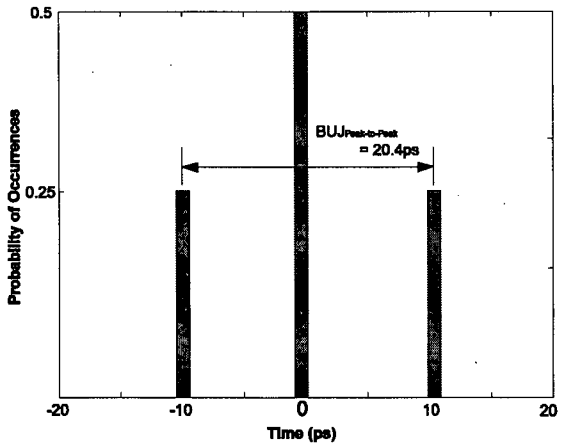
5.1.3 Peak-to-Peak BUJ with Different Aggressor Data Patterns

In this subsection, the effect of different aggressor data patterns on $BUJ_{p\text{-}p}$ is studied. Figure 5.5 (a) shows an aggressor signal with a K28.5 data pattern. K28.5 is a pattern commonly specified for jitter measurements for data rates between *1~3.125* Gbps [36]. This pattern has a length of *20-bits*. Similar to Figure 5.3, the aggressor signal is operating at half the data rate of the 1Gbps victim signal. The time deviations of the victim's falling edges are captured to generate the histogram of Figure 5.5 (b). Unlike the histogram of Figure 5.3 (b) that has only two delta lines, there are three delta lines in

Figure 5.5 (b). The left and right delta lines are due to the rising and falling edge transitions of the aggressor respectively. The middle delta line corresponds to the occurrences of consecutive *'0's* or *'1's* in the aggressor data pattern, where the lack of transitions does not have an impact on the victim's edges. The magnitude of the three delta lines in the histogram depends on the corresponding occurrences of the aggressor's edge transition events, rising edge, falling edge, no edge transition. The $BUJ_{p\text{-}p}$ in Figure 5.5 (b) is the absolute time difference between the outermost delta lines. Note that other data patterns, PRBS5 and PRBS7 [2], presented in the rest of this subsection, exhibit similar distribution but may have different $BUJ_{p\text{-}p}$.



(a) Aggressor and Victim Signals



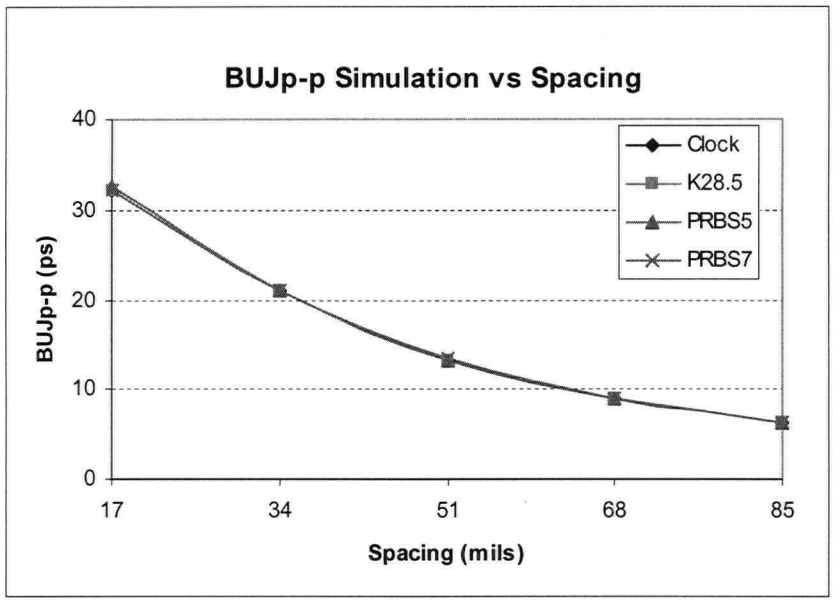
(b) Histogram (measurement)

Figure 5.5: K28.5 Aggressor Data Pattern and the Resulting BUJ Histogram

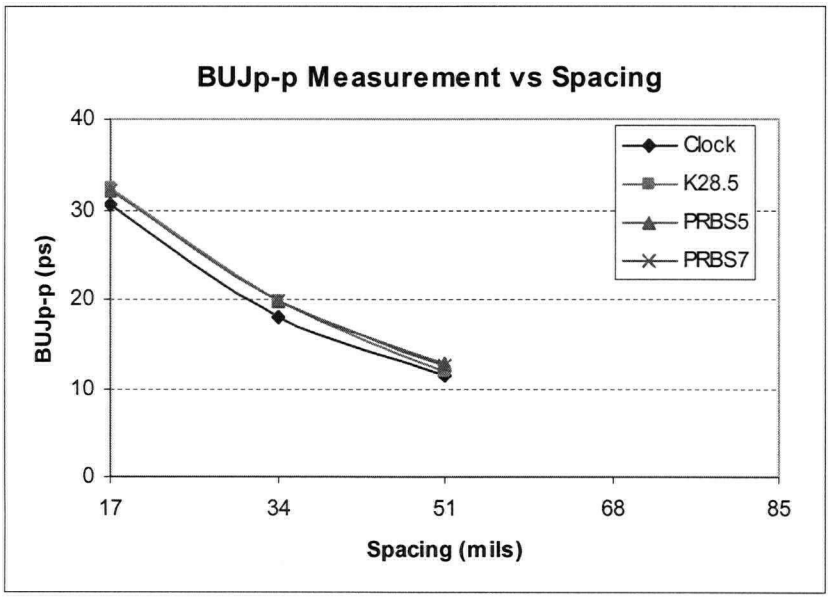
Figure 5.6 shows results for $BUJ_{p\text{-}p}$ with different aggressor data patterns for one aggressor and one victim traces. Figure 5.6 (a) shows simulation results obtained with

[2] For a description of PRBS, please refer to [9].

HSPICE and Figure 5.6 (b) shows measurement results obtained with a real-time scope. Notice the trend of $BUJ_{p\text{-}p}$ versus trace spacing. For different aggressor data patterns, the simulation results are the same. This is as expected as there are no, in our simulations, data-dependent jitter (DDJ) introduced in the aggressor sources. On the other hand, the measurement results for different data patterns show discrepancy. In particular, the clock data pattern has a smaller $BUJ_{p\text{-}p}$. This is due to the lack of DDJ, which only occurs to data patterns that have consecutive *'1's* and *'0's*, as is the case for the aggressor signal generator of the other data patterns shown. In practical situations, it is important to take into consideration the effects of DDJ, which more frequently originates from the data sources and also less frequently from reflections in PCB traces. Figure 5.6 (b) shows that the DDJ introduces an error of only *2ps* or less on our $BUJ_{p\text{-}p}$ measurements with signals generated from the Agilent *86130A*. Because of this small influence, we neglected the effect of DDJ on subsequent measurements.



(a) Simulation



(b) Measurement

Figure 5.6: Effect of Data Patterns on Peak-to-Peak BUJ

Note that measurements and simulations are in excellent agreement with only *2ps* or less difference. This shows the suitability of our simulation approach, and the effectiveness of our algorithm in removing RJ from measurements.

At this stage, we have determined the polarity of the crosstalk-induced pulse, the

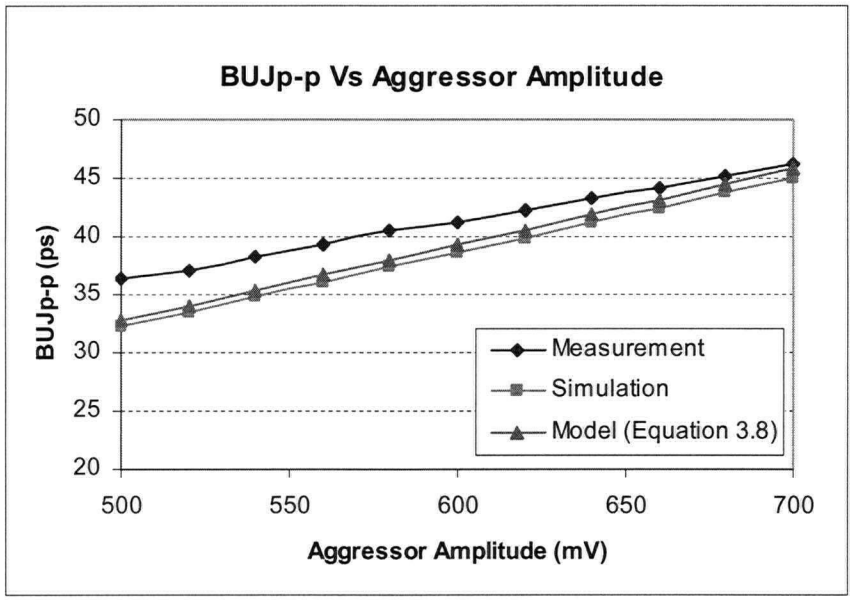
bounded effect of increasing number of aggressor signals, the effect of different aggressor data patterns on BUJ measurements, and the suitability of our measurement and simulation approach. Next, we will corroborate the equations developed in Chapter 3.

5.2 Validating the BUJ Mathematical Model

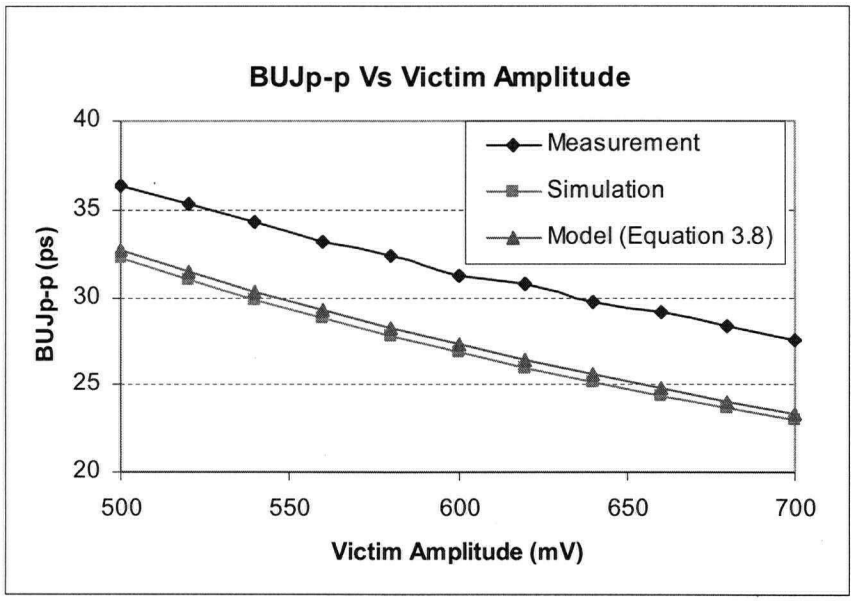
In this section, Equation 3.7 and Equation 3.8 proposed in Chapter 3 are verified. This is done by comparing their predictions against simulations and measurements. First, the equations are verified against varying voltage amplitudes and varying edge transition times of both aggressor and victim. Later, $BUJ_{p\text{-}p}$ predictions, simulations, and measurements are compared for varying spacing between traces.

Figure 5.7 shows the $BUJ_{p\text{-}p}$ values when changing the voltage amplitude of the aggressor and the victim. Signals run at *2*Gbps for the victim and *1*Gbps for the aggressors. Measurements and simulations show a direct proportionality to the aggressor and victim's voltage amplitudes; therefore, indicating that the system is operating within the first condition of Equation 3.7, thus $\Delta t = -\frac{V_p T_v}{\Delta V_v}$. As expected, the $BUJ_{p\text{-}p}$ increases with aggressor voltage since, V_p is directly proportional to ΔV_a as determined by Equation 2.10. In addition, as expected, the $BUJ_{p\text{-}p}$ decreases with the change of the amplitude of the victim edge transition, ΔV_v. Note the good agreement between predictions and simulations. However, there is a discrepancy of *4.2ps* between measurements and predictions/simulations. This discrepancy has two sources, first a consistent error of *+4.2ps* between all measurements and predictions. This is obvious along Figure 5.7 (b) and, on the left hand side of Figure 5.7 (a). This error is likely due to the equivalent-time digital sampling oscilloscope (ET-DSO) that was used for these

measurements. The second source of discrepancy is due to variations of the rising/falling transitions of the signal source for varying voltage amplitudes. In predictions and simulations, the results of Figure 5.7 (a) were computed with constant transition times. However, in measurements, small variations in transition times introduce perturbations of $BUJ_{p\text{-}p}$ such as those visible in Figure 5.7. Specifically, when the voltage amplitudes increase from *500mV* to *700mV*, the edge transition times increase from *39ps* to *41ps*. Numerical analysis of these variations explains the slope observed in Figure 5.7 (a), and confirms that the same effect has a negligible on the slope of Figure 5.7 (b).



(a) Changing the amplitude swing of the aggressor, ΔV_a

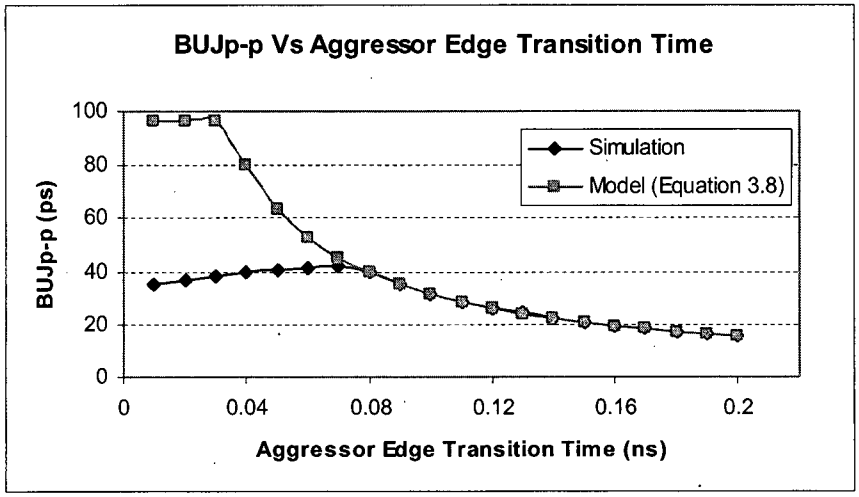


(b) Changing the amplitude swing of the victim, ΔV_v

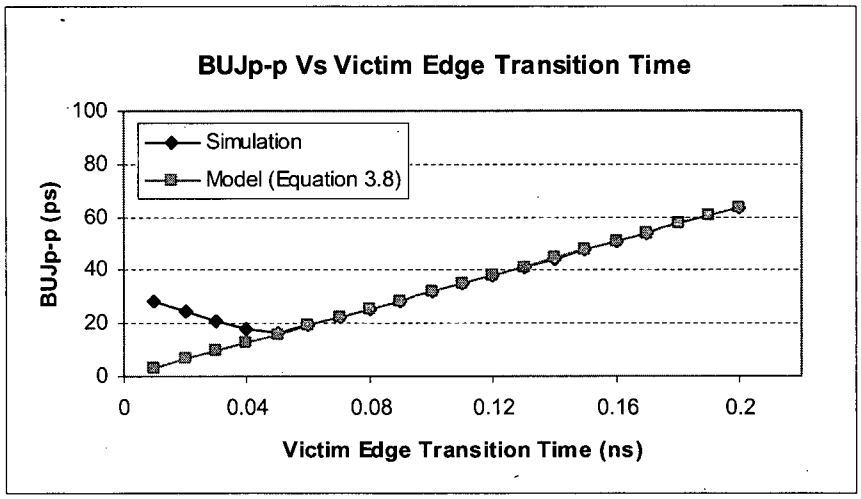
Figure 5.7: BUJ_{p-p} Vs. Aggressor/Victim Amplitudes

The edge transition time of the aggressor and victim signals are varied. Figure 5.8 shows the BUJ_{p-p} values obtained from both simulations and predictions. Full measurement results are not available because of the unavailability of instruments able to alter the rising and falling edge transition times at high-speed data rates. In Figure 5.8 (a),

the victim's edge transition time is fixed at *0.1ns* while varying the edge transition time of the aggressor signal. Note that for edge transition times above *0.07ns*, $BUJ_{p\text{-}p}$ for both the simulation and the prediction shows excellent agreement with differences of less than *1ps*. Below *0.07ns*, we observed serious discrepancies between the crosstalk-induced pulse shape derived from [28] and that calculated from *HSPICE* simulations. Specifically, for shorter edge transition times, the crosstalk-induced pulses generated by *HSPICE* become more and more trapezoidal in nature. Its crosstalk-induced pulse rising edge transition and falling edge transition is *10* times much larger than *3.2ps* from the analysis that was presented in Chapter 2.



(a) Changing the edge transition times of the aggressor, T_a



(b) Changing the edge transition times of the victim, T_v

Figure 5.8: BUJ_{p-p} Vs. Aggressor/Victim Edge Transition Times

In Figure 5.8 (b), the edge transition time of the victim signal is varied. According to Equation 3.7, BUJ_{p-p} values should be linearly proportional to the edge transition time of the victim signal. In the figure, the simulation results match with the results from prediction with differences less than *1ps*. For the same reason as in part (a), there are discrepancies at short edge transition times.

Figure 5.9 shows a comparison of measurements, simulations and predictions of

$BUJ_{p\text{-}p}$ when changing the spacing between traces. Results are for one aggressor and one victim running at *1* and *2* Gbps, respectively. All three results show the same trend seen before in Section 5.1. Note that simulations and predictions overlap and are close to measurements with *2~3ps* of difference. These results validate once more the models proposed in Chapter 3. In the next section, the algorithms described in Chapter 3 to plot BUJ histograms will be validated.

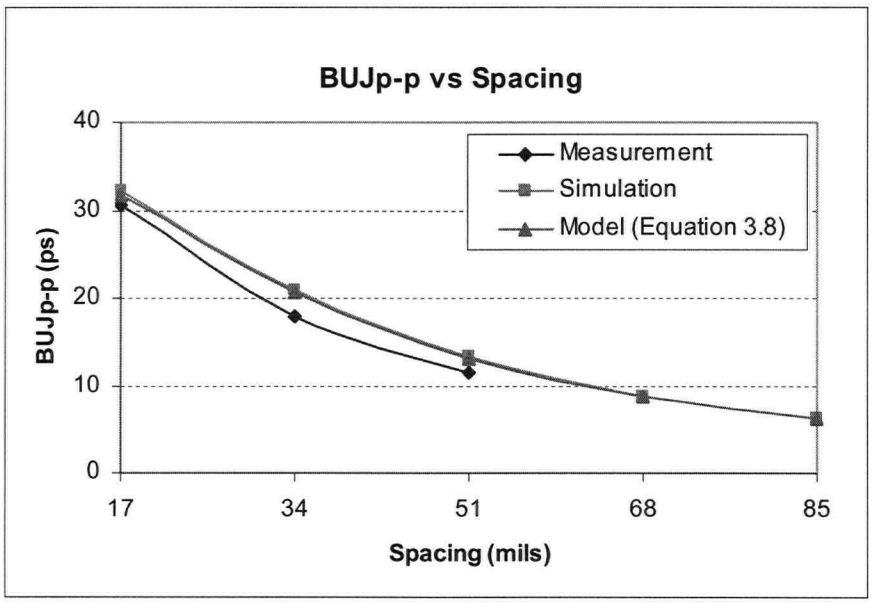


Figure 5.9: $BUJ_{p\text{-}p}$ Vs Trace Spacing

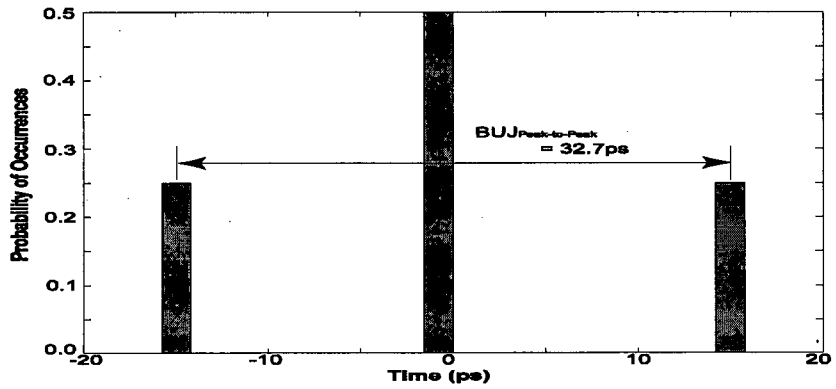
5.3 Validating the Algorithms to Generate Histograms

As can be seen from Figure 5.3 (b) and Figure 5.5 (b), histograms present more information than what a measure of $BUJ_{p\text{-}p}$ can. In this section, the histogram generated by the algorithms developed for a single aggressor trace in Chapter 3 will first be compared with the histograms obtained from measurements and simulations. Next, histograms for multiple aggressors, generated with Table 3, will be compared.

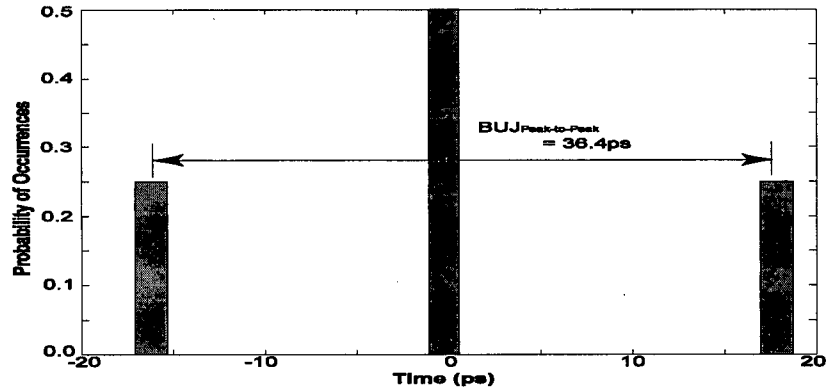
5.3.1 Two Parallel Traces

The histograms obtained using the algorithm from Table 2, as well as from measurement, and simulations, are presented in Figure 5.10. The magnitude (probability of occurrences) of the histogram is verified. The time difference of the delta lines is compared to evaluate the accuracy of the algorithm.

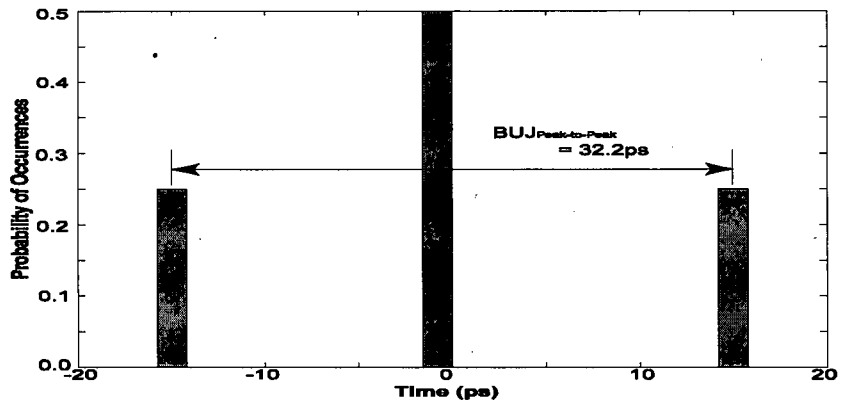
As discussed previously in subsection 5.1.3, for patterns of the aggressor signal that have consecutive *'0's* or *'1's*, there will be three delta lines in the histogram for a single aggressor configuration. Because the aggressor signal is a K28.5 data pattern (Figure 5.5(a)), there are three delta lines in the histograms of Figure 5.10. The magnitude (probability of occurrences) of the delta lines in the histogram depends on the occurrences of different edge transitions of the aggressor signal. When the aggressor signal is a K28.5 data pattern, the magnitude of the middle delta line should be twice as much as the left and right delta lines due to its edge transition pattern sequence. In Figure 5.10 (a), the delta lines generated by the algorithm have double the magnitude in the middle delta line comparing to the left and right hand side delta lines. This is also true for the histogram generated by post-processing the measurement (Figure 5.10 (b)) and simulation (Figure 5.10 (c)).



(a) Algorithm



(b) Measurement



(c) Simulation

Figure 5.10: BUJ Histograms due to a Single Aggressor Trace

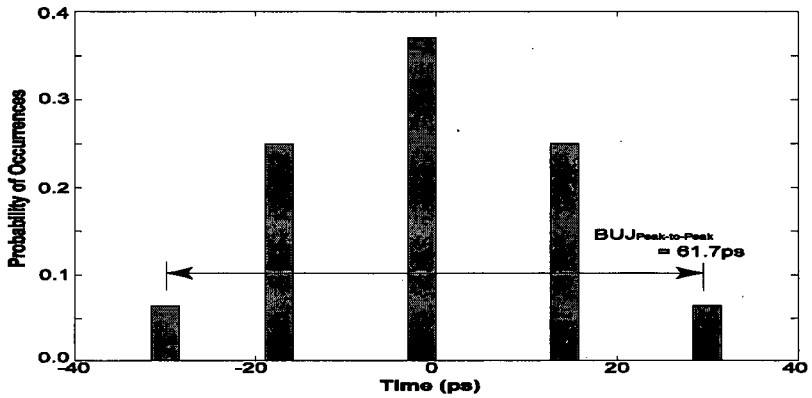
The time difference accuracy of the algorithm is evaluated by comparing the generated histogram with the histograms generated from post-processing data from measurement and simulation. Figure 5.10 (a), (b), and (c) have the middle delta line

located in the center. This is due to no edge transition activities in the aggressor signal. The left and right delta lines are due to the edge transition in the aggressor signal (i.e. a rising or a falling edge transition). Both the simulation and the algorithm (Figure 5.10 (a) and (c)) show 32.2ps and 32.7ps of $BUJ_{p\text{-}p}$. However, $BUJ_{p\text{-}p}$ from the measurement (Figure 5.10 (b)) is about ~4ps more than that from simulation and algorithm. As discussed in subsection 5.1.3 and Figure 5.6, this is due to data-dependent jitter that exists in the measurement results.

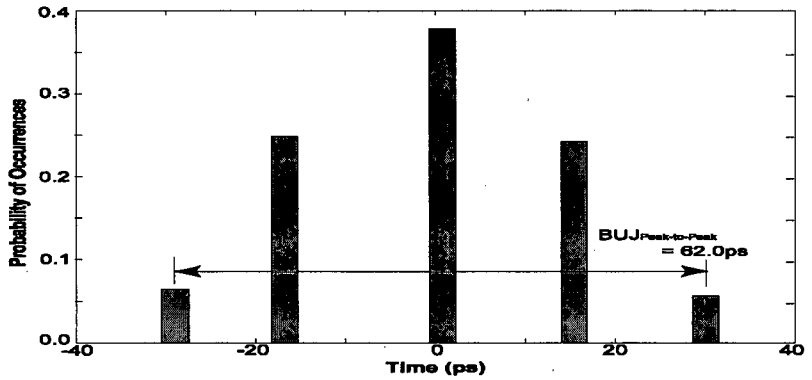
It is much faster to generate the histogram from the algorithm than it is from simulations and measurements. This is because in *HSPICE* simulation, transient analysis with picoseconds resolution is necessary to obtain accurate results. In addition, a reasonably amount of repetitive data patterns is necessary to obtain enough edge transition samples for lengthy data patterns. For example, the K28.5 data pattern has a length of twenty bits and PRBS5 has a length of thirty-one bits. Fortunately, there is no RJ in simulations; hence, the number of repetitive aggressor patterns can simply be at around *200*. On the other hand, RJ exists in measurements and hence *1000* repetitive patterns may be necessary to obtain enough samples to average out RJ. Both measurements and simulations thus suffer from long overhead times. In this particular example, the time it takes to generate the histogram by algorithm is about *5* seconds (P4 2.8GHz CPU, 512MB memory). On the other hand, simulation takes approximately 10 minutes to simulate and generate the histogram (*Sun-Blade 1000* with 800MHz CPU and 5GB of memory for *HSPICE* simulation and P4 2.8GHz CPU, 512MB memory for post-processing).

5.3.2 Multiple Parallel Traces

The algorithm of Table 3 developed in Chapter 3 adds the amplitude of the crosstalk-induced pulses from multiple aggressors and applies Equation 3.7 to find the time difference from the ideal edge crossings. Figure 5.11 below shows a comparison between the histogram generated from the algorithm in Table 3 (Figure 5.11 (a)) and the histogram generated from simulation results (Figure 5.11 (b)). The victim trace is sandwiched between two aggressor traces. To create more edge transition combinations, different aggressor data patterns are used, namely, K28.5 and PRBS5. As expected, the histograms of Figure 5.11 have five delta lines. Both simulations and predictions show almost identical histogram magnitudes and shapes with $BUJ_{p\text{-}p}$ difference of only *0.3ps*. There is a *5ps* difference between the two middle delta lines due to numerical offsets introduced in the data post-processing. Again, the time it takes to generate the histogram using the algorithm is much shorter than the time it takes from simulation. In this particular example, the time it takes to generate the histogram by algorithm is *~10* seconds (P4 2.8GHz CPU, 512MB memory). On the other hand, simulation takes approximately *20* minutes to simulate and generate the histogram (*Sun-Blade 1000* with 800MHz CPU and 5GB of memory for *HSPICE* simulation and P4 2.8GHz CPU, 512MB memory for post-processing).



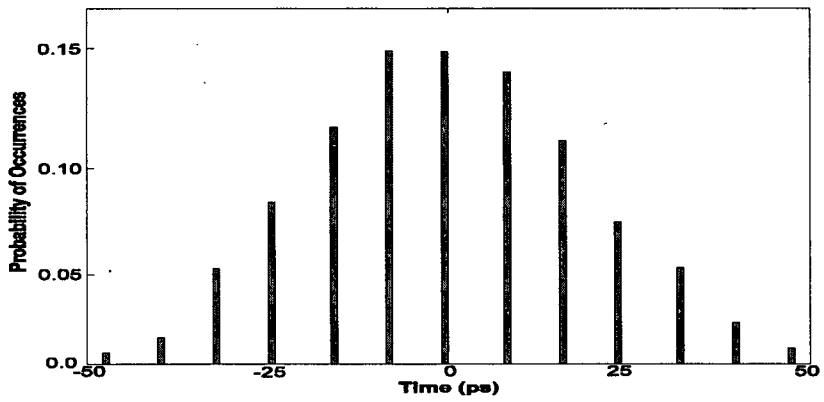
(a) Algorithm



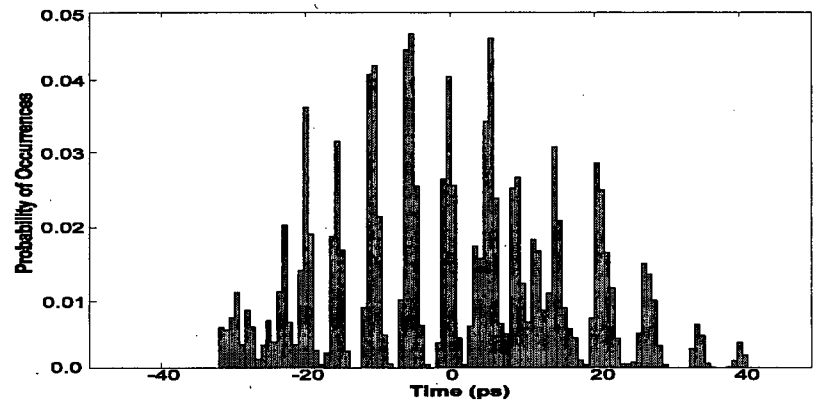
(b) Simulation

Figure 5.11: BUJ Histograms due to Two Aggressor Traces

The algorithm is further expanded to four aggressor traces with one victim trace located in the middle. Each of the aggressor traces has different data patterns of different lengths namely PRBS5, PRBS7, K28.5, and a clock signal. The histogram generated by the algorithm is shown in Figure 5.12 (a). There are more delta lines than in previous histograms due to the possible combinations of edge transition from the four aggressors. As stated in Chapter 3, there should be 81 different combinations. Since many of the combinations have the same impact on the victim edge crossing, only thirteen delta lines exist in Figure 5.12 (a).



(a) Algorithm



(b) Simulation

Figure 5.12: BUJ Histograms due to Four Aggressor Traces

Figure 5.12 (b) shows the histogram generated by post-processing the data obtained from simulation. There are many more delta lines compared to Figure 5.12 (a). This is because, in practice, the 81 combinations do have slightly different impacts on the victim. Unlike the algorithm for multiple aggressors in Table 3 where a weak coupling system is assumed, in the simulation model, aggressor signals can affect each other. Therefore, the histogram is more finely defined than Figure 5.12 (a). This is an indication that although the algorithm proved adequate for three parallel traces, for a larger trace count, the accuracy declines because the model is built around the assumption of weak coupling. In this particular example, the time it takes to generate the histogram by algorithm is *~20*

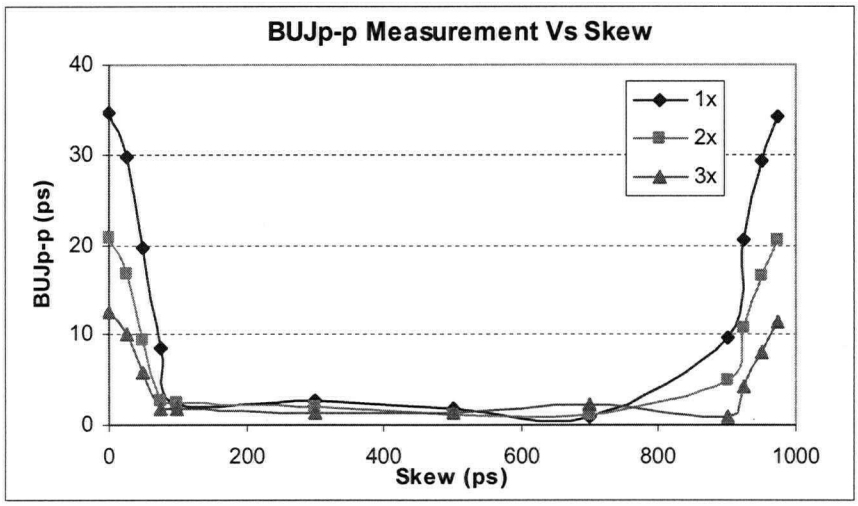
seconds (P4 2.8GHz CPU, 512MB memory). On the other hand, simulation takes approximately 30 minutes to simulate and generate the histogram (*Sun-Blade 1000* with 800MHz CPU and 5GB of memory for *HSPICE* simulation and P4 2.8GHz CPU, 512MB memory for post-processing).

The two algorithms developed (for single and multiple aggressor traces) can be incorporated with design for manufacturing (DFM) tools to allow designer to understand the effect of BUJ before manufacturing. In addition, both algorithms have very fast execution time of *10~20* seconds compared to simulation and measurement results of *10~30* minutes.

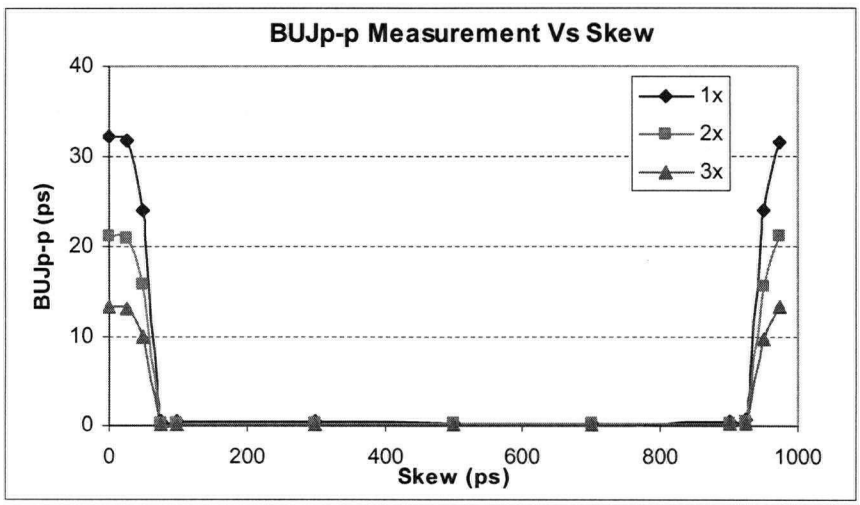
5.4 Time Skew

This section briefly studies the effect of time skew on $BUJ_{p\text{-}p}$. In high-speed interconnects, it is frequent that signals in parallel traces do not have their edge crossings phase-aligned. In other words, time skew exists between the signals. Figure 5.13 illustrates the $BUJ_{p\text{-}p}$ values obtained when changing the skew between one aggressor and one victim. Figure 5.13 (a) and (b) show measurements and simulations respectively. The measurements were performed by introducing a time skew between two signals generated with the Agilent 86130A. Also, the measurements were performed with different trace spacings of *1x, 2x,* and *3x*. In both measurements and simulations, the victim is a *1GHz* clock signal, and thus has a *1000ps* period. In both Figure 5.13 (a) and (b), the skew is varied over one full period of the victim signal. The highest $BUJ_{p\text{-}p}$ value occurs when there is no skew between the aggressor and victim signals. As the skew increases, the $BUJ_{p\text{-}p}$ reduces. At about *100ps* time skew, there is almost no impact on the victim edges. As the skew increases to *900ps*, the $BUJ_{p\text{-}p}$ increases again. The

maximum $BUJ_{p\text{-}p}$ occurs again when the time difference matches one full period, i.e., *1000ps*. Notice that as the spacing between traces increases from *1x* to *2x* and *3x*, the $BUJ_{p\text{-}p}$ decreases as expected with the trend in Subsection 5.1.2.



(a) Measurement



(b) Simulation

Figure 5.13: $BUJ_{p\text{-}p}$ Vs Time Skew of a 1GHz Victim Signal

Figure 5.13 (a) and (b) showed some discrepancies between $BUJ_{p\text{-}p}$ values. These discrepancies are obvious between time skews of *100ps* to *900ps*. This is because measurement results unavoidably include other sources of jitter such as DDJ.

Chapter 6 Conclusions and Future Work

6.1 Summary and Contributions

In this thesis, we investigated Bounded Uncorrelated Jitter (BUJ) on high-speed PCB interconnects. First, we showed through simulations that as the spacing between adjacent traces increases, the impact of BUJ slowly converges to zero. In particular, when the number of aggressor traces with identical data pattern is increased above *10*, the $BUJ_{p\text{-}p}$ values converge to a limit. This observation sets a practical bound for designers concerned with the effect of $BUJ_{p\text{-}p}$ in microstrip multi-trace configurations. Measurements and simulations of BUJ were analyzed for different data patterns, showing that $BUJ_{p\text{-}p}$ is only affected by a few picoseconds of DDJ for *1~2*Gbps signals.

Secondly, we reported BUJ measurements made from a custom high-speed PCB test-bed as well as simulations performed with the *HSPICE's* 2-D fieldsolver. Measurement and simulation results were compared and shown to be in good agreement of only *~2ps* difference. In addition, an algorithm to post-process the results from simulation and measurement was also developed. This algorithm in Table 4 allows averaging of symmetrical jitter subcomponents such as RJ and PJ so that BUJ can be directly analyzed.

Thirdly, we proposed a new model (Equation 3.7) based on signal superposition and the graphical analysis of the crosstalk-induced pulse to calculate the time difference between the distortion-free and the distorted victim edge crossings. This model was further extended to calculate $BUJ_{p\text{-}p}$ (Equation 3.8). Predictions with our equations were compared with simulations and measurements by varying the amplitude and edge transition time of the aggressor and victim signals. These results confirm the $BUJ_{p\text{-}p}$

analytical model. However, the results also showed that at short edge transition times, discrepancies exist between the models and simulations. We suspect this is due to limitations of the *HSPICE* transient analysis at short edge transition times.

Fourthly, we also developed new algorithms in Table 2 and Table 3 that can generate histograms from single and multiple aggressors. Both algorithms show excellent agreement with measurement and simulations. For example, in a two aggressor traces situation, the difference between algorithms and simulations is *0.3ps* of $BUJ_{p\text{-}p}$. When the number of aggressor traces increases to four or beyond, accuracy of the algorithms decreases due to their assumption of weak coupling between aggressor traces. Both algorithms have very fast execution time, of *5~20 seconds*, compared to simulation and measurement result times, of *10~ 30* minutes, which require post-processing of data. These algorithms can be incorporated into design for manufacturing (DFM) tools, which would allow designers to know the effect of BUJ before manufacturing the PCB interconnects.

Lastly, a brief investigation on the time skew between the aggressor and the victim signal shows that the largest jitter impact occurs when there is no time skew between signals. Thus, showing that our previous results where no skew was considered covered the worst-case BUJ scenarios.

6.2 Future Work

The mathematical models (Equation 3.7 and 3.8) and algorithms (Algorithm 1 and 2) to characterize the effect of BUJ have opened the door to possible future research opportunities, some of which are explained in the next few subsections.

6.2.1 Variable Edge Transition Time

In Figure 5.8, the edge transition time is varied by comparing the result obtained from simulation and calculated from Equation 3.8. However, due to equipment limitations, no complete set of measurement results are available. A possible simple experiment is to apply the aggressor and the victim signal through transition time converters to reshape the transition times. Another alternative is to employ a Gbps source with controllable rise and fall times.

Further measurements of the crosstalk-induced pulse may be necessary to find the root-cause of the discrepancies between the results obtained from simulations and the predicted results. It is possible that the shape of the crosstalk-induced pulse at shorter edge transitions of the victim and aggressor signals is not as described in Figure 2.10(b) for short edge transition times. In addition, it is also possible that the discrepancies arise from limitations of *HSPICE*.

6.2.2 BUJ Mathematical Model with Time Skew

In Section 5.4, investigations of the impact on the victim edge by varying the time skew between signals were performed through both, simulations and measurements. However, no mathematical model is available to predict $BUJ_{p\text{-}p}$ with time skew between signals. Equations 3.7 and 3.8 developed in Chapter 3 use superposition by studying graphically, the crosstalk-induced signals. It is also possible to develop a BUJ model that takes time skew into consideration. This model can be incorporated in the algorithm developed in Chapter 3 to plot histograms.

6.2.3 BUJ under Strong Coupling

In Subection 5.3.2, as the number of aggressor traces is increased to four, the accuracy of the histogram generated by the algorithm decreases. This is because the algorithm assumes a weak-coupling system. Further investigation of a strong coupling system, which takes into consideration the coupling effect across the aggressors, would allow the development of models and algorithms to generate accurate histograms for multiple aggressors.

6.2.4 Frequency-dependent Parameters

To study and characterize BUJ beyond approximately *4*Gbps requires consideration of frequency-dependent issues. The most well known frequency-dependent issue is the skin effect. At high frequencies, current does not flow uniformly throughout the cross-sectional area of PCB traces. Instead, the current will migrate toward the surface of the trace [3]. The skin effect will result in frequency-dependent resistance and inductance. In addition to the skin effect, the dielectric constant of the PCB substrate is also frequency dependent. These issues need to be studied and addressed for multi-Gbps transmissions.

References

[1] E. Bogatin, *Signal Integrity –Simplified*, Prentice Hall, 2003, ch. 1, 10.

[2] F. Saal, "A Case for Signal Integrity Verification," EE Times [Online]. Available: http://www.eedesign.com/story/OEG20000114S0059

[3] S. H. Hall, G. W. Hall, J. A. McCall, *High-Speed Digital System Design: A Handbook of Interconnect Theory and Design Practices*, New York: Wiley-Interscience 2001, ch. 1-3.

[4] *Measuring Jitter in Digital Systems,* Agilent Technologies Inc., Application Note 1448-1, June 2003 [Online], Available: http://cp.literature.agilent.com/litweb/pdf/5988-9109EN.pdf

[5] Johnnie Hancock, "Jitter - Understanding It, Measuring It, Eliminating It Part 1: Jitter Fundamentals," *High Frequency Electronics*, April 2004, pp. 44-50.

[6] *Characterize Jitter Using Histograms*, Lecroy Corporation, Application Note L.A.B. 715, [Online], Available: http://www.lecroy.com/tm/library/LABs/PDF/LAB715.pdf

[7] Mike Li, Jan Wilstrup, "Paradigm Shift For Jitter and Noise In Design and Test > 1Gb/s Communication Systems" IEEE International Conference on Computer Design, 2003.

[8] J. Hancock, "Jitter - Understanding It, Measuring It, Eliminating It Part 3: Jitter Fundamentals," *High Frequency Electronics*, April 2004, pp. 28-35.

[9] *Jitter Analysis and Correlations*, Tektronix Inc., Application Note, 2003 [Online], Available: http://www.tektronix.com/jitter

[10] Separating Jitter Sources, Lecroy Corporation, Application Note L.A.B. 754, [Online], Available: http://www.lecroy.com/tm/library/LABs/PDF/LAB754.pdf

[11] Y. Cai, S.A. Werner, G. J. Zhang, M.J. Olsen, and R.D. Brink, "Jitter Testing for Multi-Gigabit Backplane SerDes," in *Proc. 33rd IEEE International Test Conference (ITC 2002)*, Piscataway, NJ: IEEE Press, Sept. 2002, pp. 700-710.

[12] *Understanding Jitter*, WAVECREST Corporation, 2001 [Online], Available: http://www.wavecrest.com/technical/VISI_6_Getting_Started_Guides/6understanding.PDF.

[13] *Jitter in Digital Communication Systems, Part 1*, MAXIM Integrated Products, Application Note HFAN-04.0.3, Rev0, September 2001 [Online], Available: http://pdfserv.maxim-ic.com/en/an/5hfan403.pdf

[14] John Patrin, Mike Li, "Comparison and Correlation of Signal Integrity Measurement Techniques," *DesignCon Conference*, 2002.

[15] J. Sun, M. Lee, J. Wilstrup, "A Demonstration of Deterministic Jitter (DJ) Deconvolution," in *Proc. 19th IEEE Instrumentation and Measurement Technology Conference (IMTC 2002)*, Anchorage, AK, 2002, pp. 293-298.

[16] *Jitter Analysis Techniques for High Data Rates*, Agilent Technologies Inc., Application Note 1432, February 2003 [Online], Available: http://cp.literature.agilent.com/litweb/pdf/5988-8425EN.pdf

[17] *Understanding and Characterizing Timing Jitter*, Tektronix Inc., Application Note, 2003 [Online], Available: http://www.tektronix.com/jitter

[18] *Jitter Impacts on Equalized Link Performance Part 2,* Comms Design, April 2003 [Online], Available: http://www.commsdesign.com/showArticle.jhtml?articleID=16500827

[19] J. Quirk, "Measuring Deterministic Jitter with a Scope," Communication Systems Design, July 2002, pp. 26-33.

[20] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, and A. Ivanov, "Jitter Models for the Design and Test of High-Speed (Gb/s) Serial Interconnects," *IEEE Design and Test of Computers*, vol. 21, no.4, Jul-Aug 2004, pp. 302-313.

[21] *International Technology Roadmap for Semiconductors*, 2003 Edition: System Drivers, pp. 14, 2003.

[22] *Measuring Crosstalk on a Fibre Channel signal caused by an adjacent FC Channel Signal*, WAVECREST Corporation, 2001 [Online], Available: http://www.wavecrest.com

[23] J. D. Kraus, *Electromagnetics*, New York: McGraw-Hill, 1984, pp. 40.

[24] C. S. Walker, *Capacitance, Inductance and Crosstalk Analysis*, Boston, MA: Artech House, 1990, ch. 1-2.

[25] M. V . Schneider, "Microstrip Lines for Microwave Integrated Circuits," *Bell Syst. Tech. J.*, vol. 48, no. 5, pp. 1421-1444, May-June 1969.

[26] *How Crosstalk Causes Problems*, EE Times [Online]. Available: http://www.eedesign.com/story/OEG20020604S0046

[27] B. Young, *Digital Signal Integrity, Modeling and Simulation with Interconnects and Packages*, Upper Saddle River, NJ: Prentice Hall, 2001, pp. 98-104, 200-207.

[28] D.B. Jarvis, "The Effects of Interconnections on High-Speed Logic Circuits," *IEEE Trans. on Electronic Computers*, vol. EC-12, pp. 476-487, Oct. 1963.

[29] J. A. Defalco, "Reflection and crosstalk in logic circuit interconnections," *IEEE Spectrum*, vol. 7, pp. 44-50, Jul. 1970.

[30] L. Carin and K. J. Webb, "Isolation Effects in Single and Dual Plane VLSI Interconnects," *IEEE Transactions on Microwave Theory Techniques*, vol. 38, pp. 396-404, Apr. 1990.

[31] Agrawal, A. P., C. S. Chang, and D. A. Gernhart, "Design Considerations for Digital Circuit Interconnections in a Multilayer Printed Circuit Board," *IEEE International Conf. on Computer Design: VLSI in Computers and Processors*, pp. 472-478, 1991.

[32] C.C. Chai, B.K. Chung, and H.T. Chuah, "Simple Time-Domain Expressions for Prediction of Cross-talk on Coupled Microstrip Lines", *Progress in Electromagnetics Research*, pp. 147-175, 2002.

[33] A. Kuo, T. Farahmand, N. Ou, S. Tabatabaei, and A. Ivanov, "Jitter Models and Measurement Methods for High-Speed Serial Interconnects", to be appear in IEEE International Test Conference, Charlotte, NC, 2004.

[34] *International Technology Roadmap for Semiconductors*, 2003 Edition: Test and Test Equipment, pp. 19, 2003.

[35] ORT82G5-B1 Evaluation Board User Manual, Lattice Semiconductor Corporation, January 2003 [Online], Available: http://www.latticesemi.com/lit/docs/manuals/ort82g5_ev.pdf

[36] *Measuring Jitter in Ethernet and Fibre Channel Components*, MAXIM Integrated Products, Application Note HFAN-04.5.0, December 2000 [Online], Available: http://pdfserv.maxim-ic.com/en/an/7hfan450.pdf